1988

Regression Based Allowance Policy Determination Procedures in a General Job Shop: An Evaluation in Terms of Completion Inaccuracy Penalties

Edward S. Gee

School of Business
Virginia Commonwealth University

This is to certify that the dissertation prepared by

Edward S. Gee entitled
"Regression Based Allowance Policy Determination Procedures in a General Job Shop: An Evaluation in Terms of Completion Inaccuracy Penalties"

has been approved by his committee as satisfactory completion of the dissertation requirement for the degree of Doctor of Philosophy

[redacted]
Director of Dissertation

[redacted]
Committee Member

[redacted]
Committee Member

[redacted]
Committee Member

[redacted]
Committee Member

[redacted]
Associate Dean for Graduate Studies

[redacted]
Dean

3-17-88
Date

Regression-Based Allowance Policy Determination
Procedures in a General Job Shop: An Evaluation
in Terms of Completion Inaccuracy Penalties

A dissertation submitted in partial fulfillment
of the requirements for the degree of Doctor of
Philosophy at Virginia Commonwealth University

By

Edward S. Gee
Bachelor of Arts, University of Virginia, 1973
Master of Business Administration, University of North Carolina, 1975

Director: Dr. Charles H. Smith
Associate Professor
Decision Sciences and Business Law

Virginia Commonwealth University
Richmond, Virginia
March, 1988

Acknowledgements

I would like to thank the members of my committee, Drs. Canavos, Redmond, Kurtulus, and McDermott. Their support, constructive critiques, and suggestions of all sorts have aided my efforts a great deal. Special appreciation goes to the Chairman of my committee, Dr. Charles H. Smith, who displayed more initiative, support, and genuine interest than any doctoral candidate should have the right to expect.

The majority of thanks, however, goes to the four long-suffering ladies in my life: my wife Kristie and my three daughters Adin, Dani, and Lindsay. They made the considerable sacrifices inherent in sharing a husband/father with a full-time business career and a full-time doctoral program. My accomplishments are every bit as much products of their efforts as of mine.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

List of Tables

List of Tables (Continued)

List of Figures

List of Figures (Continued)

Abstract

REGRESSION-BASED ALLOWANCE POLICY DETERMINATION
PROCEDURES IN A GENERAL JOB SHOP: AN EVALUATION
IN TERMS OF COMPLETION INACCURACY PENALTIES

Edward S. Gee

School of Business - Virginia Commonwealth University, 1988

Major Director: Dr. Charles H. Smith

This dissertation addresses the problem of setting due dates to minimize completion inaccuracy penalties in a general job shop environment. In this simulation study, lateness penalties are generated by four defined functions: lateness variance, mean squared lateness, mean absolute lateness, and semi-quadratic lateness. Each of these functions assigns positive penalties to both early and late job completions.

The study proposes and demonstrates the benefits of an iterative simulation-regression procedure in determining allowance policies. Advantages of operation-based dispatching rules over job-based dispatching rules, as well as improvements to traditional methods of setting operation due dates, are demonstrated. Characteristics and benefits of incorporating shop congestion variables in due date setting procedures under different combinations of expected shop utilization and processing time assumptions are evaluated.

Chapter 1

Introduction

This dissertation presents a simulation study on the problem of scheduling jobs through a multi-facility shop. The study extends both the scope of scheduling research and its applicability to real-world shops by addressing three important subclasses of problems to which existing literature devotes little attention:

1. the set of problems in which both negative lateness (the completion of a job prior to its due date) and positive lateness (the completion of a job after its due date) incur a positive penalty,

2. the set of problems in which the shop utilizes relevant job-related information (such as number of tasks per job) <u>and</u> shop-related information (such as number of jobs in shop) to assign, free from external constraints (i.e., constraints imposed by the client or the marketplace), an expected completion date to each job upon its arrival at the shop, and

3. the set of problems in which job-based dispatching rules (such as "earliest job due date") are compared with operation-based dispatching rules (such as "earliest pending operation due date") under a variety of allowance policies (i.e., methods of estimating job and operation completion dates) and shop conditions.

This study demonstrates new procedures which significantly improve existing methodology in the areas of allowance policy determination and implementation of operation-based dispatching rules. Further, the benefits and characteristics of utilizing shop congestion variables (i.e., variables reflecting how crowded the shop is at the time of a given job's arrival) in allowance determination are evaluated statistically.

Much research has been devoted over the past thirty years to the job shop scheduling problem. The vast majority of this research has concerned the evaluation of different heuristic dispatching rules by which to select a job from an existing queue at each machine.

Most of these studies assume (either explicitly or implicitly) that due dates are either externally invoked or internally set, based solely on job characteristics, subject to specific marketplace or customer constraints. An example of such an external constraint is "mean job allowance must be seven times mean job processing time". Few studies have addressed the utilization of shop congestion information in setting estimated job completion dates.

This dissertation proceeds by stating the research problem (including brief definitions of problem concepts), and then discussing the significance of the research. The next section examines the general shop scheduling problem, and defines further concepts and terminology pertinent to the study. Chapter 1 concludes with a discussion of the scope and limitations of the research, and a statement of the hypotheses tested.

Chapter 2 reviews related research, and Chapter 3 discusses the research method in terms of its design and analysis procedures. Chapters 4 and 5 present and discuss the results of the research. Chapter 6 discusses managerial implications and directions for future research. A list of references is given, and the Appendices that follow provide supporting documentation as well as an alphabetized glossary of variables and acronyms.

The Research Problem

The problem under study is "To what extent does the unconstrained regression-based choice of allowance policy, interacting with various dispatching rules and shop characteristics, affect penalties associated with inaccurate job completion times in a general job shop with dynamic and probabilistic job arrivals?"

Scheduling decisions in most industrial job shops are decentralized (Kanet, 1979). Each work station chooses the next job to be processed from the queue that exists at that station based on a "dispatching rule."

Dispatching rules may be categorized as either "operation-based" or "job-based". An operation-based dispatching rule utilizes information about the pending operations of available jobs to prioritize those jobs, whereas a job-based dispatching rule utilizes information about the overall jobs themselves. An example of an operation-based dispatching rule is "select from the queue the job that has the earliest due date for its pending operation". An example of a job-based dispatching rule is "select from the queue the job that has the earliest job due date".

The allowance of job i is defined as the due date of the job minus the arrival date of the job, and represents the time job i can spend in the shop before becoming late. The shop's "allowance policy," therefore, determines a job's due date based on information available at the time the job arrives at the shop.

An allowance policy may be categorized as either "local" or "global." A local allowance policy uses only job-related information (such as number of tasks and total required processing time) to calculate a due date. A global allowance policy uses both job-related and shop-related information (such as number of jobs in the shop as of the job's arrival) to calculate a due date.

The phrase "regression-based choice of allowance policy" refers to a procedure in which specific coefficients in an allowance equation are determined from a regression analysis of the output from a previous simulation. For example, one may wish to set the allowance of each job, as it arrives, as a constant plus a fixed multiple of the job's total expected processing time. In this case, one could fix an initial constant and an initial multiple arbitrarily (for example, 0.0 and 6.0, respectively) and run a simulation. One could then perform a simple linear regression on the output from that simulation, using the time actually spent in the shop by each simulated job as the dependent variable and the expected required processing time of each job as the independent variable. The resulting linear equation would be a regression-based allowance policy.

Let d_i denote the expected completion date of a job (upon arrival at the shop), and let C_i denote the actual realized job completion date. Job lateness is defined as

$$L_i = C_i - d_i. \qquad (1.1)$$

If actual completion occurs after the due date (i.e., the job is completed later than expected), L_i will be positive; if actual completion occurs before the due date (i.e., the job is completed earlier than expected), L_i will be negative.

Penalties associated with early and late job completions are reflected by defined penalty functions. This research presents analysis on each of four different penalty functions: quadratic for L_i about the mean lateness, quadratic for L_i about zero, linear for L_i about zero, and linear for negative L_i while quadratic for positive L_i. For simplicity, the linear portion of a penalty function is assumed to have a slope of one, and therefore is equal to the absolute value of L_i.

To facilitate comparison with other research, penalties are reported on an "average penalty per job" basis. Therefore, the four penalty measures in this study are lateness variance, mean squared lateness, mean absolute lateness, and semi-quadratic lateness, and are defined as follows:

$$VAR = \sum_{i=1}^{n} \frac{(L_i - \bar{L})^2}{n} \quad , \tag{1.2}$$

$$MSL = \sum_{i=1}^{n} \frac{L_i^2}{n} \quad , \tag{1.3}$$

$$MAL = \sum_{i=1}^{n} \frac{|L_i|}{n} \quad , \text{ and} \tag{1.4}$$

$$SQL = \sum_{i=1}^{n} \frac{\begin{bmatrix} L_i^2 \text{ for positive } L_i \\ |L_i| \text{ for negative } L_i \end{bmatrix}}{n} \quad . \tag{1.5}$$

A shop is a set of facilities associated with a given set of jobs. A job consists of one or more operations, each of which must be processed at a specific type of facility. The term "job shop" typically refers to a shop in which the order of a job's operations is unknown prior to the job's arrival.

In a "dynamic" shop the jobs arrive individually over time, either "deterministically" (future arrival times are known with certainty), or "probabilistically" (arrival times follow some stochastic process). In a "static" shop, all jobs to be scheduled arrive at the same time.

Significance of the Study

Several key characteristics of this study concern aspects of scheduling largely unaddressed by existing research:

1. The minimization of positive penalties incurred by both early job completions and late job completions.

2. Allowance policies, free from external constraints, that are based on simulation-regression techniques. The iterative procedure used in this study is unique in shop research, and provides significant improvements in completion accuracy over past methods.

3. The evaluation of job-based dispatching rules vs. operation-based dispatching rules under several shop and procedural environments.

4. The evaluation of benefits of incorporating global information into the due date setting procedure.

5. The evaluation of the sensitivity of allowance procedures and dispatching rules to changes in shop characteristics.

The general areas of completion inaccuracy penalties and internal allowance policy merit further discussion.

Completion Inaccuracy Penalties

A strong intuitive foundation exists for assuming that a shop incurs penalties for early completion as well as late completion. Unless finished jobs can be shipped prior to their respective due dates, early completions will increase a shop's monetary and space investments in finished good inventory.

Given a shop in which resource constraints are significant, early completions generally occur at the expense of late completions, and therefore indicate a misallocation of resources. A shop that quotes due dates that tend to exceed completion times in a systematic manner may be foregoing a potential competitive edge.

MRP systems, which depend on accurate delivery of subassemblies, are becoming increasingly popular in the real world. The advantages of increasing delivery accuracy are detailed by Fry *et*. *al*. (1987).

Putnam, Everdell, Dorman, Cronan, and Lindgren (1971) reported that the preference of many firms is for scheduling techniques that minimize the variance of completion times around end due dates. This is analogous to the minimization of this study's VAR and MSL penalty measures. Few studies in the reviewed literature, however, specifically addressed the minimization of VAR or MSL as objectives. For examples, see Kanet (1979) and Ragatz and Mabert (1984).

Panwalker and Iskander (1973) noted a marked discrepancy between the "preferred" objective measures of research and those of industry. Actual firms placed a higher priority on meeting due dates than on typical research objectives such as minimizing mean flowtime. Similar opinions were noted by Kanet and Hayya (1982) and Baker (1984). Hax and Candea (1984) pointed to such "misdirected" research effort as a primary reason for the relative lack of application of theoretical developments to actual industrial settings. Similar findings were noted by Melnyk et. al. (1986).

Regression-Based Internal Allowance Policies

Only a few studies have addressed allowance policy as a decision variable. Those studies demonstrated that both the choice of dispatching rule and the choice of allowance policy significantly affect aggregate performance measures (Kanet, 1979; Conway, 1965a). The importance of due date assignment problems has been voiced in previous research (Weeks and Fryer, 1977; Smith and Seidman, 1983). Further, few studies have assumed that the choice of allowance policy was free from external constraints such as an imposed mean job allowance (Baker and Bertrand, 1981).

Kanet (1979) and Conway (1965a and 1965b) examined factorial designs of various dispatching rules and various allowance policies. However, both used allowance policies that were externally constrained in that the mean allowances were set equal to arbitrary levels. Forcing allowance policies to conform to such an external constraint will affect mean lateness and MSL. Further, both studies based their allowance policies solely on job-related information.

The statistic of lateness variance has been reported in an incidental manner in several studies. It is tempting to conclude that lateness variance under conditions of an external mean allowance constraint may be compared directly to MSL under conditions of no constraint, since lateness variance is calculated about any observed mean lateness.

Data presented incidentally by Kanet (1979), however, permit a posteriori analysis that shows a significant relationship (observed Chi-Squared with 4 d.f. = 136) between mean lateness and lateness variance over all pairwise combinations of dispatching rules and allowance policies. In short, since mean lateness and lateness variance are significantly related, the latter may not be used to infer "unconstrained" results from studies that assumed a mean allowance constraint.

Few available studies attempted to fit allowance policies to the inherent tendencies of various dispatching rules. Further, few published studies have evaluated allowance policies which incorporate global information in the setting of due dates (for example, the level of shop congestion at the time of job arrival). Intuitively, one would expect that the allowance set for a job that arrives when the shop is crowded should be larger than the allowance set for an identical job that arrives when the shop is empty. This conclusion has been supported in several previous studies (for example, Baker, 1984).

Ragatz and Mabert (1984) did address the use of an allowance policy which was generated by multiple regression techniques (termed RMR in their study). This was the only study found that attempted to incorporate numerous job-related and shop-related factors, simultaneously, as independent variables in the allowance estimator. It is, therefore, the most similar in intent to this research.

There are, however, several potential characteristics of their study that may limit the usefulness and validity of their conclusions. They evaluated only three dispatching rules (Shortest Processing Time, First Come First Served, and Minimum Job Slack). The first two rules do not incorporate assigned due dates in their prioritization, and all three have been shown to be inferior to other rules in minimizing objective functions related to lateness variance (Kanet, 1979 and Conway, 1965a).

The variable that Ragatz and Mabert used to reflect total shop congestion was the total number of jobs in the shop at the time of a job's arrival. This is intuitively inferior to other potential indicators such as total number of jobs (or total required processing times) in the shop that require the same machines as does the job being examined.

A potentially more telling limitation, however, concerns the procedures by which Ragatz and Mabert estimated regression coefficients in the model. They generated their allowance equation by analyzing results from a single pilot simulation, and used that equation in subsequent evaluative simulation runs. In other words, a different allowance procedure was used in the pilot simulation than was used in the evaluative simulations.

This fails to acknowledge the fact that different allowance procedures will change the characteristic performance of any dispatching rule that incorporates due dates in its prioritization (such as selecting the job that has the minimum slack, termed MINSLK by Ragatz and Mabert). In effect, for MINSLK their parameter estimation stage and their performance evaluation stage may have been performed on two different populations. This may account for their results that showed that while the RMR procedure dominated all other procedures with SPT and FCFS, it was marginally outperformed by two other simpler procedures when using the dispatching rule MINSLK.

This research addresses that limitation by iterating the simulation-regression procedure, producing successive allowance policies that tend to converge to a more stable equation. This procedure (to be discussed later in further detail) also begins by arbitrarily setting an initial allowance equation, running a simulation, and determining a revised allowance equation based on regression analysis of the initial simulation output. However, this procedure then continues by running a second simulation using the revised allowance equation, performing a regression analysis of the results, and revising the allowance equation again. The process is continued through subsequent iterations until a predetermined stopping point is reached (also to be discussed later in further detail). Such an iterative technique is analogous to the Markov decision process of policy iteration. Significant decreases in inaccuracy measures result.

Ragatz and Mabert did not investigate the robustness of their results to changes in shop characteristics.

The General Shop Scheduling Problem

The shop scheduling problem is to order the operations to be performed at each shop facility subject to routing and shop constraints, such that some measurable function of the ordering is optimized (Salvador, 1978). Research generally classifies a shop's structure as "parallel," "flow," or "job."

A "parallel shop" consists of several identical facilities, in which each job consists of a single operation to be performed on any one of those facilities (for example, checkout stations in a supermarket). A "flow-shop" is a set of different facilities, in which each job consists of identical operations in identical order (for example, an automobile assembly line). A "job shop" is a set of different facilities in which the type, number, and order of required operations for any given job are unknown prior to arrival (for example, a general-purpose machine shop).

Past research concerning the scheduling problem has sought to optimize a variety of objective functions. Examples of such goals are the minimization of makespan, mean flowtime, mean lateness, and mean tardiness. Makespan is the total time required to process n jobs (with static arrival) through a shop. If r_i represents the arrival time of job i at the shop (recall that C_i represents the time of that job's completion), then the flowtime for job i is defined as

$$F_i = C_i - r_i. \tag{1.6}$$

Tardiness is defined as L_i if L_i is positive and zero otherwise.

Objective functions may be classified as "regular" or "non-regular" (Conway, 1965a). The value of a regular function will increase only if at least one job flowtime increases. For example, mean flowtime is a regular function but the variance of job flowtimes is not.

The allowance of any job i is the due date of the job minus the arrival date of the job, or

$$a_i = d_i - r_i. \tag{1.7}$$

The time that job i stays in the shop is the sum of its actual processing time, p_i, and its actual waiting time, w_i. The allowance for job i consists of its expected processing time, $\hat{p}_i$, plus its expected waiting time, $\hat{w}_i$. Figure 1.1 visually displays the relationships among these characteristics.

Scope and Limitations

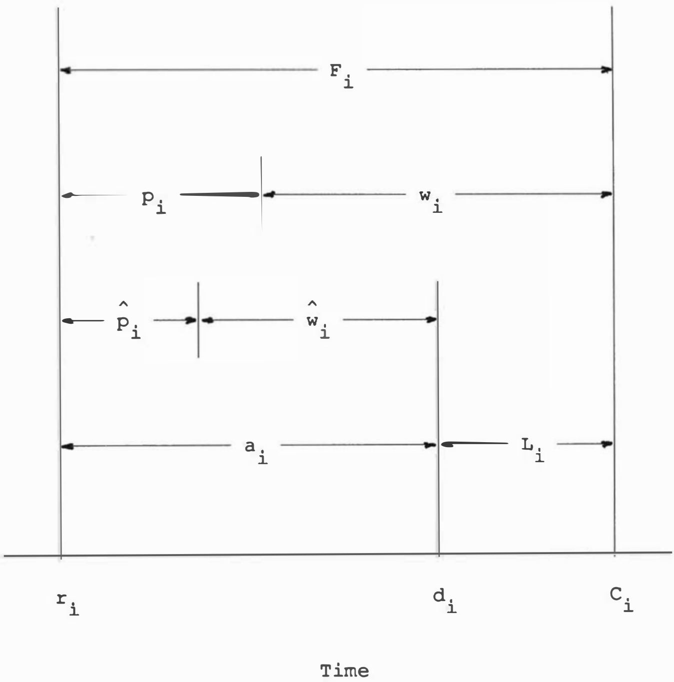
Assumptions

As is typical in scheduling research, this study makes a number of simplifying assumptions. They are:

1. Jobs consist of operations in series.
2. The order in which operations of a job must be performed may not be altered after a job's arrival at the shop. Consecutive operations on the same machine are not permitted.
3. Labor and other resources are in ample supply. Therefore, this is a "machine constrained" shop.
4. Setup times are sequence-independent and are included in the processing times.
5. Jobs move instantaneously between machines.
6. Arrivals are based on a Poisson stochastic process and processing times are based on a negative exponential stochastic process.
7. Each machine is continuously available for assignment, and no breakdowns occur.
8. A machine can process only one operation at a time.
9. A given operation can be performed only on the single specified machine in the shop.
10. Operation preemption is not allowed; once a given operation is started, processing on that operation may not be interrupted until completion.

Figure 1.1

Relationships Among Various
Job Characteristics



11. Operation overlapping is not permitted; processing of a given job operation may not begin until all previous operations in that job are completed.

12. No machine will purposely incur idle time if a job is waiting to be processed.

Most job shop research assumes that processing times are known with certainty once a job arrives at the shop. This study examines simulations where this assumption is invoked, as well as where this assumption is relaxed, permitting actual processing times to vary about expected processing times stochastically.

Harris (1965), in a study of a real job shop, concluded that the assumption of Poisson arrival time was unrealistic. However, Elvers (1974) proved that the relative performances of dispatching rules were not sensitive to the nature of the arrival distribution.

Dispatching Rules

This study evaluates six dispatching rules:

1. Earliest Due Date (EDD)
2. Minimum Job Slack (SLACK)
3. Critical Ratio (CR)
4. Earliest Operation Due Date (EOPDD)
5. Minimum Operation Slack (OPSLK), and
6. Operation Critical Ratio (OPCR).

These six dispatching rules were used by Kanet and Hayya (1982) in their study that compared the performances of operation-based dispatching rules to those of job-based dispatching rules.

EDD directs the workstation to select from the queue the job that has the earliest due date. SLACK directs the workstation to select from the queue the job that has the lowest slack. In this context, slack is defined as the time remaining until the due date minus the total remaining processing time required. CR directs the workstation to select from the queue the job which has the lowest critical ratio (CR). To define CR, let a_{ik} represent an allowance specifically assigned to operation k of job i, let n_i represent the total number of operations in job i, and let t represent the current time. Then,

$$CR = \frac{(d_i - t)}{\sum_{k=z}^{n_i} a_{ik}} \tag{1.8}$$

where z is the current operation number. Note that these first three dispatching rules are job-based, in that selections are based on characteristics of each job, and not of each job's imminent operation.

The final three dispatching rules are analogous, respectively, to the first three, but are based on pending operation characteristics as opposed to job characteristics. Let d_{ik} represent the due date of operation k of job i, and let $\hat{p}_{ik}$ represent the processing time expected to be required by operation k of job i. EOPDD directs the workstation to select from the queue the job that has the earliest due date for the pending operation. OPSLK directs the workstation to select from the queue the job that has the smallest $d_{ik} - \hat{p}_{ik}$. OPCR directs the workstation to select from the queue the job that has the minimum [time to the pending operation due date divided by the pending operation allowance], or

$$\mathrm{OPCR} = \frac{(d_{ik} - t)}{a_{ik}} \tag{1.9}$$

Kanet and Hayya (1982) concluded from their research that operation-based rules were superior to job-based dispatching rules. Kanet and Hayya, however, simplistically set their job allowances as multiples of expected total processing times, and allocated those job allowances among operations in proportion to operation processing times; they evaluated no other allowance policies. This study, therefore, serves to extend their research by evaluating different methods of setting job and operation allowances.

The dispatching rules evaluated in this research represent important categories of selection heuristic rules. EDD is the most basic dispatching rule that utilizes due dates. SLACK not only addresses due date, but also accounts for the remaining required processing time. CR has been in the past a popular rule in the real world, and has proven its ability to control lateness variance in earlier studies (Kanet, 1979).

The operation-based analogies to these rules are intuitively appealing because they provide intermediate benchmarks for job progress, and because they have demonstrated (as mentioned) promising research results in certain situations. Melnyk and Vickery (1986) report that the once-popular CR is falling into increasing disuse in the real world in favor of operation-based rules.

Allowance Policies

The various specific allowance policies to be evaluated in this current research fall into two general classes: those utilizing local (i.e., job specific) variables and those utilizing both local and global (i.e., shop congestion) variables.

A local rule defines the total allowance of job i as some function of certain job-specific variables:

$$a_i = f([LV]). \tag{1.10}$$

A global rule defines the total allowance of job i as some function of job-specific and global variables:

$$a_i = f([LV], [GV]). \tag{1.11}$$

To further define [LV] and [GV], let mik represent the number of the machine required to process operation k of job i. Then, for a given job i over all operations k, examples of local variables are n_i and p_{ik}. Examples of global variables are $TJIQ_{mik}$, $TWIQ_{mik}$, $TOIS_{mik}$, and $TWIS_{mik}$, where (as of the arrival of job i at the shop) $TJIQ_{mik}$ represents the length of the existing queue at machine mik, $TWIQ_{mik}$ represents the total processing time of operations in the queue at machine mik, $TOIS_{mik}$ represents the number of remaining operations elsewhere in the shop that require machine mik, and $TWIS_{mik}$ represents the total processing time of remaining operations elsewhere in the shop that require machine mik. Note that the final equation for each allowance policy may include transformations of the raw variables.

In keeping with the assumption of no external allowance constraints, a unique allowance policy is defined for each combination of dispatching rule/allowance policy class/shop characteristic. Of special interest is a comparison of the performance of the best global rule for a combination to that of the best corresponding local rule.

The specific coefficients for each allowance policy are derived based on multiple regression analyses of pilot simulations. The specific procedure for deriving the equations will be explained in detail in a later section.

Shop Characteristics

Evaluations are conducted at two different levels of expected shop utilization (75% and 90%) and under two different assumptions concerning actual operation processing times (actual processing times assumed equal to expected processing times, and actual processing times allowed to vary about expected processing times). Shop utilization is defined as the percentage of available machine time that is not idle. Simulating under four different shop environments serves to demonstrate the sensitivity of allowance policies and dispatching rules to variations in assumptions, and to increase the value of the research results to real world job shops.

Several studies have assumed a 90% expected utilization level (for examples, Conway, 1965b, and Kanet, 1979). This current research also evaluates performances at a 75% expected utilization level.

As previously mentioned, the vast majority of simulation research has assumed that actual operation processing times exactly equal expected operation processing times. In short, actual processing times are known with certainty upon a job's arrival at the shop. This assumption has served to eliminate one source of random variation in order to permit a clearer evaluation of experimental relationships.

This current research also evaluates performances under an environment where both the expected processing time for each operation <u>and</u> the actual processing time for an operation (given its expected processing time) are governed by stochastic processes.

Hypotheses

1. Allowance policies defined from an iterative simulation-regression procedure produce lower completion inaccuracy penalties than those defined from a single pilot simulation.

As previously discussed, the use of a single pilot simulation to define an allowance policy for use in a subsequent evaluatory simulation may give inferior results. General job stream characteristics may differ between the two simulations due to the interaction of each allowance policy with any dispatching rule using due dates in its selection process. Repeating the simulation-regression procedure until a stable allowance policy is approached (to be discussed later in further detail) addresses this source of inaccuracy.

2. Estimating operation allowances directly from defined allowance policies produces lower completion inaccuracy penalties than proportionally allocating total job allowances among operations.

Every study reviewed that addressed operation-based dispatching rules defined operation due dates by allocating a job's estimated total allowance among operations; the majority of those studies allocated in proportion to operation processing times. A properly defined allowance policy should be able to directly estimate cumulative operation allowances (and therefore due dates) with less inaccuracy.

3. Allowance procedures that utilize global information produce lower completion inaccuracy penalties than those that use only local information.

This hypothesis is based on indications from previous research (Ragatz and Mabert, 1984; Weeks, 1979; and Weeks and Fryer, 1977), as well as the common sense notion that a job arriving at a crowded shop will tend to spend more time in the shop than one that arrives at an empty shop.

4. Operation-based dispatching rules produce lower completion inaccuracy penalties than job-based dispatching rules.

Operation-based dispatching rules address job progress in relation to a series of intermediate objectives. Existing research (Kanet and Hayya, 1982) supports the proposition that completion accuracy should improve by evaluating job status in relation to near-term goals (task due dates) as opposed to a longer-term goal (job due date).

5. The lower the expected shop utilization is, the lower the incremental benefits of incorporating global information are.

Shop congestion is higher in an environment of high utilization than one of low utilization. As queue lengths increase, the effects of shop characteristics on job progress should increase, as should the benefits of directly addressing shop status information in estimating allowances.

6. Simulations run under conditions of stochastic actual operation processing times produce higher completion inaccuracy penalties than simulations run under conditions of deterministic actual processing times.

The stochastic variation of actual processing times about expected processing times is a source of variation that cannot be captured in an allowance estimator. This additional unexplained variation should decrease completion accuracy directly.

Summary

This research examines the effects of six dispatching rules and two classes (local and global) of internally set allowance policies on job completion inaccuracy penalties in a dynamic job shop environment. The procedures by which allowance policies are analytically derived, the evaluation of completion inaccuracy cost as the objective function, and the investigation of the robustness of allowance procedures differentiate this study from existing research literature. The hypotheses tested directly address existing needs of real-world shops.

Chapter 2 represents a review of existing research literature. Chapter 3 discusses the simulation structure and the experimental design of the study. Chapters 4 and 5 present analyses of research results. Chapter 6 discusses managerial implications of the research.

Chapter 2

Review of Related Research

This review proceeds according to the classification displayed in Figure 2.1. Examples of non-dispatching rule research are offered, followed by a review of analytically-based dispatching rule research. Simulation-based dispatching rule research is reviewed next in the context of two categories: studies that assume a single allowance policy, and studies that evaluate multiple allowance policies. Finally, studies that evaluate the use of global information in setting allowances are discussed.

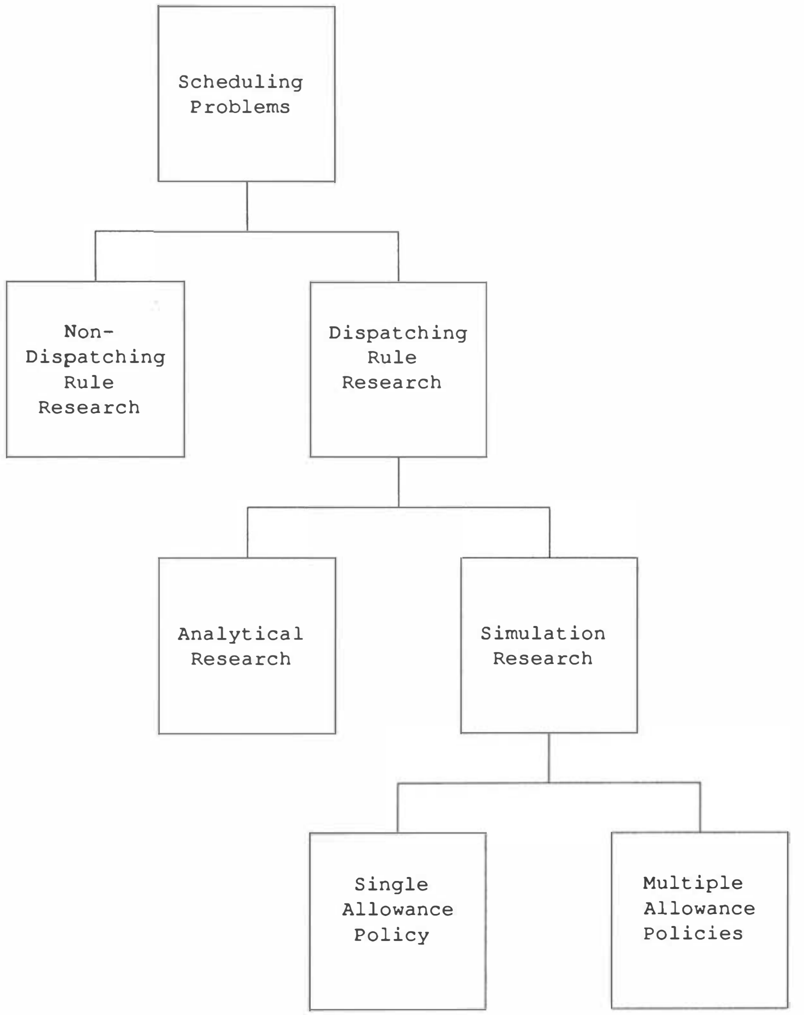
This review concentrates on dispatching rules and allowance policies that directly pertain to this research. A more detailed review may be obtained from survey papers of scheduling research by Day and Hottenstein (1970), Elmaghraby (1968), Gonzalez (1977), Lemoine (1977), Moore and Wilson (1967), Panwalker and Iskander (1977), and Salvador (1978). Baker (1974), Coffman (1976), and Conway, Maxwell, and Miller (1967) wrote books devoted to the general scheduling problem.

Non-Dispatching Rule Research

A number of analytical approaches to scheduling by methods other than dispatching rules exist. Baker and Schrage (1978) and Srinivasan (1971) adapted dynamic programming to the scheduling of a one-machine, static arrival shop to minimize tardiness. Rothkopf (1966) used similar

Figure 2.1

A Classification of
Scheduling Problem Research



techniques to minimize makespan in the context of a parallel shop with static arrivals.

Story and Wagner (1963) used integer programming techniques in static flow shop research. Manne (1960) and Fisher (1973) extended the use of this procedure to a static job shop environment.

Fisher (1976), Picard and Queyranne (1978), and Shwimer (1972) used branch and bound techniques to minimize tardiness in the context of a one-machine static shop. Similar algorithms were adapted to a static flow shop by Ignall and Schrage (1965), and to a static job shop by Brooks and White (1965) and Balas (1969).

Shild and Fredman (1962) used branch and bound techniques to evaluate a lateness objective that was quadratic for positive L_i and zero for negative L_i. Their study proved that knowledge of d_i and p_i are insufficient to determine the relative positions of two jobs in an optimal schedule. Dispatching rules based solely on these two values, therefore, cannot be developed to minimize this form of lateness objective.

Johnson (1954) developed an important algorithm to minimize makespan in a two-machine static flow shop. The intuitive interpretation of the algorithm is as follows:

1. Put the smallest p_{i1} first in the schedule so the second machine can begin processing as soon as possible.

2. Put the smallest p_{i2} last in the schedule so that total processing can be completed as soon as possible after machine 1 is finished. In the case of a single job having both the smallest p_{i1} and the smallest p_{i2}, assign it to the machine that corresponds to the smaller processing time of the two.

3. Repeat the first two steps until all jobs are scheduled.

Johnson's two-machine algorithm was the foundation of research by Burns and Rooker (1978), Jackson (1956), and Giglio and Wagner (1964), and provided the basis for heuristic procedures by Campbell (1970) and Dannenbring (1977).

Dispatching Rule Research

While some experimental studies have been performed in the context of real job shops, the vast majority of dispatching rule research can be classified as either analytically-based or simulation-based. Examples of real shop experimental research are the studies done by Elmaghraby and Cole (1963) at Western Electric and Bulkin (1966) at Hughes Aircraft.

Analytically-Based Research

Analytically-based research typically concerns one-machine static shops with regular performance criteria (those that can increase only if at least one job flow time increases). The four accuracy criteria in this research are non-regular. Certain analytical results, however, do provide insight into the performance of various dispatching rules in a general context.

Smith (1956) proved that, in a one-machine static shop, the SPT dispatching rule minimizes mean flowtime. Conway et. al. (1967) extended this proof to show that SPT also minimizes mean lateness. Further, Conway et. al. (1967) proved that the rule LPT (i.e., choose the operation with the largest processing time) maximizes mean lateness.

Smith (1956) demonstrated that the dispatching rule EDD minimizes the maximum positive job lateness. This proof supports simulation results that have shown EDD producing a consistently lower lateness variance than SPT. Conway *et. al.* (1967) showed that the dispatching rule SLACK (see glossary) maximizes the minimum job lateness. This proof supports simulation results that have shown SLACK reducing lateness variance by compressing the spread of lateness from below.

Several studies offer important interpretations of job lateness as an incurred penalty. Smith (1956) and McNaughton (1959) showed that, in the context of a one-machine static shop, if all jobs are late and the cost of lateness is linear with a slope of e_i, total cost is minimized by sequencing according to the minimum value of e_i/p_i. Fife (1965) extended this result to the case of dynamic arrivals following a Poisson process. Kanet (1979) used McNaughton's analytical proof as a foundation for the dispatching rule OPSLK/P (see glossary).

Sidney (1977) conducted the only analytical research found that concerned a non-regular performance objective. He developed a simple algorithm to minimize the maximum penalty for jobs that either start early or finish late. The severe assumptions made, however, limit the value of Sidney's work to this current research.

An important factor in the usefulness of analytically-based techniques is whether a given problem is "P-complete" or "NP-complete." A problem is P-complete if its solution time is bounded from above by a polynomial function; otherwise, the problem is NP-complete. Generally, NP-complete problems rely on approximation techniques such as dispatching rules.

Lenstra *et. al.* (1977) demonstrated that the general job shop scheduling problem is NP-complete, and that any scheduling problem with a tardiness criterion is NP-complete, even in a one-machine static context. Rinnooy Kan and Lenstra (1975) proved NP-completeness in minimizing makespan in a multi-machine parallel shop, and the flowtime problem to minimize makespan was shown to be NP-complete by Garey (1976).

Simulation-Based Research

Virtually no analytical results exist for lateness related criteria when the number of machines in a shop is greater than one (Kanet, 1979). In these cases, researchers typically use simulation techniques.

Many simulation studies have evaluated the tendencies of various dispatching rules. Panwalker and Iskander (1977) surveyed over 100 different dispatching rules from the literature. This current review concentrates on simulation research concerning dispatching rules that operate in the context of some lateness related objective criterion.

The SPT rule long occupied the position of the "standard" in research due to its ability to minimize mean flowtime and mean lateness (Nanot, 1963; Conway and Maxwell, 1962; Conway et. al., 1967). Unfortunately, the same studies that established SPT as the champion of mean flowtime also demonstrated that it produces extremely high lateness variances, due to the fact that jobs with large operation times may be continually "bumped" in a queue. Research by Conway and Maxwell (1962) showed that when SPT is altered in an attempt to prevent such large variances (either truncated or altered with another rule), the rule loses its advantages faster than its disadvantages.

Conway (1965a) examined seven dispatching rules: OPNDD, EDD, SPT, LPT, FCFS, SLACK, and S/OPN (see glossary). Conway set the mean allowance in this study at nine times the mean processing time. S/OPN produced the lowest lateness variance; Le Grande (1963) and Carroll (1965) reported similar results.

OPNDD produced an unexpectedly large lateness variance in Conway's study. A subsequent evaluation of OPNDD by Kanet (1979) did not reproduce this phenomenon.

New (1975) examined several dispatching rules with the mean allowance set at five times the average job processing time. This study showed that the dispatching rule OPSLK provided good control of lateness variance.

Putnam <u>et</u>. <u>al</u>. (1971) and Berry and Rao (1975) recommended the dispatching rule CR as an attractive alternative to S/OPN. CR has been used widely in industry (Kanet, 1979) due to its ability to control lateness variance.

Few studies specifically addressed MSL as an objective criterion to be minimized. Kanet (1979) evaluated several dispatching rules in terms of their ability to control MSL, mean absolute lateness, and maximum absolute lateness. All three criteria are non-regular, all assume that positive penalties are incurred for early job completions as well as late job completions, and all are logical measures of due date accuracy. The dispatching rule OPSLK/P produced the lowest MSL of the twelve rules evaluated. Kanet's study evaluated three different levels of mean allowances.

Few studies in the early literature evaluated different allowance policies. Research typically determined job allowances as constant multiples of processing times.

Conway (1965a and 1965b) demonstrated that performances of various dispatching rules were significantly affected by the choice of allowance policy. These studies examined four different policies:

1. CON
2. RDM
3. TWK
4. NOP

CON assigned a constant allowance to each job. RDM assigned a random allowance to each job, reflecting an environment where external forces strictly set due dates. TWK assigned each job an allowance that was a multiple of the job's processing time, and NOP assigned each job an allowance that was a multiple of the number of operations in the job. Conway set the mean allowance for each policy at nine times the average job processing time.

Elvers (1973) examined shop performance using the allowance policy TWK for several different levels of mean allowance. The results showed that varying the multiples affected the relative performances of different dispatching rules.

Kanet (1979) examined the effects of five allowance policies on the relative MSL of dispatching rules. In addition to the TWK, CON, and NOP policies evaluated by Conway, Kanet evaluated PPW (each job allowance equalled the job processing time plus a multiple of the number of operations) and PPWN (each job allowance equalled the job processing time plus a quadratic function of the number of operations). Kanet imposed, as did Elvers, various levels of mean allowance on each policy. His study, confirming the results of Elvers and Conway, showed that allowance policy and mean allowance level affected the relative performances of dispatching rules. Kanet recommended PPW as the best policy to minimize MSL. Few studies in the literature evaluated allowance policies that were totally free from external constraints.

Until the mid 1970's, investigations of multiple due date policies were limited to performance comparisons among simplistic job-oriented (local) allowance procedures with arbitrarily set mean allowance constraints. For examples, see Conway (1965a and 1965b) and Eilon and Hodgson (1967).

The earliest study found that took an innovative and promising approach to due date determination was by Eilon and Chowdhury (1976). They not only investigated different forms of job-related information in the allowance procedure (such as raising total job processing time to a power), but also proposed the incorporation of shop-related information in the form of queue lengths at required machines. They concluded that including shop workload considerations in the allowance procedure was often advantageous.

Weeks (1979) extended the concept of incorporating shop congestion information in allowance procedures to a dual (machine and labor) constrained shop. He reflected shop congestion in an expected delay time calculation which was based largely on queueing theory. The specific form of the calculation is not given here because it was later shown to perform poorly (Ragatz and Mabert, 1984).

Baker and Bertrand (1981) investigated the modification, based on a shop congestion index, of three simplistic allowance procedures. The three procedures were CON (a constant total allowance for each job), SLK (a constant waiting allowance for each job), and the popular TWK (total allowance for each job equal to a multiple of total processing time). The modification was based on the ratio of total processing time in the shop to the average total processing time. Their research supported the conclusion of Eilon and Chowdhury (1976) that incorporating congestion data is often advantageous. Their findings were limited, however, by the fact that they only examined the dispatching rules of SPT and EDD, and purposely constrained their allowance policies to very simple forms.

Bookbinder and Noor (1985) proposed an allowance policy that incorporated both job and shop related information, but performed their evaluations in the context of a one-machine shop to minimize the regular objective function "percent tardiness."

Another innovative approach to allowance policies was investigated by Baker and Kanet (1983) and Baker (1984). Although basic allowance policies utilized only job related information, these studies proposed a "modified due date" that was defined as the original due date or the early finish time, whichever was larger. They concluded that the modified due date (both in a job and operation context) performed well under a variety of mean allowance and mean shop utilization levels.

As will be discussed in more detail in a later section, this current research analyzes data in a factorial design by using regression analysis on a set of dummy variables. This same basic analytical approach was used by Weeks and Fryer (1977) in the context of a dual constrained shop with TWK-oriented allowance policies. Concerning behavior of residuals, they concluded that the residuals were not markedly non-normal, and that the effects of heteroscedasticity and autocorrelation (though present to a degree) were not significant enough to invalidate standard regression inferences. As will be discussed further in Chapter 5, analyses of the residuals from the evaluatory regression analyses in this current research support these conclusions by Weeks and Fryer.

While several of the reviewed studies touched on isolated concepts related to the current research, the study by Ragatz and Mabert (1984) came closest in intent by drawing together several key concepts that are investigated in the current study. A fairly detailed critique of their work was given in Chapter 1, and will not be repeated here.

Summary

The complexity of the general job shop scheduling problem has limited both the amount and real-world applicability of analytical scheduling research. The cited analytical studies, however, provide insights into and support for less rigorous approaches to the problem.

The majority of shop scheduling research has used computer simulation techniques to evaluate characteristics of various queue dispatching rules. A few of the more recent studies have addressed the potential benefits of varied allowance policies on the minimization of completion inaccuracy costs.

Chapter 3

Research Method

The first section of this chapter details the research design, explaining the three phases of the research, the simulation structure, and the two stages of the data generation. The second section of this chapter discusses the data analysis procedures.

Design

Research Phases

This study entails the analyses of data in the form of three factorial design matrices, reflecting three distinct phases of the research. The rows in each matrix correspond to dispatching rules under consideration. The two columns of the first matrix reflect the general procedural alternatives of determining specific allowance equations based on a single pilot simulation vs. based on an iterative simulation-regression procedure. The two columns of the second matrix reflect the general procedural alternatives of setting operation due dates by allocating the total job allowance among operations vs. estimating operation allowances directly. Both sets of procedural alternatives are discussed in further detail later in this chapter. The eight columns of the third (and largest) matrix reflect combinations of allowance policy class (local or global), utilization level (75% or 90%), and actual processing time assumption (equal to expected processing time or allowed to vary stochastically about expected processing time).

As each matrix is addressed separately on the basis of each of the four defined penalty functions (VAR, MSL, MAL, and SQL), data in the form of twelve final matrices are analyzed. Each cell contains twenty observations of the appropriate inaccuracy penalty measure, generated by twenty shop simulations using twenty different job streams.

In each simulation, data is gathered on 1000 completed jobs, providing 20,000 completed jobs per cell. This simulation size is large in relation to the majority of past research of this type. In the simulation studies surveyed by Panwalker and Iskander (1977), for example, simulation sizes varied from less than 100 jobs to 8700 jobs per cell.

The same twenty job streams, altered only as dictated by the appropriate cell environment, are used in every cell. Therefore, the twenty observations in each cell are logically matched with the twenty observations, respectively, in every other cell. The matched nature of the data points among cells represents a variance reduction technique that increases the power of subsequent data analyses over pooled techniques (Ragatz and Mabert, 1984).

The potential problems with basing allowance equation forms and coefficients on a single pilot simulation have been discussed previously. Phase 1 addresses the benefits of using an iterative simulation-regression procedure to determine allowance policies (see Hypothesis 1). As shown in Figure 3.1, the columns in Matrix 1 represent the best global allowance policy based on

Figure 3.1

Phase 1 Experimental Design
Methods of Determining Allowance Equation

	Allowance Equation Based on Single Pilot Simulation	Allowance Equation Based on Iterative Simulation-Regression Procedures
EDD		
SLACK		
CR		
EOPDD		
OPSLK		
OPCR		

1. the single simulation-regression procedure (as used, for example, by Ragatz and Mabert, 1984), and

2. the iterative simulation-regression procedure proposed in this study.

A brief discussion will clarify the mechanics of the simulation-regression procedures. A specific dispatching rule and general form of the allowance equation are selected (the general forms used in this research for local and global policies are given later in the chapter as Equations 3.2 and 3.3, respectively). An initial simulation is run using the arbitrary allowance policy of job allowance set equal to a fixed multiple of the total required processing time of the job. The multiple used in this research under the 90% expected utilization assumption is six, and the multiple used under the 75% expected utilization assumption is four. These multiples were determined from pilot simulations as the approximate ratios of mean job flowtime to mean total processing time under the respective utilization levels in this shop.

The results of that simulation are analyzed by a multiple regression procedure to produce the specific coefficients in the allowance form chosen. The resulting specific equation represents the "best" estimated allowance policy after one cycle; it is the product of the single simulation-regression procedure. This would be the policy used in subsequent evaluatory simulations in previous non-iterative research, such as that by Ragatz and Mabert (1984).

The iterative simulation-regression procedure proposed in this research, however, continues by running a second simulation using the same job stream and the allowance policy generated from the first cycle above. The results of the second simulation are analyzed by multiple regression to produce a second-cycle specific allowance equation. This procedure is continued through several subsequent cycles.

The simulations performed in Phase 1 are conducted under the assumptions of 90% utilization, operation due dates set by allocating a total job allowance to individual operations in proportion to the operation processing times, and actual processing times equal to expected processing times. The utilization target is based on past research such as Conway (1965a), Eilon and Chowdhury (1976), Weeks (1977), and Kanet (1979). The proportioning of job allowances among operations is a standard assumption in shop research, and can be seen in Conway (1965a and 1965b), Kanet (1979), and Ragatz and Mabert (1984). The assumption that actual processing times are equal to expected processing times is invoked in virtually all previous research of this type, a rare exception being Eilon and Hodgson (1967).

As is discussed in detail in Chapter 4, analyses of Phase 1 data indicate that while the iterative procedure produces major gains in accuracy in simulations that use global allowance policies, there are no significant benefits produced under environments where local allowance policies are used. Therefore, the iterative procedure is used in both subsequent phases with the exception of cells that dictate local allowance policies.

Phase 2 addresses the question of whether to set operational due dates by

1. allocating total job allowances in proportion to operation processing times (the standard method in existing research), or

2. estimating operation allowances directly by the appropriate generated allowance equation, as proposed in this study (see Hypothesis 2).

Figure 3.2 displays the structure of Matrix 2. Note that, since Phase 2 concerns operation allowances, there are only three rows in the matrix (representing the three operation-based dispatching rules).

The remainder of the Hypotheses are addressed in Phase 3. The structure of Matrix 3 is displayed in Figure 3.3. As is discussed in detail in Chapter 4, analyses of Phase 2 data indicate that significant improvements in accuracy are obtained by directly estimating operation due dates. Therefore, this method is used exclusively in Phase 3 simulations.

In order to provide a direct link between this study and that of Ragatz and Mabert (1984), data are generated for additional cells. The cell environment in the current research that specifically parallels a cell environment in the Ragatz and Mabert study entails the dispatching rule SLACK, the form of allowance estimator termed RMR in their study, an expected utilization of 90%, and known actual processing times (i.e., actual processing times assumed equal to expected processing times). The specific form of RMR is defined by Ragatz and Mabert as

$$a_i = k_1 P_i + k_2(JIS_i) + k_3(JIQ_i) + k_4(WIQ1_i) + k_5(WIQ2_i) + k_6(WIQ3_i), \qquad (3.1)$$

where:

a_i = estimated allowance for job i

P_i = total required processing time for job i

JIS_i = number of jobs in the system

Figure 3.2

Phase 2 Experimental Design
Methods of Setting Operation Allowances

	Total Allowance Allocated in Proportion to Operation Processing Times	Operation Allowance Estimated Directly from Allowance Equation
EOPDD		
OPSLK		
OPCR		

Figure 3.3

Phase 3 Experimental Design
Dispatching/Allowance/Shop Condition Evaluations

	Local Allowance				Global Allowance			
	75% Expected Utilization		90% Expected Utilization		75% Expected Utilization		90% Expected Utilization	
	Act TPT Known	Act TPT Not Known	Act TPT Known	Act TPT Not Known	Act TPT Known	Act TPT Not Known	Act TPT Known	Act TPT Not Known
EDD								
SLACK								
CR								
EOPDD								
OPSLK								
OPCR								

JIQ_i = number of jobs in queue on routing for job i

$WIQ1_i$ = total required processing time of operations in queue at the first machine on routing for job i

$WIQ2_i$ = total required processing time of operations in queue at the second machine on routing for job i

$WIQ3_i$ = total required processing time of operations in queue at the third machine on routing for job i

In order to extend the comparison of this research to that of Ragatz and Mabert, relative performance of the SLACK/RMR combination is evaluated under environments of both 90% and 75% utilization, both known and unknown actual processing times, and allowance coefficients based on both a single pilot simulation and an iterative simulation-regression procedure. Comparisons of the data in these eight cells to the data in the corresponding cells of Matrix 3 provide direct evidence as to the additional benefits inherent in the procedures proposed in this current research as opposed to past procedures.

In all, 296 final cells of data are analyzed. Each of the four penalty measures entail twelve cells in Phase 1, six cells in Phase 2, 48 cells in Phase 3, and the eight isolated cells just discussed. The specific procedures for generating the cell data are discussed in more detail later.

Simulation Structure

The shop consists of eight machines. This number of machines has been used in previous studies (for example, Kanet, 1979). Baker and Dzielinski (1960) concluded that the number of machines in a shop simulation does not significantly affect aggregate performance measures, and that a shop consisting of eight machines adequately represents performance characteristics of much larger shops.

Each job consists of from one to six operations, determined randomly according to a uniform probability distribution. Each operation is assigned randomly to one of the eight machines according to a uniform probability distribution, under the constraint that no two successive operations require the same machine (this constraint is traditional in shop research). Although non-uniform distributions have been used occasionally to establish the number of operations and machine assignments (for example, see Elvers, 1974), the use of uniform probability distributions for these purposes is traditional in job shop research.

Referring to the factorial design in Phase 3 of this research, recall that half of the cells assume that actual processing times are equal to expected processing times, and are therefore known with certainty upon a job's arrival at the shop. The other half of the cells in Matrix 3 allow the actual processing times to vary about expected processing times stochastically.

To maintain direct comparability of data from corresponding simulations among all cells, the actual processing time for any given operation of any given job in any given job stream is made consistent over all cells regardless of the processing time assumption (i.e., known or unknown as of job arrival) and utilization level (i.e., 75% or 90%). This leads directly to two specific job stream characteristics:

1. The expected processing times in any given job stream under the assumption of known actual processing times may differ from the expected processing times of that job stream under the assumption of unknown actual processing times, and

2. The mean time between job arrivals in any given job stream under the assumption of 75% expected utilization differs from the mean time between job arrivals in that job stream under the assumption of 90% utilization.

Referring to the first characteristic above, in cells where actual operation processing times are assumed to vary about expected operation processing times, the expected times are generated randomly from a negative exponential probability distribution. The resulting times are integerized by setting (0,1] = 1, (1,2] = 2, and so on, in the interest of computer run time and to facilitate interpretability of shop processes. The ramifications of integerization (as well as a specific discussion as to the expected values of assumed stochastic distributions) are discussed further in the next section.

Deviations about each expected operation processing time are generated from a second negative exponential distribution that has a standard deviation equal to .3 times the standard deviation of the expected operation processing time distribution, shifted so that the expected value of the deviational distribution is zero. The actual operation processing time, then, is the sum of these two distributions, integerized as above. In cases where the integerized sum is less than one, it is set to one.

In the cells where the actual operation processing times are assumed known (i.e., equal to the expected processing times), the expected processing times for each operation are then reset to equal the actual processing times generated as above. Thus, the actual processing time for any given operation (of any given job within any given job stream) is consistent throughout the Phase 3 design, being the result of stochastic variation about the expected time in the cells that make that assumption.

In summary, for each operation an expected processing time is generated. Based on the expected processing time for each operation, an actual processing time for each operation is generated. In cells where the expected processing time for an operation is assumed equal to the actual processing time for that operation, the expected operation processing time is then set equal to the actual processing time.

Times between job arrivals are generated from a negative exponential distribution, integerized as discussed above. Therefore, job arrivals follow (approximately) a Poisson process, slightly modified by the integerization procedure.

The mean of this exponential distribution is the value, determined from pilot simulations using the "neutral" dispatching rule FCFS, that produces the appropriate machine utilization (75% or 90%) given the correspondingly consistent set of actual operation processing times discussed previously. This method of fixing one stochastic parameter and varying another until a target utilization is achieved has been used in previous research (for example, Kanet, 1979 and Conway, 1965a).

Data Generation Procedures

There are two stages to the data generation:

1. The determination of a specific allowance equation for each cell, and
2. Twenty evaluatory simulations per cell based on the appropriate allowance equation determined in Stage 1.

As will be discussed later in further detail, specific allowance equation determinations in the first stage are based on a single pilot simulation-regression analysis in cells where local allowance policies are used, and are based on an iterative simulation-regression procedure in cells where global allowance policies are used.

All computing was performed on a UNIX-based VAX 8650 mainframe. All simulation and supporting programs were written specifically for this research, by the author, in FORTRAN. Appendix B shows the summary logic flow and the FORTRAN code for the main simulation program. Statistical analyses were performed using the BMDP Statistical Software Package.

Random numbers were generated by a multiplicative congruential method. This method has been shown to possess favorable statistical properties (Naylor et. al., 1966), and is widely used in scientific software packages (for example, the IBM Scientific Subroutine Package).

The specific generator used in this research is from the BMDP Statistical Software Package. Characteristics of various job streams produced by this generator have been checked for randomness and underlying stochastic properties by a series of Chi-squared goodness of fit tests, with consistently acceptable results. Appendix C displays the FORTRAN code for this uniform [0,1) random number generator.

As previously mentioned, values generated for times between job arrivals and operation processing times were integerized prior to use. The mean of the pre-integerized negative exponential distribution used to generate expected operation processing times was fixed at 5 units (i.e., an expected service rate of .2 operations per period). The process of integerization increased the mean service time to 5.5, or an expected rate of .181 operations per period. The summation of the expected processing times with the deviational values (as previously discussed) tended to offset that effect somewhat, and the true mean of the integerized processing times ended up as approximately 5.2 units.

Given the fixed integerized processing time distribution, the desired means for the negative exponential distributions that generated the times between job arrivals were determined as the values which (in pilot simulations) yielded 75% utilization and 90% utilization, respectively. These means turned out to be 2.5 periods for the 75% utilization environment and 2.0 periods for the 90% utilization environment, on a pre-integerized basis. Integerization shifted those means to 2.5 periods and 3.0 periods, respectively.

Duplication of this shop without integerization should, of course, use the post-integerized means given above for underlying distributions. The minor shape effects to the underlying theoretical distributions caused by the integerization process (i.e., from smooth to discrete profiles) were not expected to have any significant effects, due to previous research on the insensitivity of shop performance characteristics to changes in underlying distributions (Elvers, 1974). Informal parallel simulations conducted without integerization supported the previous research findings. Table 3.1 displays frequency tables of operation processing times and times between job arrivals for one of the job streams used in this research.

Stage 1. - In order to achieve a steady state prior to data collection, the shop was pre-loaded with the same job set prior to each simulation. This pre-load set was generated from a simulation using the neutral FCFS (first-come-first-served) dispatching rule under conditions

Table 3.1

Frequency Tables of Selected Characteristics of a Typical Stream of 1800 Jobs

Value	Frequency: Number of Tasks Per Job	Frequency: Machine Assignment by Task	Frequency: Times Between Jobs	Frequency: Expected Operation Processing Times	Frequency: Actual Operation Processing Times
1	296	791	751	1124	1679
2	333	780	403	882	691
3	299	791	245	755	619
4	272	790	165	631	568
5	309	807	86	531	468
6	291	755	60	415	376
7		776	33	325	359
8		748	22	296	248
9			13	226	236
10			6	198	176
11			11	174	152
12			1	123	126
13			1	92	92
14			0	84	83
15			1	68	64
16			0	63	53
17			2	40	52
18				40	24
19				26	34
20				28	29
21				22	23
22				20	13
23				12	12
24				10	17
25				14	13
26				14	3
27				2	7
28				5	2
29				2	3
≥30				16	16

of an expected 83% machine utilization. The FCFS rule was used to generate the pre-load set due to its lack of direct dependence on any job characteristics or stated due dates. Further, no statistics were collected on the first 300 jobs in each simulation; Kanet (1979) used this same cutoff point. Data was collected on jobs 301 through 1300, inclusive. Each simulation continued until all of the jobs in this window were completed.

The assumption that allowance policies are free from external constraints is crucial to the significance of this study. Given a particular combination of dispatching rule and general allowance structure (under a particular shop environment), the shop is free to choose the optimal specific allowance equation based on steady state performance characteristics of the shop. The sole purpose of this first stage of data generation is the determination of that specific allowance equation for each cell of each matrix.

Extensive evaluation of various forms of allowance determination equations yielded forms of a local allowance estimator and a global allowance estimator that, in general, produce optimal or near optimal accuracy over all cells. The form of the local allowance equation is

$$a_i = \alpha + \beta_1(TPT_i) + \beta_2(NOP_i) + \beta_3(TPT_i^2) + \beta_4(NOP_i^2), \qquad (3.2)$$

where: a_i = estimated allowance for job i

TPT_i = total required estimated processing time for job i

NOP_i = number of operations in job i.

The form of the allowance equation for cells that allow the due date determination procedure to incorporate global variables is

$$a_i = \alpha + \beta_1(TPT_i) + \beta_2(TWIQ_i) + \beta_3(TWISM_i) + \beta_4(TPT_i^2) + \beta_5(TWIQ_i^2), \quad (3.3)$$

where: a_i = estimated allowance for job i

TPT_i = total required estimated processing time for job i

$TWIQ_i$ = total required estimated processing time for operations in queue along the routing of job i

$TWISM_i$ = total required estimated processing time for operations elsewhere in the shop that require machines that are required by job i.

Multiple regression analyses are used to estimate the coefficients in the above equations for each cell. As expected, correlations between variables and their squared terms were high (often above .9), and correlations among other pairs of independent variables were often as high as .5. Since deviations from the necessary independence assumption underlying formal regression analysis therefore exist to some degree in each determination, no standard regression inferences are made based upon these regression procedures. Instead, regression analysis is used here merely as a tool for producing a good allowance estimator equation to be evaluated by further analyses.

Further, in cells where the iterative process is beneficial (i.e., those with global allowance policies), analysis showed that, in general, significant incremental improvements in accuracy were achieved through six cycles, but not thereafter. This is discussed in detail in Chapter 4. The iterative procedure, therefore, is carried through the sixth cycle in all simulations with global allowance policies. As indicated by Phase 1 analyses, no iteration is implemented in simulations that use local allowance policies. The choice of general allowance equation forms and the determination of six cycles as optimal are discussed in detail in Chapter 4.

Stage 1 of the data generation is complete after the specific allowance equation for each cell is determined. Each of the simulations performed in Stage 1 use the same job stream, modified only to the extent required by the particular cell environment. For example, the job stream used for a particular cell in a 90% utilization environment is identical to the job stream used for the corresponding cell in a 75% environment, except that each time between job arrivals is drawn from an integerized negative exponential distribution with a smaller mean (the percentile position in each distribution, however, is identical).

Stage 2. - The purpose of Stage 2 is to generate multiple observations per cell in order to provide indications of performance reliability and to permit valid statistical comparisons of performance among cells. Twenty simulations are run per cell, using twenty different job streams, matched among cells as discussed previously. As in Stage 1, the shop is pre-loaded prior to each simulation, and data is collected only on jobs 301 through 1300, inclusive.

While only four inaccuracy measures per simulation (VAR, MSL, MAL, and SQL) pertain directly to stated hypotheses, other data such as observed machine utilizations and mean latenesses were stored and analyzed. These incidental data provided useful insights into the characteristics of the dispatching rules and allowance policies under particular shop environments.

Simulation Validation

The simulation program was validated by two methods. First, shop status details were examined at each successive time period for a series of jobs with known characteristics. This verified that the jobs arrived and moved through the shop properly and that resulting statistics were accurate.

Further, the shop was recreated using the GPSS simulation programming language by Dr. Richard Redmond of Virginia Commonwealth University. Summary characteristics of FORTRAN and GPSS simulations were compared which verified that similar results were produced for identical shop environments. Table 3.2 displays examples of selected shop characteristic distributions.

Data Analysis

Two important previous studies that examined factorial designs are those by Kanet (1979) and Conway (1965a). Both studies avoided statistical analysis of results due to concerns about non-normality, serial autocorrelation, and other systematic effects. Neither study, however, generated multiple observations of the objective measure per cell with which to perform valid statistical tests.

Although Ragatz and Mabert (1984) did generate multiple, matched observations in different cells, the technique used to compare cells on a pairwise basis was a t-test. As only five observations per cell were generated, the marked non-normality of raw observations within each cell do not support strongly the assumption required for this technique.

Table 3.2

Selected Shop Characteristic Frequency Tables
Twenty Simulations

Machine Utilization	Freq
85%	1
86%	1
87%	2
88%	2
89%	3
90%	5
91%	3
92%	2
93%	1

Maximum Shop Load	Freq
35 - 49	2
50 - 64	5
65 - 79	8
80 - 94	3
95 -109	2

Maximum Queue Length	Freq
9 - 11	7
12 - 14	29
15 - 17	34
18 - 20	25
21 - 23	18
24 - 26	19
27 - 29	5
30 - 32	7
33 - 35	6
36 - 38	5
39 - 41	2
42 - 44	3

Average Queue Length	Freq
3.75 - 4.99	4
5.00 - 6.24	5
6.25 - 7.49	6
7.50 - 8.74	5

Data in Phase 1 and Phase 2 are analyzed by pairwise cell comparisons using the Wilcoxon Signed Rank test. This technique is preferred to its parametric analog, the t-test, due to the extreme positive skew of the data.

The Wilcoxon technique, however, is too limited to adequately address by itself the larger and more complex structure of the matrix in Phase 3 of the research. Although tests directly pertaining to most stated hypotheses would entail a comparison of only two treatments, addressing the significance of overall treatment effects and interactions is necessary to test Hypothesis 5, as well as beneficial in terms of general information.

The 960 observations in Matrix 3, therefore, are analyzed in the context of stated hypotheses by a stepwise multiple regression procedure. The regression uses, as potential independent variables, one scalar variable (mean lateness) and 46 dummy variables representing dispatching rules, allowance policy classes, shop environments, job streams, and interactions. A separate regression analysis is performed on each of the four inaccuracy penalty measures, using ten times the natural logarithm of the appropriate measure (due to favorable residual behavior and scaling) as the dependent variable.

Appendix D displays the specific independent variables made available to the stepwise procedure. These consist of mean lateness, nineteen dummy variables representing job streams, one dummy variable representing allowance policy class, one dummy variable representing utilization level, one dummy variable representing the assumption invoked on actual processing times, five dummy variables representing dispatching rules, eighteen dummy variables representing second order interactions, and one dummy variable representing a third order interaction.

Residual analyses showed that, in general, assumptions underlying regression theory were not significantly violated. Multicolinearity among independent variables was, of course, expected to a degree because of the nature of the independent variables (dummy variable sets). The stepwise procedure, however, served to minimize effects of colinearity.

Conclusions from the regression procedures that pertain to the stated hypotheses were augmented by multiple Wilcoxon Signed Rank Tests. The regression analyses, the residual analyses, and the supporting use of the Wilcoxon procedure are discussed in more detail in Chapter 5.

Summary

For each of 296 cells in various factorial designs, job shop simulations and regression techniques are used to specify near-optimal allowance policy equations. Twenty simulations are run per cell, based on twenty job streams (matched among cells) and the specified allowance equation. Stated hypotheses are tested by use of a stepwise multiple regression procedure and the conclusions verified by use of multiple Wilcoxon Signed Rank tests.

Chapter 4

Phase 1 and Phase 2 Results

The first section of this chapter discusses the determination of the forms of the allowance equations used in the research. The following two sections discuss results from Phase 1 and Phase 2, respectively. The final section provides a chapter summary.

Allowance Equation Forms

The forms of allowance equations used in this research for local and global allowance policies were given in Equations 3.2 and 3.3, respectively. Numerous alternative forms were evaluated, from which these two were selected as being generally optimal.

Potential predictor variables in allowance equations can be based either on amount of processing time involved (for example, total processing time in a particular queue) or on number of operations/jobs (for example, number of jobs in a particular queue). The form selected for local allowance policies (Equation 3.2) includes variables of both types.

The form (Equation 3.3) selected for global allowance policies, however, consists solely of time-related variables. Pilot evaluations indicated that inclusion of variables based on numbers of operations or tasks did not contribute to predictive power.

This indication is not surprising. Analogous variables of the two types largely address the same job or shop characteristics, and are expected to be positively correlated. For example, over all jobs in a simulation, the number of operations per job would be positively correlated with the total processing time per job. In the local allowance policy form, each of the two specific variables just mentioned contribute enough unique information to warrant inclusion. In the global allowance policy form, however, the several included time-related variables (each of which are logically related to the number of operations per job, for instance) combine to explain enough of the variation in variables such as the number of operations per job as to make their inclusion superfluous.

Note that, in the forms selected, characteristics are aggregated over all operations in a job (for example, the variable TPT_i represents the sum of required operation processing times for all operations in job i). Other allowance forms that were evaluated but not selected for use entailed individual (disaggregated) operation characteristics. These alternate forms were more complex and did not appear to contribute to predictive power. An example of such a policy form is:

$$\begin{aligned} a_i = \alpha &+ \beta_1(TPT_{i,1}) + \ldots + \beta_z(TPT_{i,z}) \\ &+ \beta_{z+1}(TWIQ_{i,1}) + \ldots + \beta_{2z}(TWIQ_{i,z}) \\ &+ \beta_{2z+1}(TWISM_{i,1}) + \ldots + \beta_{3z}(TWISM_{i,z}), \end{aligned} \tag{4.1}$$

where there are z operations in job i, and:

a_i = estimated allowance for job i

$TPT_{i,k}$ = processing time required for operation k of job i

$TWIQ_{i,k}$ = total processing time of operations in the queue of the machine required by operation k of job i

$TWISM_{i,k}$ = total processing time of operations elsewhere in the shop that require the machine required by operation k of job i.

Other variables evaluated but not included in the selected forms were total work in the shop, total number of jobs in the shop, and total number of operations in the shop. Also evaluated but not included were interactions such as total work in the shop times the number of operations in a job.

The form of the allowance policy termed RMR in the research of Ragatz and Mabert (1984) was given in Equation 3.1. Note that this form includes both aggregate and operation-specific variables. As is discussed in detail in a later section, the simpler and completely aggregated global form given in Equation 3.3 outperformed the RMR form in terms of completion inaccuracy penalties in this current research.

The early finish time of any operation k of any job i as of its arrival at the shop is defined as

$$EFT_{ik} = r_i + \sum_{z=1}^{k} p_z \tag{4.2}$$

where: EFT_{ik} = the earliest possible finish time for operation k of job i as of the job's arrival at the shop

r_i = the arrival time of job i at the shop

p_z = the required processing time of operation z of job i.

Without prior knowledge of allowance equation coefficients, one cannot rule out the possibility of an estimated due date being earlier than the appropriate early finish time, although with an intelligently set allowance equation this should be a highly improbable occurrence. Therefore, all allowance procedures in this research add the constraint that any due date that is earlier than the appropriate early finish time is set to that early finish time. Note that this is not a dynamic process, and is done only once, upon a job's arrival at the shop.

Phase 1

The first phase of the research addresses Hypothesis 1 and investigates whether an iterative simulation-regression procedure provides lower inaccuracy penalty measures than a single simulation-regression procedure. The assumptions under which Phase 1 simulations were run reflected 90% expected utilization, actual processing times equal to expected processing times, and operation allowances determined by allocating total job allowances in proportion to operation processing times. These assumptions are considered standards of existing job shop research.

As previously discussed, the rationale behind the hypothesized benefits of an iterative process is based on the fact that for a dispatching rule that incorporates the job due date in the selection process, two simulations run under two different allowance policies can produce different sets of general scheduling characteristics. By basing an allowance equation on the results of a single simulation that was run under an arbitrary allowance policy, the shop will not have had an "opportunity to adapt" to the general tendencies of the interaction between the allowance policy and the dispatching rule used. It is hypothesized that an iterative process produces successive allowance policies that tend to converge to stability, resulting in lower completion inaccuracy penalties associated with the eventual policy evaluated.

Convergence to Stability

The theory of convergence was supported by Phase 1 analyses. While convergence did not appear to be a monotonic process, allowance policies produced by successive simulations later in an iterative process tended to be more similar than those produced by successive simulations at the beginning of an iterative process.

Figure 4.1 displays an example of convergence to stability, over the first ten cycles of the iterative procedure, of the global allowance policy form defined in Equation 3.3. This chart presents the sets of standardized regression coefficients generated by successive cycles using the dispatching rule CR, under the standard assumptions previously stated. The standardized regression coefficients tended to change more radically in the first several cycles than thereafter; a measure of stability was apparently achieved after the first six or seven cycles.

Table 4.1 addresses to what extent two iterative sequences, under the same environment but starting with two vastly different initial allowance policies, approach each other. The first column presents the allowance coefficients from the first and ninth cycles, using the standard arbitrary initial allowance policy of 6 times the total processing time required by the job. The second column presents the corresponding coefficient sets from the iterative series initialized with the allowance policy -6 times the total processing time required by the job. This allowance policy, of course, is unrealistic as it assigns negative allowances to jobs. Further, the larger the processing time

Figure 4.1

Standardized Regression Coefficients
Dispatching Rule CR, Cycle 1 - Cycle 10
Global Allowance Policy, 90% Utilization

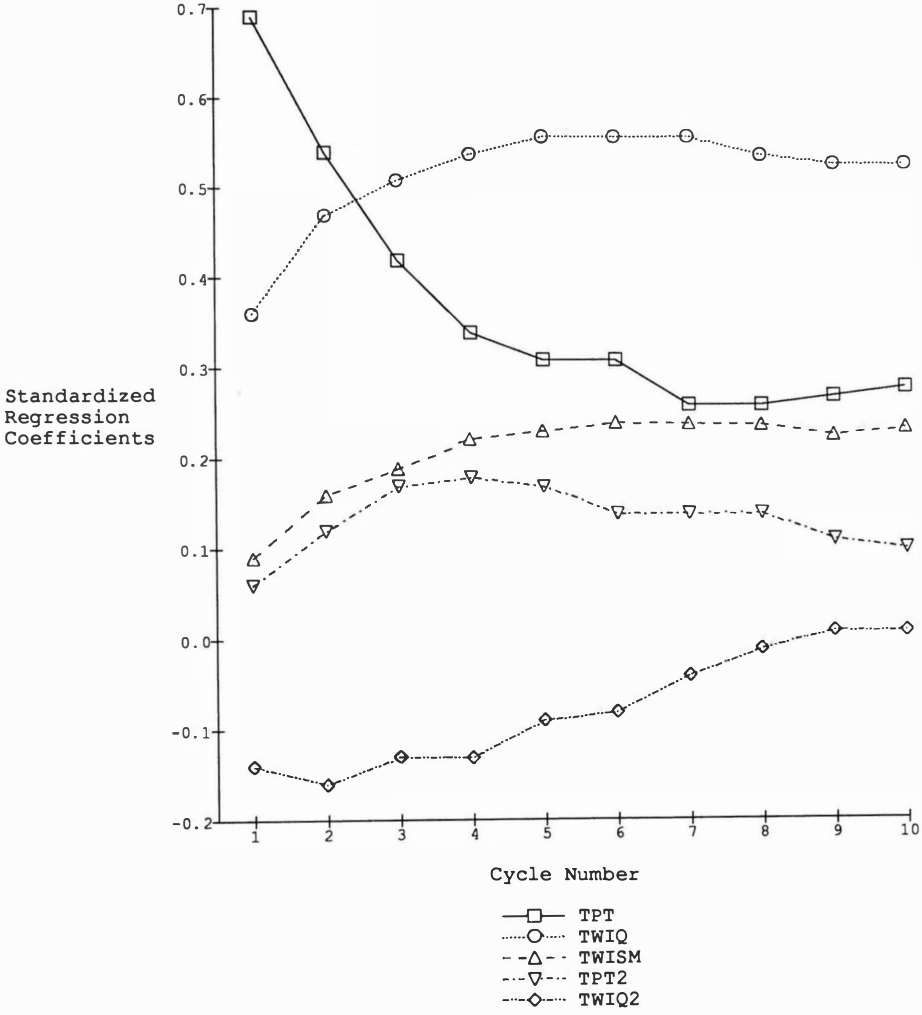


Table 4.1

Allowance Equation Coefficients:
Cycles from Iterations Using Different
Initial Allowance Equations

		Initial Equation	
	Coefficients	6.0*TPT	-6.0*TPT
Cycle 1	Intercept	-25.3612	25.5311
	TPT	3.8743	.1175
	TWIQ	.5001	1.0440
	TWISM	.0283	.0707
	TPT^2	.0298	.0140
	$TWIQ^2$	-.0006	-.0017
	Coefficients	6.0*TPT	-6.0*TPT
Cycle 9	Intercept	-14.0693	-23.7918
	TPT	.5836	.8997
	TWIQ	.1654	.2678
	TWISM	.0956	.1651
	TPT^2	.0711	.0450
	$TWIQ^2$	.0019	.0012

required by a particular job is, the more negative is the allowance assigned to that job. Therefore, the ranking of allowances under this initial policy is exactly the opposite of the ranking of allowances produced by the first initial policy.

One sees that while the coefficient sets from the first cycle differ in several major respects from each other (note especially the intercepts and the TPT coefficients), by the ninth iterative cycle the two coefficient sets have become similar in that the respective coefficients are of the same sign and roughly the same magnitude. These results give indications that the iterative process not only stabilizes allowance equations given an initial policy, but also drives allowance equations toward a common coefficient set regardless of the initial allowance policy used.

Effects of Iteration on Penalty Measures

The fact that an iterative process apparently produces successively more stable allowance equations has little relevance to the stated research problem unless this convergence to stability manifests itself in systematic beneficial effects on resulting inaccuracy penalty measures. The presence and nature of such systematic effects can be evaluated by examining, for each of the four penalty measures addressed in this research, the medians of the measures produced by the twenty evaluatory simulations under each dispatching rule/ allowance policy/ iteration cycle combination, as well as performing pairwise statistical tests of significance using the Wilcoxon Procedure.

Results indicate that the effects of iteration are different for local allowance policy forms than for global allowance policy forms.

Evaluations of iterative effects, therefore, are presented separately for each of these two general classes of allowance policies. All simulations performed in Phase 1 analyses are under the "standard" environment of 90% expected utilization, known actual processing times, and operation allowances set by allocating total job allowances among operations in proportion to the operations' respective required processing times.

Local Allowance Policies. Figures 4.2 through 4.5 present the medians, by cycle and by dispatching rule, of the penalty measures VAR, MSL, MAL, and SQL, respectively. Again, each point represents the median of the appropriate measures produced by twenty evaluatory simulations using local allowance policies.

These charts present little or no compelling evidence of any systematic beneficial effects of iteration on penalty measures using local allowance policy forms. Apparent tendencies range from cyclical movements (for example, with the dispatching rule CR for the measures VAR, MSL, and MAL) to monotonic upward pressures on penalties (for example, with the dispatching rule OPSLK for the measures MSL and MAL).

The impressions given by the charts are supported by pairwise statistical comparisons of the points, which show few statistically significant differences between successive cycles. Table 4.2 displays the significance levels of all pairwise comparisons of the median variances produced by the first six cycles using the dispatching rule EDD. Note that no two successive values are statistically different from each other, based on the Wilcoxon Signed Rank test.

Figure 4.2

Median VAR Penalty Measures
Local Allowance Policy, 90% Utilization

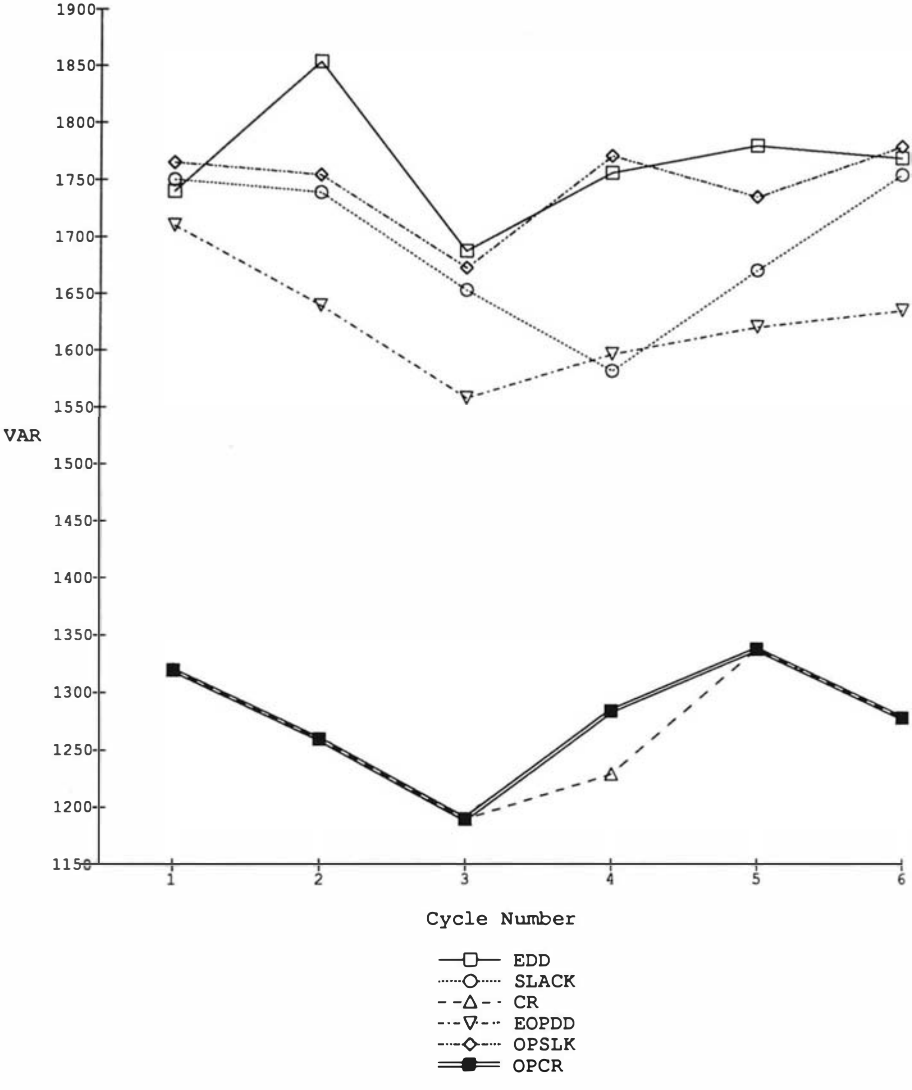


Figure 4.3

Median MSL Penalty Measures
Local Allowance Policy, 90% Utilization

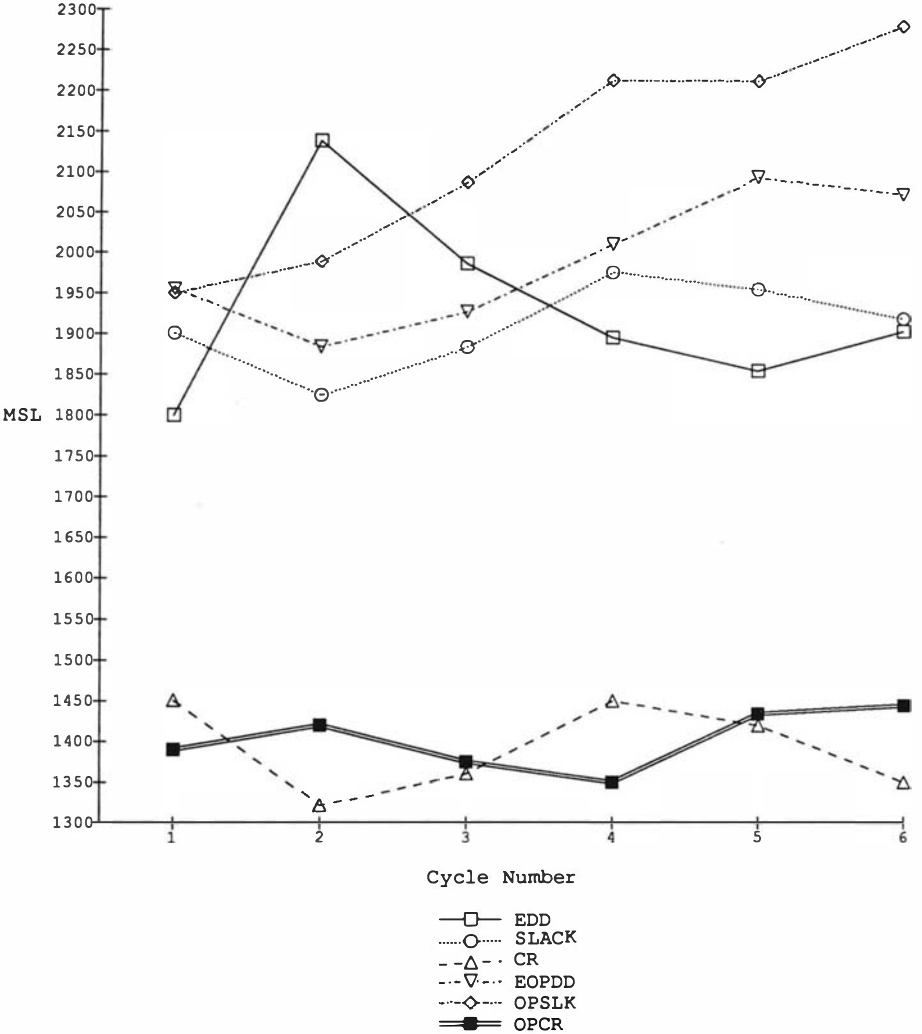


Figure 4.4

Median MAL Penalty Measures
Local Allowance Policy, 90% Utilization

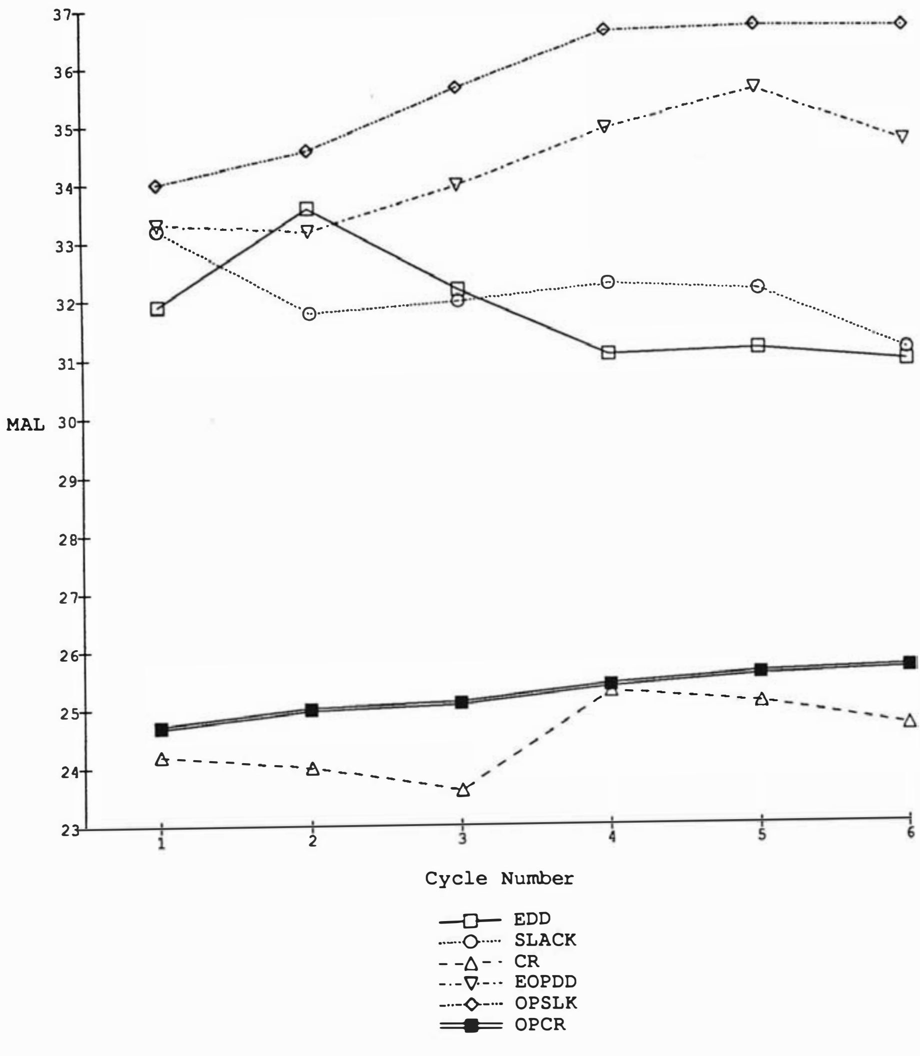


Figure 4.5

Median SQL Measures
Local Allowance Policy, 90% Utilization

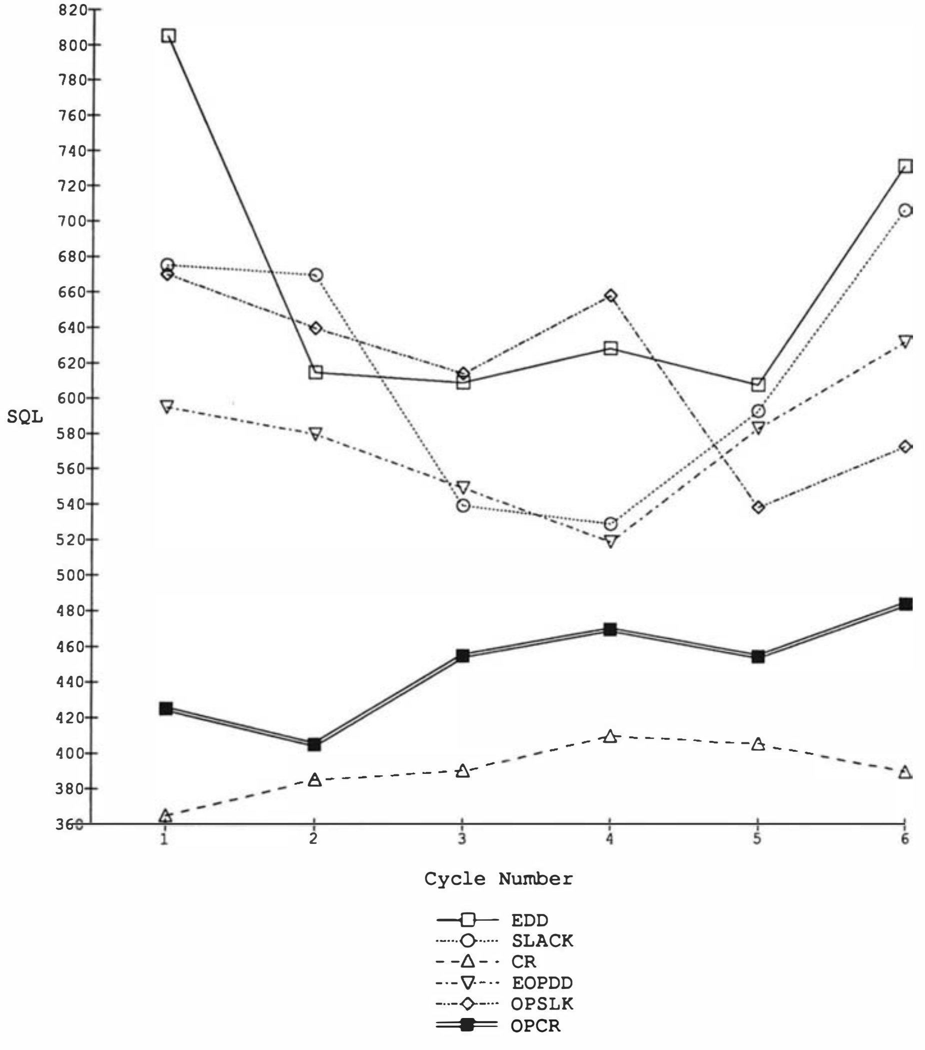


Table 4.2

VAR Comparisons Between Cycles Under EDD
Local Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR
1	1740
2	1853
3	1688
4	1759
5	1787
6	1775

Level of Significance of Pairwise Cycle Differences
(Two-tailed Wilcoxan Signed Rank Test)

		Cycle				
		1	2	3	4	5
	2	.9702				
	3	.4330	.6542			
Cycle	4	.5755	.5503	.4781		
	5	.6542	.6813	.5257	.9702	
	6	.7652	.9405	.8813	.4115	.6542

Global Allowance Policies. Figures 4.6 through 4.9 present the penalty median/cycle charts for the four penalty measures, respectively, using global allowance policy forms. Unlike the analogous charts using local forms, these charts display a strong general tendency for the early stages of the iterative process to produce successively lower median penalty measures, as hypothesized. In fact, the only case in which the median penalty measures do not generally decrease throughout the iterative process is the penalty measure SQL with the dispatching rule SLACK.

These charts provide several strong visual indications. The majority of benefits apparently occur in the first few cycles, with the median penalty measures apparently asymptotically (though not necessarily monotonically) approaching a lower limit. Virtually all benefits are achieved, generally, by the fifth or sixth cycle.

The dispatching rules EDD and SLACK seem to perform similarly, producing higher measures of inaccuracy penalties than the other four dispatching rules. For the measures VAR, MSL, and MAL, the dispatching rule CR produces the lowest inaccuracy penalty measures in later cycles, with the dispatching rules EOPDD, OPSLK, and OPCR performing similarly to each other. The relative performances of the six dispatching rules are evaluated quantitatively in chapter 5.

The penalty measure SQL appears to react differently to the iterative process than do the other three measures. The beneficial effects of iteration seem less compelling, and the successive median measures seem to exhibit more unsystematic variation.

These characteristics, unique to SQL in this research, are explainable and were anticipated. Early and late completions are

Figure 4.6

Median VAR Measures
Global Allowance Policy, 90% Utilization

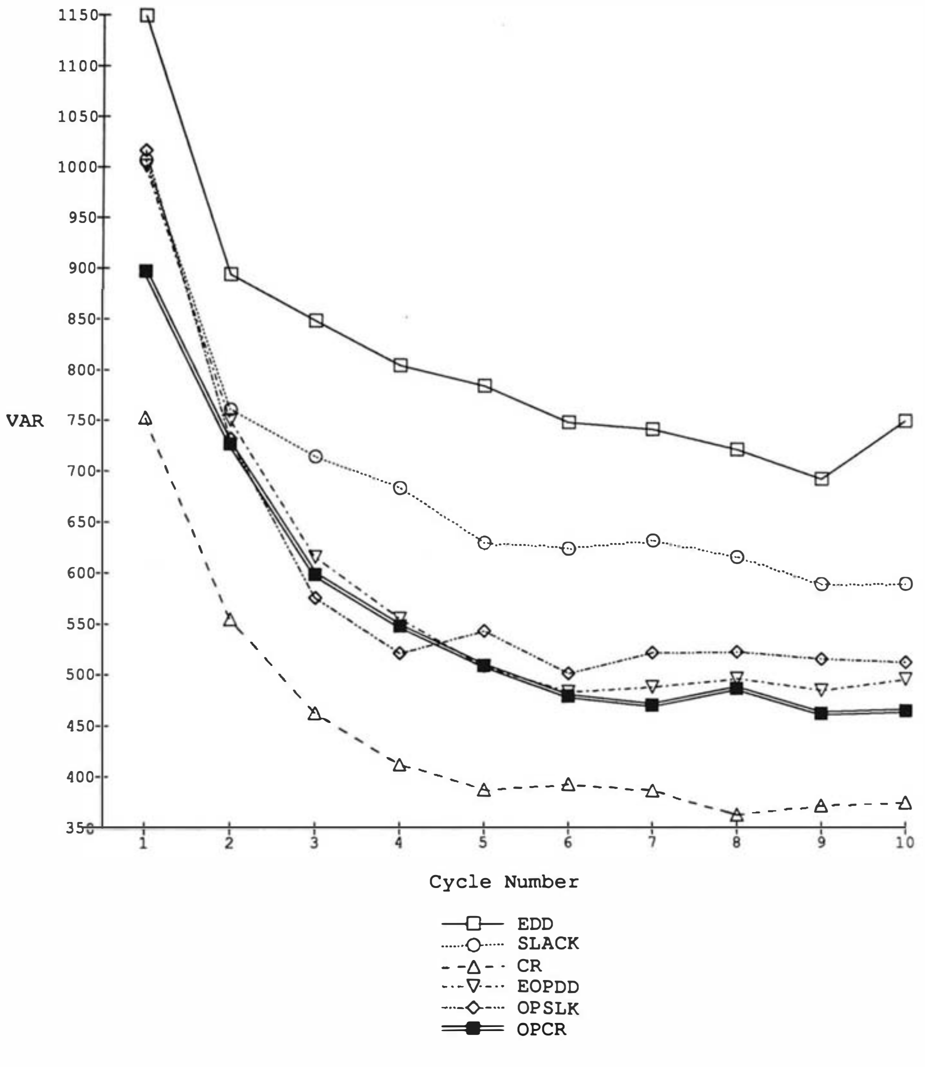


Figure 4.7

Median MSL Measures
Global Allowance Policy, 90% Utilization

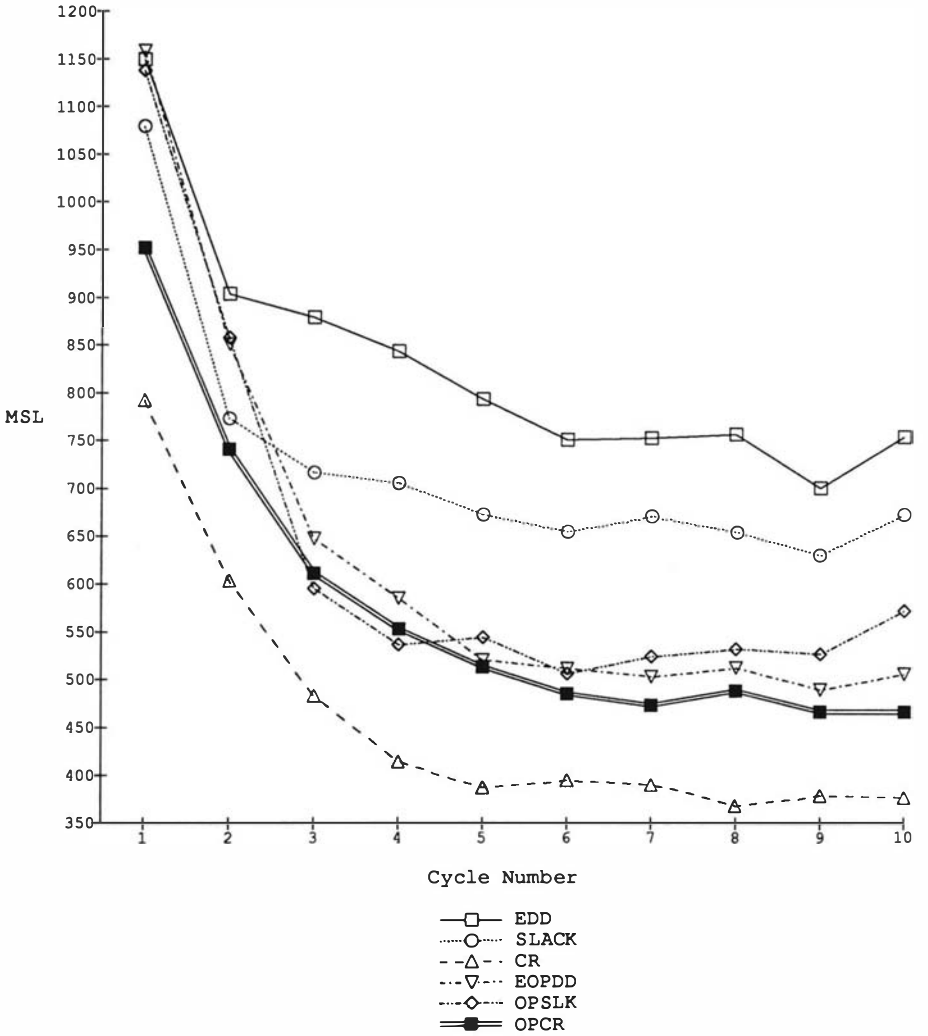


Figure 4.8

Median MAL Measures
Global Allowance Policy, 90% Utilization

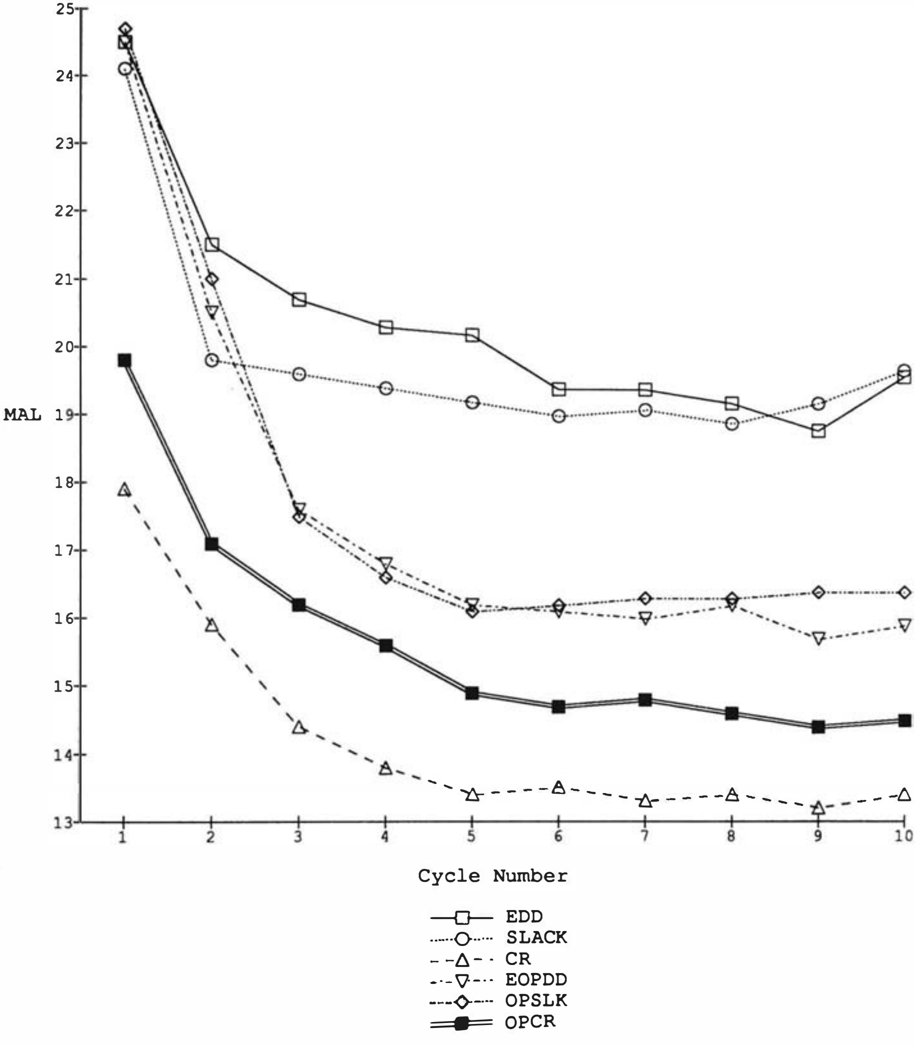
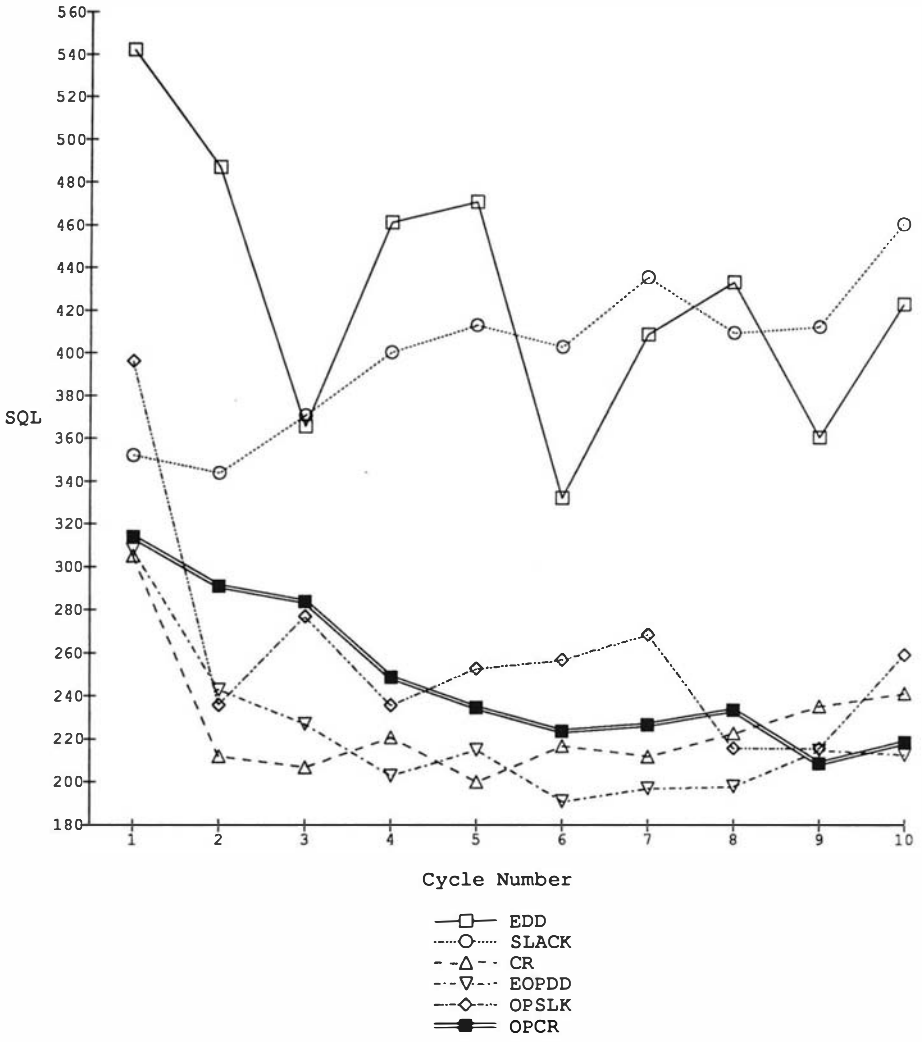


Figure 4.9

Median SQL Measures
Global Allowance Policy, 90% Utilization



penalized symmetrically under the penalty functions VAR, MSL, and MAL (about the mean lateness in the first function and about zero lateness in the last two functions). Under SQL, however, early completions are penalized according to a linear function while positive latenesses are penalized according to a quadratic function.

This non-symmetrical penalty assignment results in SQL being highly sensitive to the observed mean lateness in any given simulation. In the lateness ranges existing in this research, simulations in which the observed mean latenesses ended up as less than zero produced systematically lower SQL measures than simulations in which positive mean latenesses occurred. SQL measures, therefore, were often more erratic (for example, the dispatching rule EDD in Figure 4.9) than corresponding VAR, MSL, and MAL measures. In one case (the dispatching rule SLACK in Figure 4.9), the phenomenon of iteration producing successively higher mean latenesses overpowered the inherently beneficial effects of iteration and produced the previously discussed upward pressure on median SQL measures.

The visual indications provided by Figures 4.6 through 4.9 that successive iterative cycles generally produce successively lower completion inaccuracy penalties are strongly supported by pairwise statistical comparisons. Whereas under local allowance policy forms there were only occasional systematic statistically significant differences between successive cycles, under global allowance policy forms the penalty measure decreases produced by iteration are systematic and statistically significant. Table 4.3 displays, for the dispatching rule EDD, the median penalty measures for each cycle. Further, for each cycle in the iterative series, the table notes which other cycles in the

Table 4.3

Penalty Comparisons Between Cycles Under EDD
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	1149	2	3	4	5	6	7	8	9	10
2	894		3	4	5	6	7	8	9	10
3	849			4	5	6	7	8	9	10
4	806					6	7	8	9	10
5	786					6	7	8	9	10
6	750									
7	744									
8	724									
9	695									
10	753									

Cycle	Median MSL									
1	1149	2	3	4	5	6	7	8	9	10
2	904		3	4	5	6	7	8	9	10
3	880			4	5	6	7	8	9	10
4	845					6	7		9	
5	795					6	7		9	
6	753									
7	755									
8	759									
9	703									
10	757									

Cycle	Median MAL									
1	24.5	2	3	4	5	6	7	8	9	10
2	21.5		3	4	5	6	7	8	9	10
3	20.7			4	5	6	7	8	9	10
4	20.3					6	7	8	9	10
5	20.2					6	7	8	9	
6	19.4									
7	19.4									
8	19.2									
9	18.8									
10	19.6								9	

Cycle	Median SQL									
1	542		3			6	7		9	
2	487		3	4		6	7		9	
3	366									
4	462		3			6	7		9	
5	472		3			6	7		9	
6	333									
7	410					6			9	
8	435					6			9	
9	362									
10	425					6	7		9	

series were significantly less (in terms of the appropriate penalty measures) than that cycle. All comparisons are based on a one tailed Wilcoxon Signed Rank test, using .05 as the probability of a type I error.

The superscripts by the median penalty measure in any cycle indicate which other cycles were significantly less than that particular cycle. For example, for the measure VAR one sees that the median of the twenty variances produced by the first cycle was 1149, and that the variances produced in this cycle were significantly larger than those of cycles two through ten, inclusive. The median variance produced by the fifth cycle was 786, which was significantly larger than cycles six through ten, inclusive. The variances produced by cycles six through ten, however, were not significantly different from each other.

The patterns of significant pairwise differences for the other five dispatching rules are similar to those of EDD shown in Table 4.3, and strongly support the hypothesis of iteration generally producing successively lower penalty measures under global allowance policy forms. Under this hypothesis, one would expect measures produced in any cycle to tend to be significantly greater than or equal to those in subsequent cycles, and significantly less than those in few (if any) subsequent cycles. While the analogous tables for the other five dispatching rules are not shown here, a complete set of tables is provided in Appendix E.

As previously discussed, Figures 4.6 through 4.9 give visual indications that the majority of benefits from iteration occur in the first five or six cycles. Further, the data shown in Table 4.3 and in Appendix E provide indications of the point in the iterative process at which the incremental benefits of an additional cycle become insignificant.

Again, refer to the penalty measure VAR in Table 4.3. One sees that the first cycle was significantly greater (in terms of the penalty measure) than 100% (nine out of nine) of all other cycles. The fourth cycle was significantly greater than 56% (five out of nine) of all other cycles. The sixth cycle was significantly greater than no other cycles.

Under the hypothesized iteration effects, one would expect this "greater than" percentage to decrease throughout the iteration process until marginal benefits become insignificant. For the measure VAR in Table 4.3, this threshold of insignificance appears to occur at the sixth cycle. Figure 4.10 displays the within-series "greater than" percentages, aggregated across all dispatching rules, for VAR, MSL, MAL, SQL, and the four measures combined. Although one again sees the somewhat more erratic nature of SQL, the figure displays a strong indication that marginal benefits from iteration occur through the sixth cycle, but not thereafter.

The iteration process produces other systematic beneficial effects in terms of lateness penalties. Not only do successive cycles produce successively lower penalty measures, but the dispersion of the twenty observations produced at each successive cycle is also decreased. Figure 4.11 displays an example of this benefit, in the form of the standard deviations of the twenty VAR observations for each cycle and dispatching rule. Similar patterns exist for the other three penalty measures.

Figure 4.10

Percentage of "Significantly Greater Than" Pairwise Comparisons Within Iteration Series, Collapsed Across Dispatching Rules

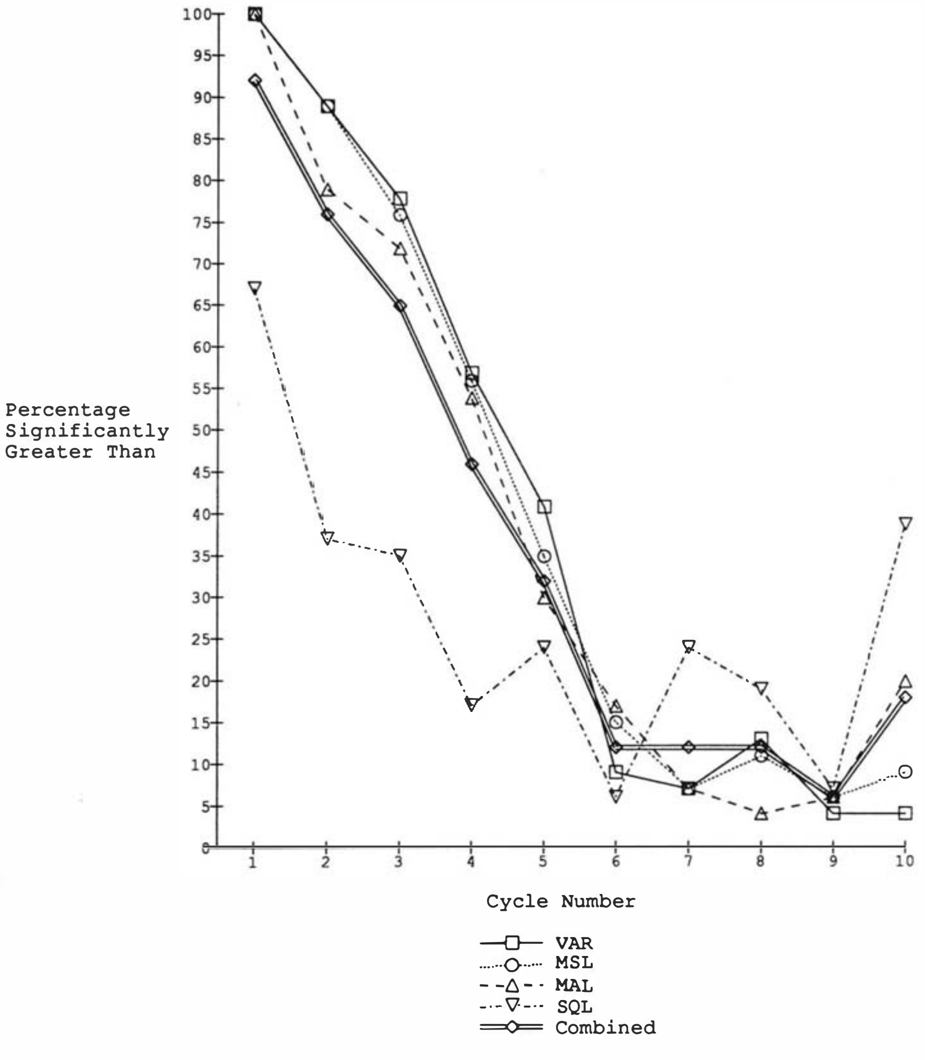
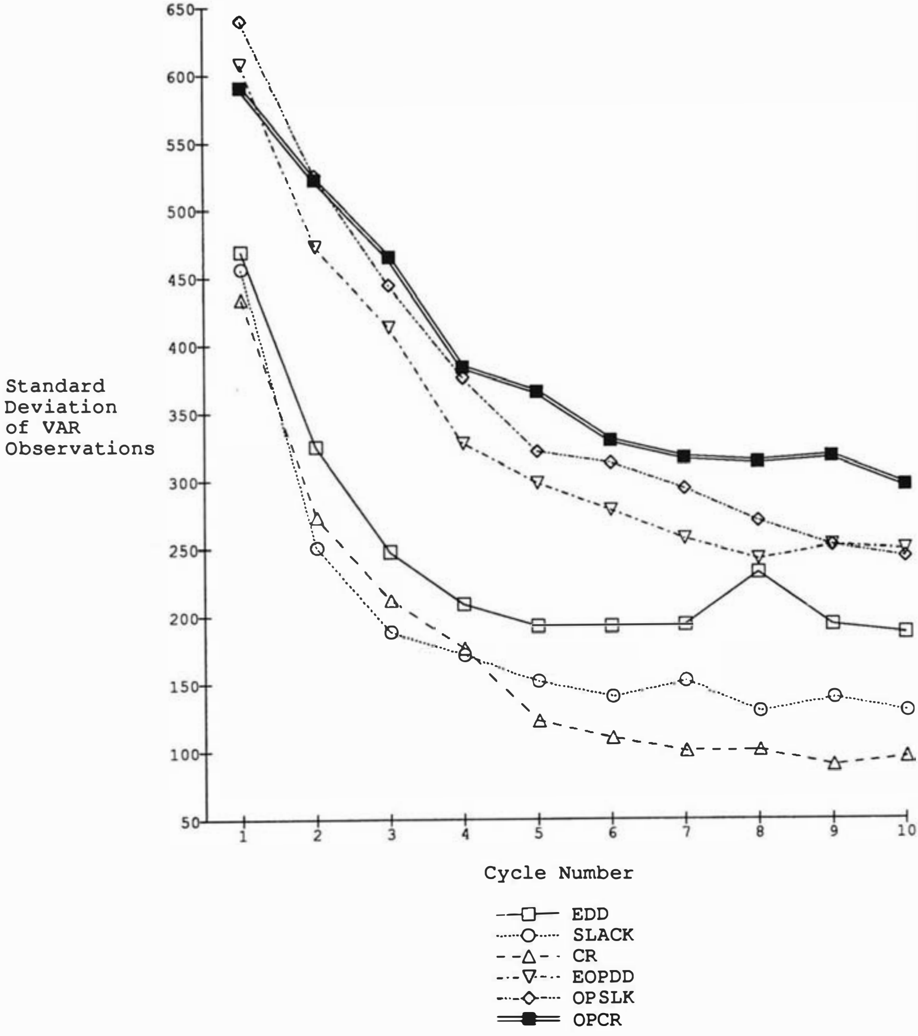


Figure 4.11

Standard Deviations of VAR Observations
Global Allowance Policy, 90% Utilization



Discussion of Global vs. Local Effects. The fact that using an iterative process in allowance equation determination provides benefits under global allowance policy forms but not under local allowance policy forms is not unexpected. An examination of the ordinates in Figures 4.2 through 4.9 indicates that allowance estimators that are limited to local variables are considerably cruder than those that also incorporate global variables. The nature of the iterative process itself is one of fine tuning; a good initial solution is improved upon by process repetition. Local allowance policy estimators simply are not sensitive enough to exploit the potential benefits of iteration.

Consider the first two cycles of an iterative procedure. Since the same incoming job stream is presented to each cycle, any given job i arrives at the same time in the second cycle as in the first and has identical operation characteristics. Therefore, the only factors relevant to allowance determination that have changed from the first cycle to the second cycle as of the arrival of job i are shop related factors. Global allowance policies explicitly account for these shop factors, whereas local allowance policies do not. It is logical, therefore, to expect global allowance estimators to be more sensitive to the effects of iteration than local allowance estimators.

Tests of Hypothesis 1

Table 4.4 displays the formal tests that address whether or not, under global allowance policy forms, allowance policies determined from an iterative process produce significantly lower inaccuracy penalties than policies determined from a single pilot simulation (cycle 1 policies). For each of the four penalty measures within each

Table 4.4

Phase 1 Statistical Comparisons
Allowance Policy Determination by
Single Pilot (Cycle 1) vs. Iterative Process (Cycle 6)
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

		Cycle 1	Cycle 6	P-Values[1]
EDD	var	1149	750	<.00005
	msl	1149	753	<.00005
	mal	24.5	19.4	<.00005
	sql	542	333	.0045
SLACK	var	1006	625	<.00005
	msl	1079	656	<.00005
	mal	24.1	19.0	<.00005
	sql	352	404	.6450
CR	var	752	392	<.00005
	msl	792	394	<.00005
	mal	17.9	13.5	<.00005
	sql	305	217	.0183
EOPDD	var	1001	484	<.00005
	msl	1158	512	<.00005
	mal	24.5	16.1	<.00005
	sql	308	191	.0011
OPSLK	var	1016	502	<.00005
	msl	1138	507	<.00005
	mal	24.7	16.2	<.00005
	sql	396	257	.0018
OPCR	var	897	480	<.00005
	msl	952	486	<.00005
	mal	19.8	14.7	<.00005
	sql	314	224	.0020

[1] One-tailed H_0: Cycle 1 ≤ Cycle 6
H_1: Cycle 1 > Cycle 6

dispatching rule, the twenty observations from the first cycle are tested against the twenty observations from the sixth cycle using a one-tailed Wilcoxon Signed Rank test. The column labelled "P-value" shows the level of significance at which the sixth cycle is less than the first cycle in terms of the penalty measures.

With the exception of the previously noted case of SQL under the dispatching rule SLACK (where there is no significant difference between the first cycle and the sixth cycle), every test shows that penalty values from the sixth cycle are significantly less than those from the first cycle. The median percentage decreases in the penalty measures VAR, MSL, MAL, and SQL are 47%, 50%, 26%, and 32%, respectively. Significant benefits are produced by the use of an iterative procedure in setting allowance equations.

Phase 2

Recall that in cases where operation-based dispatching rules have been evaluated, it has been common practice in past research to set operation allowances by allocating a total job allowance among operations, usually in proportion to their respective operation processing times. Hypothesis 2 proposes that, with an effective allowance estimator, increased accuracy results from estimating cumulative operation allowances directly from the allowance equation. Phase 2 of this research addresses Hypothesis 2.

Tests of Hypothesis 2

As with the effects of iteration tested in Phase 1, benefits afforded by direct estimation of operation allowances appear to differ between local allowance policy forms and global allowance policy forms. Tables 4.5 and 4.6 display the results of the Wilcoxon tests of Hypothesis 2 for local and global allowance policy forms, respectively. Both tables represent environments of 90% expected utilization and actual processing times that are known as of a job's arrival at the shop. Based on Phase 1 analyses, simulations under local allowance policy forms are first cycle simulations, whereas those under global forms are sixth cycle simulations.

The tests in Table 4.5 indicate that direct estimation of operation allowances neither increased nor decreased inaccuracy penalties on a systematic basis under local allowance policy forms. For the measures VAR, MSL, MAL, and SQL under the dispatching rule EOPDD, direct estimation significantly decreased inaccuracy penalties. For the measure SQL under OPSLK and the measures MAL and SQL under OPCR, direct estimation significantly increased inaccuracy penalties. For all other combinations of penalty measures and dispatching rules there were no statistically significant differences. As with the effects of iteration, local allowance policy forms do not appear sensitive enough to exploit potential advantages of direct estimation.

The tests in Table 4.6, however, indicate that direct estimation provides consistently significant benefits in terms of inaccuracy penalties under global allowance policy forms. In each of the twelve combinations of dispatching rules and penalty measures, inaccuracy penalties associated with direct estimation of operation allowances are significantly lower than those associated with proportional allocation of job allowances. The median percentage decreases in the penalty

Table 4.5

Phase 2 Statistical Comparisons
Operation Allowance Determination by
Allocation vs. Direct Estimation
Local Allowance Policy Forms, Cycle 1 Simulations
(90% Utilization, Known Actual Processing Times)

		Job Allowance Allocation	Direct Estimation	P-Values[1]
EOPDD	var	1713	1611	.0430
	msl	1957	1809	.0002
	mal	33.4	32.4	.0004
	sql	599	569	.0152
OPSLK	var	1765	1727	.4407
	msl	1955	1962	.7492
	mal	34.0	33.4	.1754
	sql	667	801	.9994
OPCR	var	1301	1282	.2629
	msl	1392	1417	.8839
	mal	24.7	25.1	.9963
	sql	426	449	.9874

[1] One-tailed H_0: Allocation ≤ Direct
H_1: Allocation > Direct

Table 4.6

Phase 2 Statistical Comparisons
Operation Allowance Determination by
Allocation vs. Direct Estimation
Global Allowance Policy Forms, Cycle 6 Simulations
(90% Utilization, Known Actual Processing Times)

		Job Allowance Allocation	Direct Estimation	P-Values[1]
EOPDD	var	484	342	<.00005
	msl	512	374	<.00005
	mal	16.1	13.7	<.00005
	sql	191	125	<.00005
OPSLK	var	502	380	<.00005
	msl	507	391	.0001
	mal	16.2	14.6	.0002
	sql	257	152	<.00005
OPCR	var	480	333	<.00005
	msl	486	358	.0001
	mal	14.7	13.1	.0010
	sql	224	136	.0008

[1] One-tailed H_0: Allocation $\leq$ Direct
H_1: Allocation $>$ Direct

measures VAR, MSL, MAL, and SQL are 29%, 26%, 11%, and 39%, respectively.

Direct estimation therefore produces significant positive effects under global forms and no consistently positive or negative effects under local forms. In order to maintain procedural consistency, and since there are no compelling reasons not to, all simulations performed in Phase 3 analyses use the direct estimation procedure.

Summary

Under global allowance policy forms, determination of allowance equation coefficients by an iterative simulation-regression procedure significantly reduces completion inaccuracy penalties (Hypothesis 1), as well as reducing the dispersion of those penalties. The allowance equations generated by the iterative procedure tend to become successively more stable as cycles are repeated.

Under global allowance policy forms and operation-based due dates, direct estimation of cumulative operation allowances in setting due dates produces significantly lower completion inaccuracy penalties than allocation of total job allowances among operations in proportion to their respective operation processing times (Hypothesis 2).

Chapter 5

Phase 3 Results

The first section of this chapter discusses the results from Phase 3. The next section presents comparisons to the Ragatz and Mabert study (1984), and the final section provides a chapter summary.

Phase 3

Phase 3 analyses address Hypotheses 3 through 6. The experimental design of Phase 3 was illustrated previously in Figure 3.3. This factorial design is evaluated for each of the four penalty measures.

For each penalty measure evaluated, each cell in the experimental design matrix contains twenty observations, representing (as in Phases 1 and 2) simulations run under the same assumptions but on twenty different job streams. The observations within any given cell therefore comprise a random sample of observed measures within the given environment. Since the design consists of 48 cells, each matrix contains 960 observations (20 observations by 48 cells).

As indicated by analyses in Phases 1 and 2, the data in each cell that corresponds to a local allowance policy are generated from first cycle (no iteration) simulations, whereas the data in each cell that corresponds to a global allowance policy are generated from cycle 6 simulations. Further, all data in cells that correspond to operation-based dispatching rules utilize direct estimation of operation allowances.

Observed Measures

Table 5.1 displays the medians of the twenty appropriate observations for each of the four penalty measures in each of the 48 combinations of assumptions. The observations from which each median was calculated tend to be positively skewed. As examples, frequency tables of individual VAR observations under assumptions of global allowance forms and 90% utilization are displayed in Table 5.2.

Comparisons of the medians in Table 5.1 give immediate support to stated hypotheses. Within each of the four penalty measures there are 24 possible comparisons of local vs. global allowance forms (local vs. global under EDD, 75% utilization, and known actual processing times; local vs. global under EDD, 75% utilization, and unknown actual processing times, etc.). Within each of the four measures the median penalty under a global policy is less than the corresponding median penalty under a local policy in all 24 cases. Under the assumptions of 90% utilization and known actual processing times, the median percentage decreases in the penalty measures VAR, MSL, MAL, and SQL are 72%, 74%, 46%, and 64%, respectively. Benefits of utilizing global variables in allowance estimation are highly significant.

Likewise, within each penalty measure there are 24 comparisons of operation-based dispatching rules vs. job-based dispatching rules. An example is EOPDD vs. EDD under local allowance policies, 75% utilization, and known actual processing times. For the penalty measure VAR, the median for the operation-based dispatching rule is lower than

Table 5.1

Phase 3 Observed Median Penalty Measures
Dispatching/Allowance/Shop Condition Evaluations

		Local Allowance				Global Allowance			
		75% Utilization		90% Utilization		75% Utilization		90% Utilization	
		Act TPT Known	Act TPT Not Known	Act TPT Known	Act TPT Not Known	Act TPT Known	Act TPT Not Known	Act TPT Known	Act TPT Not Known
EDD	var	694	691	1740	1799	292	302	750	735
	msl	712	707	1800	2014	295	302	753	757
	mal	18.8	19.2	31.9	33.6	12.3	12.7	19.4	19.8
	sql	349	358	803	683	183	176	333	461
SLACK	var	652	673	1750	1832	237	251	625	659
	msl	703	705	1902	1951	239	251	656	673
	mal	18.8	19.1	33.3	33.8	11.4	11.7	19.0	18.8
	sql	379	317	676	633	153	128	404	354
CR	var	566	592	1322	1267	170	188	392	417
	msl	624	622	1453	1437	170	189	394	418
	mal	17.0	17.2	24.3	25.2	9.2	9.6	13.5	14.0
	sql	236	218	364	381	100	101	217	217
EOPDD	var	509	525	1611	1694	156	171	342	364
	msl	515	546	1809	1950	157	171	374	392
	mal	16.6	17.0	32.4	32.6	9.0	9.6	13.7	14.6
	sql	289	273	569	781	75	88	125	111
OPSLK	var	593	625	1727	1771	198	229	380	384
	msl	608	663	1962	2157	199	230	391	397
	mal	18.2	19.1	33.4	35.0	10.3	11.2	14.6	14.7
	sql	348	326	801	752	107	112	152	173
OPCR	var	467	495	1282	1365	157	179	333	335
	msl	513	550	1417	1498	159	180	358	354
	mal	16.0	16.8	25.1	26.1	9.0	9.9	13.1	13.4
	sql	139	157	449	407	87	86	136	130

Table 5.2

Frequency Distributions of Variances in Cells
Global Allowance Policy, Cycle 6
(90% Utilization, Known Actual Processing Times)

	Frequencies					
Variance	EDD	SLACK	CR	EOPDD	OPSLK	OPCR
200 - 299	0	0	2	0	0	1
300 - 399	0	0	10	2	1	2
400 - 499	0	2	5	9	4	8
500 - 599	2	6	2	2	8	4
600 - 699	4	6	0	4	2	2
700 - 799	8	4	1	1	2	0
800 - 899	2	0	0	1	1	2
900 -1000	1	2	0	0	1	0
1000 -1099	1	0	0	0	0	0
≥ 1100	2	0	0	1	1	1

the median for the corresponding job-based dispatching rule in 23 of the 24 cases. For MSL, the operation-based rule is lower in 20 out of 24 cases. For MAL as well as for SQL, the operation-based rule is lower in 18 out of 24 cases. Under the assumptions of 90% utilization, global allowance policy forms, and known actual processing times, the median percentage decreases in the penalty measures VAR, MSL, MAL, and SQL are 39%, 40%, 23%, and 38%, respectively. Operation-based dispatching rules generally provide significant benefits over corresponding job-based dispatching rules.

Within each measure there are 24 direct comparisons of actual processing times known as of a job's arrival at the shop vs. unknown as of a job's arrival at the shop. An example is known vs. unknown under EDD, local allowance policies, and 75% utilization. For the measures VAR, MSL, and MAL, values under known times are less than those under unknown times in 21 out of 24 cases, 20 out of 24 cases, and 23 out of 24 cases, respectively. For the measure SQL, however, the median penalty under known times is less than the median penalty under unknown times in only 9 out of 24 cases. Apparently, the previously discussed characteristic that SQL values are highly variable due to increased sensitivity to observed mean latenesses overwhelms the additional systematic variation contributed by actual processing times varying about expected processing times. If the chosen variance of the deviational distribution that defined the stochastic differences of actual about expected times had been sufficiently large, logic dictates that observed SQL values under known times also would have been significantly less than those under unknown times.

Conclusions that are not specifically related to stated hypotheses can be made from comparisons of the median measures in Table 5.1. Within each measure, values produced under 75% utilization environments are significantly lower than those produced under 90% utilization environments (24 out of 24 cases within each of the four measures).

Within each of the eight combinations of allowance policy class, utilization level, and actual processing time assumption, one can compare the relative performances of the six dispatching rules for each of the four penalty measures. The dispatching rule OPCR produces either the lowest or second lowest VAR, MSL, and SQL measures in 8 out of 8 cases. OPCR produces the lowest or second lowest MAL measures in 7 out of 8 cases. No other dispatching rule exhibits such overall superiority.

The dispatching rules EDD and SLACK exhibit strong tendencies to be the worst performers among the six dispatching rules examined. EDD is one of the bottom two performers for VAR in 8 out of 8 cases, for MSL in 7 out of 8 cases, for MAL in 6 out of 8 cases, and for SQL in 7 out of 8 cases. SLACK is one of the bottom two performers for VAR in 8 out of 8 cases, for MSL in 7 out of 8 cases, for MAL in 8 out of 8 cases, and for SQL in 5 out of 8 cases.

Regression Analyses

The indications provided above are based on pairwise comparisons of the 48 observed medians within each of four factorial design matrices. These indications can be supported and extended by regression analyses that specifically address all 960 observations within each matrix. As discussed in Chapter 3, four stepwise multiple regression analyses were performed; in each, ten times the natural logarithm of the appropriate penalty measure was the dependent variable, and potential independent variables presented to the stepwise procedure consisted of one scalar variable (lateness) and numerous dummy variables representing assumption combinations, as well as selected second and third order interactions. This section presents results of the regression procedures in the forms of observed coefficients and residual analyses. The following section presents interpretations of these results in terms of the stated hypotheses.

Observed Coefficients. Selected coefficients produced by the four regression analyses, as well as the corresponding standard errors of the coefficients, are displayed in Table 5.3. Only coefficients that are significantly different from zero and relevant to the stated hypotheses are included in this table. While numerous other variables in each regression exhibit significant coefficients (for example, the dummy variable denoting job stream 10), these variables are included in the regressions only to account directly for certain systematic variations and are not specifically relevant to stated hypotheses.

In each regression the base from which each dummy variable deviates represents an environment of local allowance form, 75% expected utilization, actual processing times that are known as of a job's arrival at the shop, and the dispatching rule EDD run on the job stream denoted as job stream 1. In cases where coefficients for a given variable are not significant, the table is blank (for example, the

Table 5.3

Phase 3 Regressions
Coefficients and Standard Errors of Coefficients

	Dependent Variable: 10 x Natural Logarithm of							
	VAR		MSL		MAL		SQL	
Variable	Coeff	Std Error	Coeff	Std Error	Coeff	Std Error	Coeff	Std Error
Global	- 7.599	.225	- 8.247	.265	- 4.200	.135	- 6.964	.315
High	11.208	.173	11.999	.204	5.812	.116	8.583	.255
Unknown	.433	.099	.442	.117	.278	.059		
CR	- 1.276	.241	- 1.260	.284	- 1.345	.146	- 4.735	.269
EOPDD	- 1.421	.202	- 1.326	.238	- .709	.145	- 1.457	.345
OPCR	- 2.256	.241	- 2.441	.284	- 1.689	.146	- 4.952	.347
Global*High	- 2.449	.201	- 2.774	.237	- 1.500	.121	- 2.675	.340
Global*SLACK	- 2.047	.242	- 2.009	.286	- 1.023	.162	- 1.149	.489
Global*CR	- 3.617	.315	- 3.572	.372	- 1.342	.189		
Global*EOPDD	- 5.007	.316	- 4.792	.373	- 2.524	.189	- 5.998	.490
Global*OPSLK	- 4.371	.269	- 4.343	.318	- 2.226	.145	- 5.267	.379
Global*OPCR	- 3.951	.315	- 3.426	.372	- 1.170	.189	- 2.486	.488
High*CR	- 1.333	.277	- 1.620	.327	- .961	.175		
High*OPSLK	- 1.239	.233	- 1.234	.275				
High*OPCR	- .974	.278	- 1.199	.328	- .775	.175		
High*SLACK					.580	.146	1.682	.448
High*EOPDD					.525	.175		
SLACK							- 1.164	.412

variable "unknown" where ten times the natural logarithm of the observed penalty measure under SQL is the dependent variable).

One interesting result not displayed in Table 5.3 concerns the variable "lateness". The standardized coefficients for this variable produced by the VAR, MSL, MAL, and SQL regressions are .110, .067, .056, and .512, respectively. The higher standardized coefficient in the SQL regression supports previous statements that the measure SQL is more sensitive to observed mean latenesses than are the other three measures.

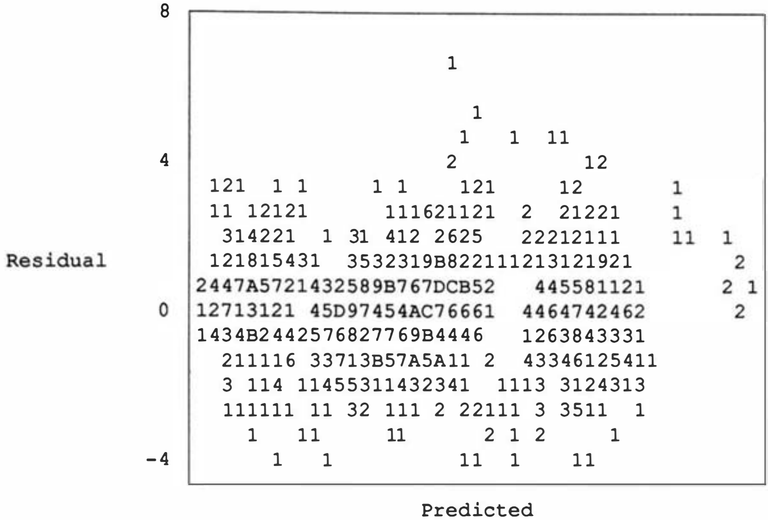
Residual Analyses. Statistical conclusions made in the next section rely on the fact that assumptions underlying the regression analysis procedures are not violated to the extent that such conclusions are invalid. Figures 5.1 through 5.4 show residual plots and expected normal value plots for the VAR, MSL, MAL, and SQL regressions, respectively.

The four residual plots give no visual indications of the presence of heteroscedasticity. The plots of the expected normal values display some evidence of non-normality in the positive tails of the VAR, MSL, and MAL plots and in both tails of the SQL plot. All four sets of residuals display a bell-shaped distribution with a slight tendency of leptokurtosis, and the sets for VAR, MSL, and MAL exhibit a slight positive skew. χ^2 goodness-of-fit tests show that, while residuals are not significantly non-normal for the VAR and MSL residual sets at α = .01, the MAL and SQL residual sets are significantly non-normal at α =.01.

The potential presence of autocorrelation of residuals (when residuals are ranked by magnitude of predicted penalty measure) is

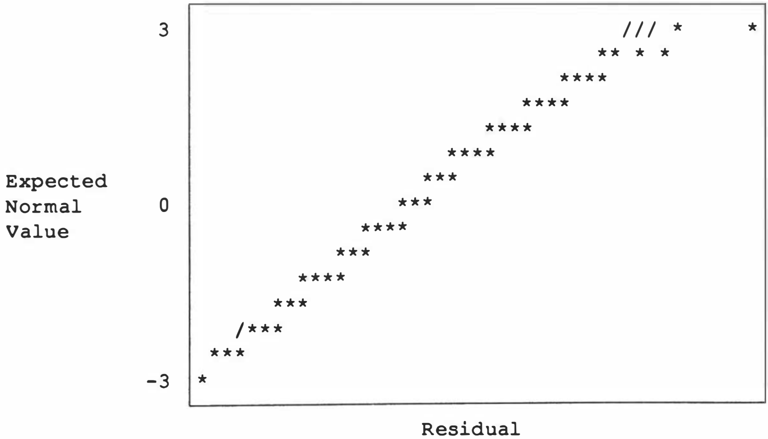
Figure 5.1

Phase 3 Regression Residual Plots - VAR



(Note: 1=1 obs,...,9=9 obs,A=10 obs,B=11 obs, etc.)

Normal Probability Plot of Residuals



(Note: Expected Normal Probability Value Should Lie on ///)

Figure 5.2

Phase 3 Regression Residual Plots - MSL

```
            10 |                                            |
               |                      1                     |
               |                      1                     |
               |                                            |
               |                            1               |
               |                       1                1   |
             5 |                      11      21       12   |
               |                    1 2     1111311    1    |
               |  1 11121      1122 1 131   1  3112     1   |
Residual       |  4 41221  12 3211423111   1 13121      11  |
               |  224655 11 43214345956     3 225 21     22 |
               |1266883436147B7967EEJ931   12 72353 1       |
             0 | 25868525358985957BC9992   1124C93264       |
               |  211116 33713B57A5A11 2  43346125411       |
               |  3 114 11455311432341  1113 3124313        |
               |  111111 11 32 111 2 22111 3 3511  1        |
               |    1   11     11     2 1 2     1           |
               |      1   1          11  1    11            |
            -5 |                                            |
                                  Predicted
         (Note: 1=1 obs,...,9=9 obs,A=10 obs,B=11 obs, etc.)
```

Normal Probability Plot of Residuals

```
             3 |                             ///       *  * |
               |                           //***   *        |
               |                         *****              |
               |                      ****                  |
               |                   ****                     |
               |                 ***                        |
Expected       |               ***                          |
Normal       0 |             ***                            |
Value          |           ***                              |
               |         ***                                |
               |      ****                                  |
               |    ***                                     |
               |  ***                                       |
               | /**                                        |
            -3 | **                                         |
                                  Residual
     (Note: Expected Normal Probability Value Should Lie on ///)
```

Figure 5.3

Phase 3 Regression Residual Plots - MAL

```
          6  +------------------------------------------+
             |                                          |
             |                     1                    |
             |                                          |
             |                      1                   |
             |                         1                |
          3  |                       1  1             12|
             |                       1   1 122        1 |
             |                 1       11   341         |
             |    114131 1    225214311 3  241   21   112
Residual     |    34637 1275264339533  3   2 33   1     |
             |   249C6455986AA8BAC9B1  2232242641       |
          0  |   16D9E91A5B9ACA8EMFG2 13253477B45       |
             |   2562433356D55798J9B1 241435E94121      |
             |    1453 11545637496A33  7464275621       |
             |     12 112 1323131112   336121 321       |
             |            1 1 2  21   23 2    1  1      |
             |             3                            |
         -3  |                                          |
             +------------------------------------------+
                            Predicted
       (Note: 1=1 obs,...,9=9 obs,A=10 obs,B=11 obs, etc.)
```

Normal Probability Plot of Residuals

```
          3  +--------------------------------//--------*---*
             |                             ///**** *         |
             |                           ******              |
             |                        ****                   |
             |                     ****                      |
             |                   ***                         |
Expected     |                 ***                           |
Normal    0  |               ***                             |
Value        |             ***                               |
             |          ****                                 |
             |        ***                                    |
             |      ***                                      |
             |    ***                                        |
             |*//**                                          |
         -3  |*                                              |
             +-----------------------------------------------+
                               Residual
   (Note: Expected Normal Probability Value Should Lie on ///)
```

Figure 5.4

Phase 3 Regression Residual Plots - SQL

```
         9   |            21      1
             |           1      1 1
             | 1        1      1 1
             |  2 1     22    1 11
             |311121 1 2112   3222
             |1 1234125334255647355 1       1
             |  322926987CE84A49223331      21
             | 311572DG789D88AB382633
         0   | 1 476CEJD9CA6E3C6642462 1
Residual     |  1A58697BIDDCB884152141 1 41
             |   66C769C669674213122  1 1 2
             |  1 9155875335431  2 1  1 1 11     1
             |13  2 452312 1     1          11     2
             |      3211 2     1 3                   11
             |  1        1
             |         1 1
        -9   |         1                                11
             |                                           1
```

Predicted
(Note: 1=1 obs,...,9=9 obs,A=10 obs,B=11 obs, etc.)

Normal Probability Plot of Residuals

```
          3   |                                        //  *
              |                                     //*******
              |                                  *****
              |                               ****
              |                             ***
              |                           ***
Expected      |                        ****
Normal    0   |                      ***
Value         |                   ****
              |                 ***
              |              ****
              |            ***
              |         ****
              |  ***  /**
         -3   | **  //
```

Residual
(Note: Expected Normal Probability Value Should Lie on ///)

addressed for each of the four residual sets by the Durbin-Watson test as well as by a test of the number of positive and negative runs. The observed Durbin-Watson values for the VAR, MSL, MAL, and SQL residual sets are 1.813, 1.920, 1.882, and 1.751, respectively. None are sufficiently low to reject the hypothesis of no autocorrelation at α = .05. The observed z values for the runs tests are -1.29, -.65, +.71, and -3.17, respectively. Only in the SQL residual set is there evidence of autocorrelation.

In summary, while there are no indications of problems with heteroscedasticity, there are statistically significant indications of moderate departures from normality in the MAL and SQL residual sets, and some indications of autocorrelation in the SQL residual set.

Tests of Remaining Hypotheses

Since some assumptions underlying regression theory are violated to some degree, statistical conclusions provided by the four regression analyses and relating to stated hypotheses are tested further by sets of Wilcoxon Signed Rank tests. In all cases the regression conclusions are supported by the Wilcoxon tests; apparently, the observed departures from underlying assumptions are not sufficient to affect the regression results to a meaningful extent.

As previously mentioned, the dependent variables in the four regression analyses are ten times the natural logarithms of the appropriate inaccuracy penalty measures. Since the natural logarithm is a monotonic function of the raw penalty value, throughout this section conclusions from the regression analyses are stated in terms of the untransformed penalty values. Supporting Wilcoxon tests (α = .05) are performed in terms of the untransformed values, and are therefore interpreted naturally.

Hypothesis 3. Hypothesis 3 proposes that global allowance forms produce lower inaccuracy penalties than do local allowance policy forms. Coefficients for the dummy variable "global" are negative and significant in all four regressions (see Table 5.3), indicating that the main effect of global allowance forms is a significant reduction in penalty measures. Further, coefficients for all significant interaction variables that address the global state are negative and significant, indicating that the main global superiority exists in every case.

Table 5.4 presents the p values associated with all pairwise comparisons, within each penalty measure, between the twenty observations in the global cell under given assumptions and the twenty matched observations in the corresponding local cell under the same assumptions. For example, the twenty observations under global/ EDD/ 75% utilization/ known actual processing times are compared with the twenty observations under local/ EDD/ 75% utilization/ known actual processing times.

Within each penalty measure, global values are significantly lower than local values at the stated α of .05 in all 24 comparisons. The observed Wilcoxon tests strongly support the regression conclusions in affirming Hypothesis 3.

Hypothesis 4. Hypothesis 4 proposes that operation-based dispatching rules produce lower inaccuracy penalty measures than do the corresponding job-based dispatching rules. Both the regression

Table 5.4

Hypothesis 3 Pairwise Tests
Global vs Local P-Values[1]

Global vs Local Under Stated Dispatching/ Utilization/ Processing Time Assumptions	VAR	MSL	MAL	SQL
EDD/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
EDD/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
EDD/ 90%/ Known	<.00005	<.00005	<.00005	.0002
EDD/ 90%/ Unknown	<.00005	<.00005	<.00005	.0032
SLACK/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
SLACK/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
SLACK/ 90%/ Known	<.00005	<.00005	<.00005	.0025
SLACK/ 90%/ Unknown	<.00005	<.00005	<.00005	.0050
CR/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
CR/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
CR/ 90%/ Known	<.00005	<.00005	<.00005	.0020
CR/ 90%/ Unknown	<.00005	<.00005	<.00005	.0007
EOPDD/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
EOPDD/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
EOPDD/ 90%/ Known	<.00005	<.00005	<.00005	<.00005
EOPDD/ 90%/ Unknown	<.00005	<.00005	<.00005	<.00005
OPSLK/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
OPSLK/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
OPSLK/ 90%/ Known	<.00005	<.00005	<.00005	<.00005
OPSLK/ 90%/ Unknown	<.00005	<.00005	<.00005	<.00005
OPCR/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
OPCR/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
OPCR/ 90%/ Known	<.00005	<.00005	<.00005	<.00005
OPCR/ 90%/ Unknown	<.00005	<.00005	<.00005	<.00005

[1] One tailed H_0: Global $\geq$ Local
H_1: Global $<$ Local

analyses and the Wilcoxon tests indicate that, in general, there are statistically significant benefits associated with operation-based dispatching rules. However, both methods of analysis indicate that there are specific combinations of shop assumptions where benefits are not significant.

Practical interpretations of the regression analyses are difficult due to the large number of significant interactions that are observed. When one combines appropriate coefficients and variances/covariances, one sees that within the measure VAR (collapsed across dispatching rules) the operation-based rules produce significantly lower penalties than do the analogous job-based rules in 18 out of 24 cases. Within MSL, MAL, and SQL, the penalties produced by the operation-based rules are significantly lower in 20 out of 24 cases, in 18 out of 24 cases, and in 18 out of 24 cases, respectively.

Wilcoxon tests show that within VAR, MSL, MAL, and SQL, penalties produced by operation-based dispatching rules are significantly (α = .05) lower than those produced by corresponding job-based rules in 18 out of 24 cases, in 16 out of 24 cases, in 12 out of 24 cases, and in 15 out of 24 cases, respectively (see Table 5.5). The hypothesized benefits of operation-based rules are, in general, strongly supported.

Both the regression and Wilcoxon analyses indicate, however, that benefits tend to be insignificant for the measure MAL under the dispatching rule comparison OPCR vs. CR, for all measures under local allowance forms and 90% expected utilizations, and for all measures under local forms and the dispatching rule comparison OPSLK vs. SLACK. For example, within penalty measures under local allowance forms and 90%

Table 5.5

Hypothesis 4 Pairwise Tests
Operation vs Job Based Dispatching P-Values[1]

Under Stated Dispatching/ Utilization/ Processing Time Assumptions: H_0: EOPDD ≥ EDD; H_1: EOPDD < EDD	VAR	MSL	MAL	SQL
Local/ 75%/ Known	.0002	.0002	.0005	.0310
Local/ 75%/ Unknown	<.00005	<.00005	.0005	.0007
Local/ 90%/ Known	.0677	.6174	.7134	.0727
Local/ 90%/ Unknown	.1659	.0727	.1354	.9924
Global/ 75%/ Known	<.00005	<.00005	<.00005	<.00005
Global/ 75%/ Unknown	<.00005	<.00005	<.00005	<.00005
Global/ 90%/ Known	<.00005	.0001	<.00005	.0007
Global/ 90%/ Unknown	<.00005	.0002	.0001	.0007

H_0: OPSLK ≥ SLACK; H_1: OPSLK < SLACK	VAR	MSL	MAL	SQL
Local/ 75%/ Known	.0366	.0780	.3075	.2878
Local/ 75%/ Unknown	.0430	.0957	.3614	.9817
Local/ 90%/ Known	.3006	.6863	.7310	.9964
Local/ 90%/ Unknown	.1953	.6727	.7609	.9960
Global/ 75%/ Known	.0013	.0011	.0006	.0003
Global/ 75%/ Unknown	.0013	.0016	.0031	.0080
Global/ 90%/ Known	.0003	.0008	.0008	.0008
Global/ 90%/ Unknown	<.00005	.0002	<.00005	.0008

H_0: OPCR ≥ CR; H_1: OPCR < CR	VAR	MSL	MAL	SQL
Local/ 75%/ Known	<.00005	<.00005	.0004	.0007
Local/ 75%/ Unknown	.0001	<.00005	.0040	.0008
Local/ 90%/ Known	.3006	.0500	.9389	.9603
Local/ 90%/ Unknown	.8520	.5149	.6950	.7942
Global/ 75%/ Known	.0365	.0630	.2711	.3476
Global/ 75%/ Unknown	.0040	.0039	.8570	.0016
Global/ 90%/ Known	.0084	.0284	.2005	.0085
Global/ 90%/ Unknown	.0045	.0166	.2333	.0014

[1] One Tailed

utilization, operation-based rules produce significantly (α = .05) lower penalties in 0 out of 6 cases, in 1 out of 6 cases, in 0 out of 6 cases, and in 0 out of 6 cases for VAR, MSL, MAL, and SQL, respectively. In short, while overall benefits of operation-based rules are significant, several specific assumption combinations (primarily under local allowance forms) yield no significant benefits.

Hypothesis 5. Hypothesis 5 proposes that benefits produced by incorporating global variables in allowance equations are greater under conditions of 90% expected utilization than under conditions of 75% expected utilization. The coefficients for the interaction variable "global*high" are negative and significant in all four regression analyses (see Table 5.3). This indicates that within each of the four measures, the differences in penalties produced by global vs. local forms under conditions of 90% utilization are significantly greater than the differences in penalties produced by global vs. local forms under conditions of 75% utilization.

The Wilcoxon tests support the regression conclusions. Pairwise evaluation of this hypothesis entails the statistical comparison of two sets of relative performances: under each appropriate combination of assumptions, the relative performance of local vs. global forms under 75% utilization against the relative performance of local vs. global forms under 90% utilization. As previously discussed, penalty effects of any given assumption change tend to be proportional, as opposed to scalar, in nature (hence the use of the logarithm transformation in the regression analyses). Therefore, the relative performances mentioned above are determined in terms of the ratios of local to global penalties.

Accordingly, the Wilcoxon procedure tests the local/global ratios under 75% utilization vs. the local/global ratios under 90% utilization for each of 12 possible comparisons. For example, for the dispatching rule EDD and known actual processing times, the 20 observed values of [(local under 75%)/(global under 75%)] are tested against the matched values of [(local under 90%)/(global under 90%)]. If the effects proposed in Hypothesis 5 exist, one would expect the second ratio to be significantly larger than the first ratio since the marginal benefits of global forms are hypothesized to be larger under 90% utilization than under 75% utilization.

Table 5.6 displays the p values associated with each Wilcoxon test, for each of the 12 possible comparisons, within each penalty measure. The results show that within VAR, MSL, MAL, and SQL, the second ratio is significantly larger (α = .05) than the first ratio in 7 out of 12 cases, in 10 out of 12 cases, in 10 out of 12 cases, and in 6 out of 12 cases, respectively. Under operation-based dispatching rules, the second ratio is significantly larger than the first in all cases.

Hypothesis 6. Hypothesis 6 proposes that penalties produced under the assumption that actual processing times are known upon a job's arrival at the shop (i.e., assumed to be equal to the expected processing times) are lower than penalties produced under the assumption that actual processing times are unknown upon a job's arrival at the shop (i.e., allowed to vary stochastically about expected processing times). The observed coefficients for the variable "Unknown" are

Table 5.6

Hypothesis 5 Pairwise Tests
90% Utilization-Global Interaction P-Values[1]

Under Stated Dispatching/ Processing Time Assumptions:	VAR	MSL	MAL	SQL
EDD/ Known	.5691	.8666	.0010	.0777
EDD/ Unknown	.8254	.0250	.0016	.9950
SLACK/ Known	.5777	.1025	.0139	.9982
SLACK/ Unknown	.5446	.0337	.0061	.9989
CR/ Known	.7953	.0261	.8895	.8844
CR/ Unknown	.0702	.0495	.1216	.7248
EOPDD/ Known	.0001	<.00005	.0008	.0149
EOPDD/ Unknown	<.00005	<.00005	<.00005	<.00005
OPSLK/ Known	<.00005	<.00005	<.00005	.0007
OPSLK/ Unknown	<.00005	<.00005	<.00005	.0011
OPCR/ Known	.0002	.0003	.0096	.0002
OPCR/ Unknown	<.00005	.0002	.0024	.0007

[1] H_0: [Local/Global Ratio, 75% Utilization] ≥ [Local/Global Ratio, 90% Utilization]

H_1: [Local/Global Ratio, 75% Utilization] < [Local/Global Ratio, 90% Utilization] positive and significant in the VAR, MSL, and MAL regressions; in the SQL regression, this coefficient is not significant. The regression analyses therefore provide statistically significant support for Hypothesis 6 for three out of the four penalty measures.

Table 5.7 shows the p values associated with the Wilcoxon tests of known vs. unknown for each of the 24 possible comparisons within each penalty measure. These results support the regression conclusions. Within VAR, MSL and MAL, the penalties produced under the assumption of known actual processing times are significantly lower (α = .05) than those produced under the assumption of unknown actual processing times in 12 out of 24 cases, in 10 out of 24 cases, and in 16 out of 24 cases, respectively.

For the measure SQL, the penalties produced under the assumption of known actual processing times are significantly lower (α = .05) than those produced under the assumption of unknown actual processing times in 5 out of 24 cases. However, the penalties produced under the assumption of known actual processing times are significantly <u>higher</u> (α = .05) than those produced under the assumption of unknown actual processing times in 8 out of 24 cases. In no comparison within VAR, MSL, or MAL were "known" penalties significantly higher than "unknown" penalties.

Recall that the measure SQL is very sensitive to observed latenesses. While the effects of observed mean latenesses are specifically accounted for in the SQL regression analysis, they are not addressed in the SQL Wilcoxon series. Apparently, systematic effects of observed mean latenesses are producing significant differences in both directions in the Wilcoxon SQL tests.

Table 5.7

Hypothesis 6 Pairwise Tests
Known Processing Time vs Unknown Processing Time P-Values[1]

Known vs Unknown Under Stated Dispatching/ Allowance/ Utilization Assumptions	VAR	MSL	MAL	SQL
EDD/ Local/ 75%	.9370	.9043	.0292	.3271
EDD/ Local/ 90%	.4260	.0590	.0542	.9690
EDD/ Global/ 75%	.0114	.0138	.0054	.5886
EDD/ Global/ 90%	.8910	.2165	.2878	.0004
SLACK/ Local/ 75%	.3137	.3969	.1090	.9974
SLACK/ Local/ 90%	.2752	.0727	.1802	.5446
SLACK/ Global/ 75%	.0630	.0836	.0132	.9874
SLACK/ Global/ 90%	.1022	.2752	.6450	.9952
CR/ Local/ 75%	.1754	.9323	.0159	.9848
CR/ Local/ 90%	.7432	.8605	.2391	.3969
CR/ Global/ 75%	.0009	.0009	.0002	.0429
CR/ Global/ 90%	.0261	.0229	.0121	.0585
EOPDD/ Local/ 75%	.2277	.2058	.0255	.9991
EOPDD/ Local/ 90%	.0152	.3969	.4260	<.00005
EOPDD/ Global/ 75%	.0002	.00025	<.00005	.0188
EOPDD/ Global/ 90%	.0002	.0003	.0003	.9891
OPSLK/ Local/ 75%	.0365	.1590	.0243	.9950
OPSLK/ Local/ 90%	.0542	.1661	.1659	.8614
OPSLK/ Global/ 75%	.0002	.0002	.0001	.0590
OPSLK/ Global/ 90%	.0200	.0032	.0119	.0003
OPCR/ Local/ 75%	.0001	<.00005	<.00005	.0836
OPCR/ Local/ 90%	.0062	.0094	.0006	.9273
OPCR/ Global/ 75%	.0001	.0002	.0002	.7942
OPCR/ Global/ 90%	.1568	.8148	.0219	.6796

[1] One tailed H_0: Known $\geq$ Unknown
H_1: Known < Unknown

Comparisons to Ragatz and Mabert Study

As previously discussed, Ragatz and Mabert (1984) recommend a global allowance form termed RMR for use under the SLACK dispatching rule (termed MINSLK in their study). Their proposed allowance form is noted in Chapter 3 of this current study as Equation 3.1.

The RMR equation contains variables that are job-based (P_i, JIS_i, and JIQ_i) and operation-based ($WIQ1_i$, $WIQ2_i$, and $WIQ3_i$). The specific equation coefficients are based on a regression analysis of results from a single pilot simulation (equivalent to Cycle 1 values in this current research). The global form recommended in this current research contains only job-based variables (TPT_i, $TWIQ_i$, $TWISM_i$, TPT_i^2, and $TWIQ_i^2$), with coefficients based on the sixth cycle of an iterative simulation-regression procedure.

Table 5.8 displays the median penalty measures and Wilcoxon p values associated with direct comparisons of the global allowance form and coefficient determination procedures recommended by Ragatz and Mabert (RMR) vs. those recommended in this current research (GEE). Comparisons are made for each of the four possible combinations of utilization level and actual processing time assumptions, under the dispatching rule SLACK that is common to both studies.

Within the measures VAR, MSL, MAL, and SQL, GEE produces significantly lower penalties than does RMR in 4 out of 4 cases, in 4 out of 4 cases, in 4 out of 4 cases, and in 2 out of 4 cases, respectively. In 13 out of the 14 cases in which GEE produces significant benefits, the p values associated with the tests are <.00005 (in the 14th case the p value is .0018). These results provide

Table 5.8

Pairwise Wilcoxon Tests of GEE (Cycle 6) vs RMR (Cycle 1)
Median Penalty Measures and P-Values[1]
(Under SLACK Dispatching Rule)

GEE vs RMR Under Stated Utilization/ Act Proc Time Assumptions		VAR	MSL	MAL	SQL
90%/ Known	GEE Median	625	656	19.0	404
	RMR Median	948	1012	23.2	325
	P-Value	<.00005	<.00005	<.00005	.9634
90%/ Unknown	GEE Median	659	673	18.8	354
	RMR Median	1081	1168	24.8	349
	P-Value	<.00005	<.00005	<.00005	.5247
75%/ Known	GEE Median	237	239	11.4	153
	RMR Median	365	366	13.8	191
	P-Value	<.00005	<.00005	<.00005	<.00005
75%/ Unknown	GEE Median	251	251	11.7	128
	RMR Median	359	359	13.6	162
	P-Value	<.00005	<.00005	<.00005	.0018

[1] One tailed H_0: GEE ≥ RMR
H_1: GEE < RMR

compelling statistical evidence that the form and procedures recommended in this current research provide significant and meaningful improvements in terms of completion inaccuracy penalties over the Ragatz and Mabert form and procedures.

Summary

Global allowance policy forms produce significantly lower completion inaccuracy penalties than do local allowance policy forms (Hypothesis 3). Generally, operation-based dispatching rules produce significantly lower completion inaccuracy penalties than do job-based dispatching rules (Hypothesis 4), although specific combinations of assumptions exist where benefits are not significant.

The significant benefits produced by incorporating global variables in the allowance determination procedure are less under conditions of 75% expected utilization than under conditions of 90% expected utilization (Hypothesis 5). Generally, simulations run under the assumption that actual processing times are known upon a job's arrival at the shop produce significantly lower completion inaccuracy penalties than do simulations run under the assumption that actual processing times vary stochastically about expected times (Hypothesis 6).

The global allowance form and coefficient determination procedures recommended in this current research produce significantly lower completion inaccuracy penalties than do the global form and procedures recommended by Ragatz and Mabert (1984).

Chapter 6

Implications and Directions

The current research findings provide management of general job shops with directions for immediate benefits and for potential future benefits. This chapter discusses managerial implications in terms of benefits and costs of adopting recommended procedures, and then discusses directions of potentially valuable future research.

Managerial Implications

Implications of this research for shop management include both benefits and costs. The benefits are associated with guidance in selecting dispatching rules and optimal allowance policy forms, and providing procedures for determining specific allowance equations and operation due dates. The costs are associated with the studies and informational mechanisms necessary to support the improved methods. Further, the method of research itself (i.e., simulation of a shop based on a set of stated assumptions) may provide beneficial tools for management above and beyond dispatching rules and allowance policies.

Benefits

Previously cited surveys and studies such as Putnam *et. al.* (1971), Panwalker and Iskander (1973), Kanet and Hayya (1982), Baker (1984), and Hax and Candea (1984) state that the vast majority of existing job shop research has not addressed the objectives of the real world. This current research addresses real world needs by minimizing penalties associated with the dispersion of actual job completions about expected completions.

Concerning selection of dispatching rules, this research has supported and extended the conclusions of Kanet and Hayya (1982) that operation-based dispatching rules outperform job-based dispatching rules. Specifically, this current research indicates that the rule OPCR generally is superior to other dispatching rules evaluated.

The majority of existing job shop simulation-based research has assumed naive allowance policies and concentrated on evaluating the performances of various dispatching rules under given assumptions. This current research has concentrated on the development and evaluation of optimal allowance policies under different combinations of dispatching rules and shop assumptions. The benefits of incorporating global (i.e., shop congestion) variables into the allowance policy have been demonstrated.

An important point, though, is that while this research has demonstrated meaningful benefits inherent in extending beyond naive allowance methods, the global allowance forms used in this research have not been overly complex in nature. The two global variables incorporated in the recommended allowance form (the total work of operations in the queues of machines required by a given job, and the total work of operations elsewhere in the shop that require those same machines) are summary in nature and feasible to maintain in a real world shop.

Given an allowance policy form, this research has demonstrated the benefits of an iterative simulation-regression procedure for determining the specific allowance equation. Further, this research has demonstrated the advantages of setting operation due dates directly from the defined allowance equation rather than proportionally allocating total job allowances among operations.

The benefits offered by these recommendations are shown to be both statistically significant and meaningful in magnitude. The median percentage decreases in the observed penalty measures VAR, MSL, MAL, and SQL produced by operation-based dispatching rules (over job-based dispatching rules) were 39%, 40%, 23%, and 38%, respectively. The median percentage decreases in the observed penalty measures produced by global allowance policies (over local allowance policies) were 72%, 74%, 46%, and 64%, respectively. The median percentage decreases in the observed penalty measures produced by an iterative simulation-regression procedure (over a single simulation-regression procedure) were 47%, 50%, 26%, and 32%, respectively. The median percentage decreases in the observed penalty measures produced by direct estimation of operation due dates (as opposed to proportional allocation of total job allowances) were 29%, 26%, 11%, and 39%, respectively.

Costs

The costs inherent in adopting the procedures recommended by this research entail both computational costs associated with initial implementation and informational costs associated with maintenance and operation. As indicated by previous research (for example, Kanet, 1979), the choice of the optimal dispatching rule and the choice of the optimal allowance policy appear to be dependent. This current research (as well as logic) indicates that both may be dependent on specific shop characteristics. Implementation of recommended procedures, therefore, should occur on a shop-specific basis.

Envisioned implementation would require an initial simulation study based on the specific shop structure, management objectives, and relevant observed distributions. Management objectives would dictate the choice of the appropriate penalty measure to be minimized. The observed distributions would reflect job/ machine characteristics such as number of operations per job, operation-machine assignments, times between job arrivals, and operation service times (including setup and breakdown times).

The iterative simulation-regression procedure would produce an optimal combination of dispatching rule and specific allowance equation for the shop. A simpler procedure that would produce near-optimal results would be to adopt a generally superior dispatching rule such as OPCR and to simulate in order to specify only the optimal specific allowance policy to be used.

The initial study could be accomplished utilizing in-house programming and computing capabilities or external expertise such as a consultant. All required computing could be implemented feasibly on a personal computer with moderate speed and memory storage capacity. Conceivably, a general user-friendly software package could be developed and marketed to individual shops to provide sufficient capabilities for the initial study.

The on-going use of a global allowance policy defined by the initial study would entail an information structure to support its requirements. In short, certain shop congestion information would have to be maintained in order to provide necessary data to be used in the specific allowance equation. As mentioned previously, the shop congestion information required by the global allowance form recommended in this research consists of total work of operations in the queues of machines required by a given job, and total work of operations elsewhere in the shop that require those machines. This information could be maintained effectively with or without in-house computer capabilities.

Simulation as a Management Tool

With in-house computing capabilities and the proper software, the simulation procedures utilized in this research could provide shop management with beneficial directions beyond dispatching rules and allowance procedures. This current research indicates that as shop utilization decreases, penalty measures associated with inaccurate job completions decrease. In short, a shop can increase completion accuracy if it is willing to accept more machine idle time and lower asset utilization.

This suggests that, for a given shop, there may be a theoretically optimal combination of excess capacity and completion accuracy. Simulation studies based on specific shop and job characteristics could aid in defining such a point, and provide quantitative guidance to capacity expansion/contraction decisions. Similarly, simulation analyses could provide input into decisions concerning areas including shop balancing, shop layout, job pricing, and preventive maintenance.

Directions for Future Research

The majority of existing research has concentrated on simplistic shop scheduling algorithms due to a perceived lack of real world capabilities to implement more sophisticated ones. With the current availability of powerful and relatively inexpensive personal computer systems, increased computational sophistication is within the reach of even the smallest shops. Future research should not constrain itself based on limitations that no longer exist.

Perhaps the most pressing immediate need for further research lies in the application of theoretical procedures to real-world shops. Although instances of studies in actual shops have been cited (for example, Elmaghraby and Cole, 1963, and Bulkin, 1966), examples of applying proposed procedures to real-world shops are relatively scarce in the literature. Topics such as the indicated advantages of global allowance policies over local allowance policies, the indicated advantages of operation-based dispatching rules over job-based dispatching rules, and the external validity of simulation-based optimization procedures should be verified by researchers in real-world situations in order to facilitate the wide acceptance and adoption of recommended procedures.

This is not to say that further simulation research outside of existing shops would not be worthwhile. This current research and other relatively recent studies such as Baker and Bertrand (1981), Baker and Kanet (1983), Ragatz and Mabert (1984), and Bookbinder and Noor (1985) provide new foundations for potentially valuable research in a variety of areas.

One promising topic of future efforts is research on global allowance policies. The small number of previously cited studies (including this present research) that have examined the inclusion of shop congestion variables in the allowance determination procedure have all demonstrated significant benefits resulting from their proper inclusion. However, the topic is relatively new, and further research should be undertaken with a focus on the specific form and content of global allowance policies.

The benefits of moving from local allowance forms to global allowance forms that incorporate information about shop congestion as of the moment a job arrives at the shop have been established. Further benefits may be gained by moving from global information as of a job's arrival to expected global information in the near future. For example, when a job arrives at the shop, one may know with certainty that the second machine on its path will not be available for thirty more time units, and by the time it is available, two more jobs currently being processed on other machines will have joined the queue at that machine. Such certain or highly probable knowledge about near-term movements of the shop may provide valuable predictive information.

If a shop has computing power available, a further potentially beneficial step may be to start with the shop status as of the job's arrival, run a small simulation procedure through that job's simulated completion, and base the job's due date upon that simulated completion. Since virtually all of the better-performing dispatching rules have included job/ operation due dates in their selection prioritizations, this procedure would likely be an iterative one, where initial job/ operation due dates are set and then refined with each iterative stage.

Another promising area of future research concerns the development and evaluation of algorithms for preempting jobs. Gains in completion accuracy may be offered by the ability of shop management to interrupt the processing of a current operation in favor of another job's imminent operation. Virtually all existing simulation-based shop scheduling research has assumed that preemption is not allowed.

The topic of job expediting holds great promise for future research efforts, and the development and evaluation of job expediting algorithms should be undertaken. This could entail static expediting (for example, setting a high/ medium/ low priority to a job upon its arrival at the shop, with the job keeping the same rating throughout its stay in the shop) or dynamic expediting (changing the relative priorities of jobs during their stay in the shop). Actually, the use of dispatching rules to select jobs from queues is a mild form of dynamic expediting. This is a practice that must be considered a reality in actual shops (for example, receiving rush orders where the necessary due date is earlier than a constraint-free allowance policy would dictate or giving special treatment to a preferred customer) but has received little attention in past research.

The topic of dynamic expediting points to a related area of potential future research: the separate consideration of due dates stated to the customer upon a job's arrival at the shop and expected completion dates that are updated as the jobs move through the shop. In effect, many dispatching rules (such as SLACK) are based on this theme, as they take both the job due date and the remaining work required by the job into account. Potentially, improved scheduling algorithms could be developed by calculating an updated expected completion date for each job (based on the defined allowance policy) and basing the selection prioritization on the relationship between the updated expected completion date and the original expected completion date (that was stated to the customer).

Other potential areas of future research include extending this analysis to dual (i.e., machine and labor) constrained shops, imposing certain types of external allowance constraints (such as maximum mean flowtimes) on the simulation-regression procedures, and examining the effects of differential machine loading (for example, one machine may be used twice as much as other machines). While numerous promising new directions based on this current research undoubtedly exist, they must be undertaken with a recognition of the needs and limitations of real-world shops.

Summary

This current research has many direct implications for the management of real-world shops. Meaningful improvements to existing methods have been proposed and demonstrated. Implementation, however, would require efforts such as initial shop-specific simulation analyses and maintenance of global information necessary to optimal allowance policies. Simulation methods such as employed in this research could prove useful to shop management in areas other than job scheduling.

Numerous important areas for future research exist, based on the results of this current research and other relatively recent studies. Examples of such areas are the application of recommended improvements to real-world shops, further research into global allowance policies, operation preemption, and job expediting.

List of References

List of References

Baker, C. T. and Dzielinski, B. P. Simulation of a Simplified Job Shop. Management Science, April 1960.

Baker, K. R. Introduction to Sequencing and Scheduling. New York: Wiley, 1974.

Baker K. R. Sequencing Rules and Due-Date Assignments in a Job Shop. Management Science, 1984.

Baker, K. R. and Bertrand, J. M. W. An Investigation of Due Date Assignment Rules with Constrained Tightness. Journal of Operations Management, February 1981.

Baker, K. R. and Kanet, J. J. Job Shop Scheduling with Modified Due Dates. Journal of Operations Management, November 1983.

Baker, K. R. and Schrage, L. E. Finding an Optimal Sequence by Dynamic Programming: An Extension to Precedence-Related Tasks. Operations Research, January - February 1978.

Balas, E. Machine Sequencing Via Disjunctive Implicit Enumeration Algorithm. Operations Research, November - December 1969.

Berry, W. L. and Rao, V. Critical Ratio Scheduling: An Experimental Analysis. Management Science, October 1975.

Bookbinder, J. H. and Noor, A. I. Setting Job-Shop Due-Dates with Service Level Constraints. Journal of the Operational Research Society, November 1985.

Brooks, G. H. and White, C. R. An Algorithm for Finding Optimal or Near-Optimal Solutions to the Production Scheduling Problem. Journal of Industrial Engineering, January 1965.

Buffa, E. S. and Taubert, W. H. Production Inventory Systems: Planning and Control. Homewood, Ill: Irwin, 1972.

Bulkin, M. H., Colley, J. L., and Steinhoff, H. W. Load Forecasting, Priority Sequencing, and Simulation in a Job-Shop Control System. Readings in Production and Operations Management. New York: Wiley, 1966.

Burns, F. and Rooker, J. Three Stage Flow Shops with Recessive Second Stage. Operations Research, January - February 1978.

Campbell, H. G., Dudek, R. A., and Smith, M. L. A Heuristic Algorithm for the n Job m Machine Sequencing Problem. Management Science, June 1970.

Carroll, D. C. Heuristic Sequencing of Single and Multiple Component Jobs. Unpublished Doctoral Thesis. M. I. T., 1965.

Coffman, E. G. Computer and Job Shop Scheduling Theory. New York: Wiley, 1976.

Conover, W. J. Practical Nonparametric Statistics. New York: Wiley, 1971.

Conway, R. W. Priority Dispatching and Work in Process Inventory in a Job Shop. Journal of Industrial Engineering, March - April 1965a.

Conway, R. W. Priority Dispatching and Job Lateness in a Job Shop. Journal of Industrial Engineering, July - August 1965b.

Conway, R. W., Johnson, B. M., and Maxwell, W. L. An Experimental Investigation of Priority Dispatching. Journal of Industrial Engineering, May - June 1960.

Conway, R. W. and Maxwell, W. L. Network Dispatching by the Shortest Operation Discipline. Operations Research, January - February 1962.

Conway, R. W., Maxwell, W. L., and Miller, L. W. Theory of Scheduling. Reading, Mass: Addison-Wesley, 1967.

√Dannenbring, D. G. An Evaluation of Flow Shop Scheduling Heuristics. Management Science, July 1977.

Day, J. E. and Hottenstein, M. P. Review of Sequencing Research. Naval Research Logistics Quarterly, March 1970.

Eilon, S. and Chowdhury, I. G. Due Dates in Job Shop Scheduling. International Journal of Production Research, 1976.

Eilon, S. and Hodgson, R. M. Job Shop Scheduling with Due Dates. International Journal of Production Research, 1967.

Elmaghraby, S. E. The Machine Sequencing Problem - Review and Extensions. Naval Research Logistics Quarterly, June 1968.

Elmaghraby, S. E. and Cole, R. T. On the Control of Production in Small Job Shops. Journal of Industrial Engineering, July - August 1963.

Elvers, D. A. Job Shop Dispatching Rules Using Various Delivery Date Setting Criteria. Production and Inventory Management, 1973, 4th Q.

Elvers, D. A. The Sensitivity of the Relative Effectiveness of Job Shop Dispatching Rules with Respect to Various Arrival Distributions. AIIE Transactions, March 1974.

Fife, D. W. Scheduling with Random Arrivals and Linear Loss Functions. Management Science, January 1965.

Fisher, M. L. Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I. Operations Research, September - October 1973.

Fisher, M. L. A Dual Algorithm for the One-Machine Scheduling Problem. Mathematical Programming, December 1976.

Fry, T. D., Philipoon, P. R., and Markland, R. E. An Investigation of Due Date Assignment Rules in a Multi-Stage Production Environment. Working Paper, University of South Carolina, January 1987.

Garey, M. R., Johnson, D. S., and Sethi, R. Complexity of Flow Shop and Job Shop Scheduling. Mathematics of Operations Research, May 1976.

Gere, W. S. Heuristics in Job Shop Scheduling. Management Science, November 1966.

Giglio, R. J. and Wagner, H. M. Approximate Solutions to the Three-Machine Scheduling Problem. Operations Research, March - April 1964.

Gonzalez, M. J. Deterministic Processor Scheduling. Computing Surveys, September 1977.

Hax, A. and Candea, D. Production and Inventory Management. Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

Ignall, E. J. and Schrage, L. E. Application of the Branch and Bound Technique to Some Flow Shop Scheduling Problems. Operations Research, May - June 1965.

Jackson, J. R. An Extension of Johnson's Results on Job Lot Scheduling. Naval Research Logistics Quarterly, September 1956.

Jackson, J. R. Job Shop Like Queueing Systems. Management Science, October 1963.

Johnson, S. M. Optimal Two and Three Stage Production Schedules with Setup Times Included. Naval Research Logistics Quarterly, March 1954.

Kanet, J. J. Job Shop Scheduling to Meet Due Dates: A Simulation Study. Published Doctoral Dissertation. The Pennsylvania State University, 1979.

Kanet, J. J. and Hayya, J. C. Priority Dispatching with Operation Due Dates in a Job Shop. Journal of Operations Management, May 1982.

Le Grande, E. The Development of a Factory Simulation Using Actual Operating Data. Management Technology, May 1963.

Lemoine, A. J. Networks of Queues - A Survey of Equilibrium Analysis. Management Science, December 1977.

Lenstra, J. L., Rinnooy Kan, A. H. G., and Brucker, P. Complexity of Machine Scheduling Problems. Annals of Discrete Mathematics, 1977.

Manne, A. S. On the Job Shop Scheduling Problem. Operations Research, March - April 1960.

McNaughton, R. Scheduling with Deadlines and Loss Functions. Management Science, October 1959.

Melnyk, S. A., Vickery, S. K., and Carter, P. L. Scheduling, Sequencing, and Dispatching: Alternative Perspectives. Production and Inventory Management, 2nd Q. 1986.

Moore, J. M. and Wilson, R. C. A Review of Simulation Research in Job Shop Scheduling. Production and Inventory Management, 1967, Vol. 8.

Naylor, T. H., Balintfy, J. L., Burdick, D. S., and Kong, C. Computer Simulation Techniques. New York: Wiley and Sons, 1966.

Nanot, Y. R. An Experimental Investigation and Comparative Evaluation of Priority Disciplines in Job Shop Like Queuing Networks. Unpublished Doctoral Dissertation, UCLA, 1963.

New, C. C. Job Shop Scheduling: Who Needs a Computer to Sequence Jobs? Production and Inventory Management, 1975, 4th Q.

Panwalker, S. S. and Iskander, W. A Survey of Scheduling Rules. Operations Research, January - February 1977.

Picard, J. C. and Queyranne, M. The Time Dependent Travelling Salesman Problem and Its Application to the Tardiness Problem in One Machine Scheduling. Operations Research, January - February 1978.

Putnam, A. D., Everdell, R., Dorman, G. H., Cronan, R. R., and Lindgren, L. H. Updating Critical Ratio and Slack Time priority Scheduling Rules. Production and Inventory Management, 1971.

Ragatz, G. L. and Mabert, V. A. A Simulation Analysis of Due Date Assignment Rules. Journal of Operations Management, November 1984.

Rinnooy Kan, A. H. G., Lageweg, B. J., and Lenstra, J. K. Minimizing Total Costs in One Machine Scheduling. Operations Research, September - October 1975.

Rothkopf, M. Scheduling Independent Tasks on Parallel Processors. Management Science, January 1966.

Salvador, M. S. Scheduling and Sequencing. In Moder, J. J. and Elmaghraby, S. E. Handbook of Operations Research: Models and Applications. New York: Van Nostrand Reinhold, 1978.

Schild, A. and Fredman, I. J. Scheduling Tasks with Deadlines and Non-Linear Loss Functions. Management Science, October 1962.

Schrage, L. E. A Proof of the Optimality of the Shortest Service Time Discipline. Operations Research, May - June 1968.

Shwimer, J. On the n-Job, One Machine, Sequence Independent Scheduling Problem with Tardiness Penalties: A Branch and Bound Approach. Management Science, February 1972.

Sidney, J. B. Optimal Single Machine Scheduling with Earliness and Tardiness Penalties. Operations Research, January - February 1977.

Smith, D. R. A New Proof of the Optimality of the Shortest Remaining Processing Time Discipline. Operations Research, January - February 1978.

Smith, M. C. and Seidmann, A. Due Date Selection Procedures for Job Shop Simulation. Computers and Industrial Engineering, 1983.

Smith, W. E. Various Optimizers for Single State Production. Naval Research Logistics Quarterly, March 1956.

Srinivasan, V. A Hybrid Algorithm for the One Machine Sequencing Problem to Minimize Total Tardiness. Naval Research Logistics Quarterly, September 1971.

Story, A. E. and Wagner, H. M. Computational Experience with Integer Programming for Job Shop Scheduling. In Muth, J. F. and Thompson, G. L. Industrial Scheduling. Englewood Cliffs, New Jersey: Prentice-Hall, 1963.

Weeks, J. K. A Simulation Study of Predictable Due Dates. Management Science, April 1979.

Weeks, J. K. and Fryer, J. S. A Methodology for Assigning Minimum Cost Due Dates. Management Science, April 1977.

Wilcoxon, F. Some Rapid Approximate Statistical Procedures. American Cyanamid Company, Stamford Research Laboratories, 1949.

Appendix A

Glossary of Variables and Acronyms

a_i	The total allowance assigned to job i.
a_{ik}	The total allowance assigned to the k^{th} operation of job i.
C_i	The actual completion time of job i.
CON	An allowance policy that assigns a constant allowance to each job.
CR	The dispatching rule that selects from the queue the job with the lowest job based critical ratio (see page 16).
d_i	The due date assigned to job i.
d_{ik}	The due date assigned to the k^{th} operation of job i.
EDD	The dispatching rule that selects from the queue the job with earliest due date.
EFT_i	The early finish time of job i (see page 58).
EOPDD	The dispatching rule that selects from the queue the job that has the earliest pending operation due date.
F_i	The total time a job spends in the shop (flow time).
FCFS	The dispatching rule that selects from the queue the job with the earliest arrival at the shop.
GEE	The global allowance policy recommended in this current research with aggregate variables and coefficients based on the sixth cycle of an iterative simulation-regression procedure.
JIQ_i	The number of jobs in the queues of machines required by job i as of its arrival at the shop.
JIS_i	The number of jobs in the shop as of the arrival of job i at the shop.
L_i	The observed lateness of job i (see page 4).
LPT	The dispatching rule that selects from the queue the job with the largest pending operation processing time.

MAL The mean absolute lateness penalty measure (see page 5).

MINSLK The dispatching rule equivalent to SLACK.

MSL The mean squared lateness penalty measure (see page 5).

NOP The allowance policy that assigns an allowance to each job that is a multiple of the number of operations in the job.

OPCR The dispatching rule that selects from the queue the job that has the smallest pending operation critical ratio (see page 17).

OPNDD A dispatching rule that equivalent to EOPDD.

OPSLK The dispatching rule that selects from the queue the job that has the smallest pending operation slack (see page 16).

OPSLK/P The dispatching rule that selects from the queue the job that has the smallest ratio of pending operation slack to pending operation required processing time.

p_i The actual processing time required by job i.

$\hat{p}_i$ The processing time expected to be required by job i as of its arrival at the shop.

p_{ik} The actual processing time required by the k^{th} operation of job i.

$\hat{p}_{ik}$ The processing time expected to be required by the k^{th} operation of job i as of its arrival at the shop.

PPW The allowance policy that assigns a job allowance equal to the total required processing time plus a constant times the number of operations.

r_i The time that job i arrives at the shop.

RDM An allowance policy that assigns a random total allowance to each job as of its arrival at the shop.

RMR The global allowance policy recommended by Ragatz and Mabert consisting of both aggregated and operation specific variables and coefficients determined from a regression analysis of the results of a single pilot simulation.

SLACK The dispatching rule that selects from the queue the job with the smallest slack (time to due date less remaining required processing time).

S/OPN The dispatching rule that selects from the queue the job that has the smallest ratio of slack to number of remaining operations.

SPT The dispatching rule that selects from the queue the job that has the smallest pending operation required processing time.

SQL The semi-quadratic lateness penalty measure (see page 5).

TPT_i The total processing time expected to be required by job i.

$TWIQ_i$ The total expected required processing time of imminent operations in the queues of machines required by job i as of the arrival of job i at the shop.

$TWISM_i$ The total expected required processing time of pending but not imminent operations in the shop that require the machines required by job i as of its arrival at the shop.

TWK The allowance policy that assigns an allowance equal to a multiple of the job's expected total processing time.

VAR The variance penalty measure (see page 5).

w_i The actual amount of time job i spends waiting in queues.

$\hat{w}_i$ The amount of time that job i is expected to spend waiting in queues as of its arrival at the shop.

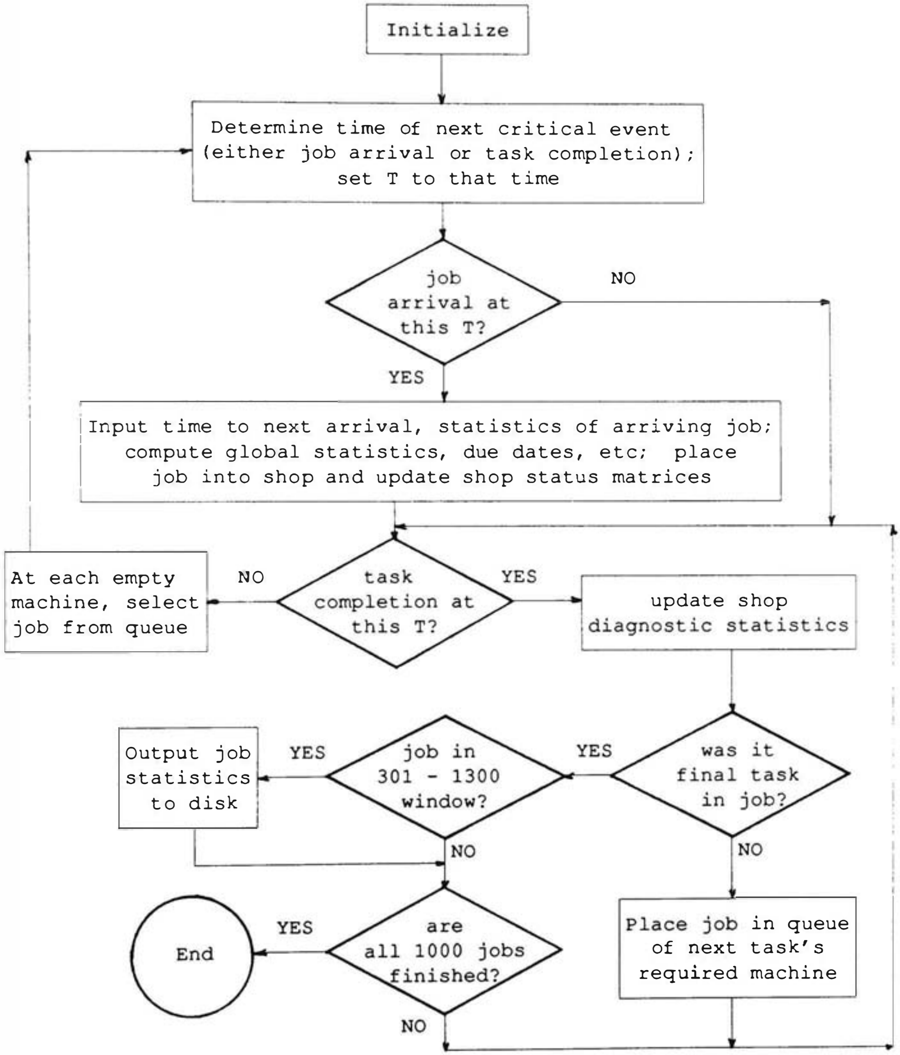
$WIQ1_i$ The total processing time of imminent operations in the queue of the machine required by the first operation of job i.

$WIQ2_i$ The total processing time of imminent operations in the queue of the machine required by the second operation of job i.

$WIQ3_i$ The total processing time of imminent operations in the queue of the machine required by the third operation of job i.

Appendix B

Summary Flow Logic of Main Simulation Program



Appendix B (continued)

Main Simulation Program Code
Global Allowance Policy, EDD Dispatching Rule

```
      implicit integer (a-y)
c    jobmat(110,65) is matrix showing 65 items of jobs in shop
c      Item 1: Job #
c           2: # tasks in job
c           3: current task #
c           4-9: arrival time of job at task 1, 2, etc.
c           10-15: machine required by task 1, 2, etc.
c           16-21: expected processing time of task 1, 2, etc.
c           22-27: actual processing time of task 1, 2, etc.
c           28-33: due date of task 1, 2, etc.
c           34-39: actual completion date of task 1, 2, etc.
c           40-45: # jobs in queue of machine required
c                  by task 1, 2, etc. as of job arrival
c                  at shop
c           46-51: total expected processing time of tasks in
c                  queue at machine required by task 1, 2, etc.
c                  as of job arrival at shop
c           52-57: # of tasks elsewhere in shop requiring machine
c                  required by task 1, 2, etc. as of job arrival
c                  at shop
c           58-63: total expected processing time of tasks elsewhere
c                  in shop requiring machine required by task 1,
c                  2, etc. as of job arrival at shop
c           64: # of tasks in shop as of job arrival at shop
c           65: total expected processing time of tasks in shop
c               as of job arrival at shop
c
c
c    macmat(8,80) is matrix of queues of machines 1-8
c       Item 1: # jobs at machine
c            2: slot (column) containing jobmat row number of
c               current job in progress at machine
c            3-80: jobmat row numbers of jobs in queue
c
c
c    jobst.* is file of data on the *th jobstream (21 items per job)
c       Item 1: job #
c            2: time to next job arrival
c            3: # of tasks in this job
c            4-9: machine required by task 1, 2, etc.
c            10-15: expected processing time of task 1, 2, etc.
c            16-21: actual processing time of task 1, 2, etc.
c
```

```
c
c    simres.a is output file of 65 items on jobs 301-1300
c    pload.d is input file of snapshot of shop (preload)
c    zcoeff.d is input file of coefficients to use in allowance equation
c    reginp.a is output file containing items pertinent to regression
c             analysis of output
c
c
      dimension jobmat(110,65), macmat(8,80), numsum(8), ptsum(8)
      dimension twka(8), twkq(8), maxq(8),zcoef(8),reg(11)
      open (unit=9, file= 'pload.d')
      open (unit=10, file= 'jobst.0')
      open(unit=11, file= 'simres.a')
      open(unit=12,file='zcoeff.d')
      open(unit=13,file='reginp.a')
c    read allowance equation coefficients
      do 864 i=1,8
      read(12,1114) zcoef(i)
864   continue
1114  format(f10.6)
c
c
c    totfin: # of jobs in 301-1300 window finished
c    tbeg: time that first job in window finishes
c    maxj: maximum # of jobs in shop
c    jbsm, msum, empsum: summary variables to be used in further
c                        calculations
c
c
      totfin=0
      tbeg=0
      maxj=0
      jbsm=0
      msum=0
      empsum=0
1111  format(1x,'max queue(',i1,') was ',i3)
1112  format(/'max shop load was ',i3,' jobs')
1113  format(/4i7)
900   format(i4)
1000  format(i5)
c    preload macmat, jobmat, etc.
      do 50 i=1,8
      do 51 j=1,80
      read(9,1000) macmat(i,j)
51    continue
50    continue
      do 52 i=1,110
      do 53 j=1,65
      read(9,1000) jobmat(i,j)
53    continue
52    continue
      read(9,1000) t
      do 54 i=1,8
      maxq(i)=0
```

```
54    read(9,1000) numsum(i)
      do 55 i=1,8
55    read(9,1000) ptsum(i)
c   read job # of first job in stream and time to next arrival
      read(10,1000) jobnum
      read(10,900) tnj
      njarr=t+tnj
c   find time of next task completion
999   ntskc=99999999
      do 100 i=1,8
      if ((macmat(i,1).eq.0).or.(macmat(i,2).eq.0)) go to 100
      jrow = macmat(i,macmat(i,2))
      curtsk = jobmat(jrow,3)
      if (jobmat(jrow,curtsk+33).lt.ntskc) ntskc=jobmat(jrow,curtsk+33)
100   continue
      minev=ntskc
c   find time of next critical event: min(next arr, next task comp)
      if(njarr.lt.minev) minev=njarr
      tdelt=minev-t
c   if not in 301-1300 window, don't augment summary statistics
      if (tbeg.eq.0) go to 103
      snap=0
      macbus=0
      do 629 jj=1,8
      if(macmat(jj,1).gt.maxq(jj)) maxq(jj)=macmat(jj,1)
      if(macmat(jj,1).gt.0) macbus=macbus+1
629   snap=snap+macmat(jj,1)
      if(snap.eq.0) empsum=empsum+tdelt
      if(snap.gt.maxj) maxj=snap
      msum=msum+tdelt*macbus
      jbsm=jbsm+tdelt*snap
c   update t; if no task comp at this time, branch to job arrival
    section
103   t=minev
      if(ntskc.gt.t) go to 760
c   next 20 lines of code adjusts shop for any tasks ending at this time
      do 750 i=1,8
      if ((macmat(i,1).eq.0).or.(macmat(i,2).eq.0)) go to 750
      jrow=macmat(i,macmat(i,2))
      curtsk=jobmat(jrow,3)
      if (jobmat(jrow,curtsk+33).gt.t) go to 750
      macmat(i,1)=macmat(i,1)-1
      numsum(i)=numsum(i)-1
      ptsum(i)=ptsum(i)-jobmat(jrow,curtsk+15)
      macmat(i,macmat(i,2))=0
      macmat(i,2)=0
      if(jobmat(jrow,2).eq.curtsk) go to 670
      ntask=curtsk+1
      jobmat(jrow,3)=ntask
      jobmat(jrow,ntask+3)=t
      nmach=jobmat(jrow,ntask+9)
      macmat(nmach,1)=macmat(nmach,1)+1
      maccol=2
630   maccol=maccol+1
```

```
      if (macmat(nmach,maccol).gt.0) go to 630
      macmat(nmach,maccol)=jrow
      go to 750
c   if completed job is not in window, skip output section
670   if((jobmat(jrow,1).le.300).or.(jobmat(jrow,1).gt.1300)) go to 730
      if(tbeg.eq.0) tbeg=t
c   next 16 lines outputs data on finished jobs to files
      totfin=totfin+1
      write(11,1100) (jobmat(jrow,ii), ii=1,65)
1100  format(i4,2i2,6i6,6i2,12i3,12i6,6i3,6i5,6i4,6i5,i4,i5)
      do 1020 kk=1,8
1020  reg(kk)=0
      reg(1)=jobmat(jrow,jobmat(jrow,2)+33)-jobmat(jrow,4)
      do 1021 kk=1,jobmat(jrow,2)
      reg(2)=reg(2)+jobmat(jrow,kk+15)
      reg(3)=reg(3)+jobmat(jrow,kk+45)
1021  reg(4)=reg(4)+jobmat(jrow,kk+57)-jobmat(jrow,kk+45)
      reg(5)=jobmat(jrow,65)*jobmat(jrow,2)
      reg(6)=reg(2)*reg(2)
      reg(7)=reg(3)*reg(3)
      reg(8)=reg(4)*reg(4)
      write(13,3579) (reg(kk),kk=1,8)
3579  format(8i7)
c   zero slots vacated by completed job; if all 1000 jobs completed, go
    to summary output and end
730   do 101 ii=1,65
101   jobmat(jrow,ii)=0
      if(totfin.eq.1000) go to 2222
750   continue
c   if no job arrival at this time, skip to select section
760   if(njarr.gt.t) go to 770
c   next 88 lines of code read new arrival characteristics, place job
    into shop, and update shop status
      jrow=0
210   jrow=jrow+1
      if(jobmat(jrow,1).gt.0) go to 210
      jobmat(jrow,1)=jobnum
      read(10,900) jobmat(jrow,2)
      nop=jobmat(jrow,2)
      do 80 i=1,nop
80    read(10,900) jobmat(jrow,i+9)
      do 81 i=1,nop
81    read(10,900) jobmat(jrow,i+15)
      do 82 i=1,nop
      jobmat(jrow,i+39)=macmat(jobmat(jrow,i+9),1)
82    read(10,900) jobmat(jrow,i+21)
      jobmat(jrow,3)=1
      opsum=0
      twksum=0
      read(10,1000) jobnum
      read(10,1000) tnj
      njarr=t+tnj
      do 90 macrow=1,8
      numq=0
```

```
      twka(macrow)=0
      twkq(macrow)=0
      if(macmat(macrow,1).eq.0) go to 89
      maccol=2
111   maccol=maccol+1
      jinbin=macmat(macrow,maccol)
      if(jinbin.eq.0) go to 111
      numq=numq+1
      tem=jobmat(jinbin,3)
      twkq(macrow)=twkq(macrow)+jobmat(jinbin,tem+15)
      if(numq.lt.macmat(macrow,1)) go to 111
      if(macmat(macrow,2).eq.0) go to 89
      jinbin=macmat(macrow,macmat(macrow,2))
      tem=jobmat(jinbin,3)
      if(jobmat(jinbin,tem+33).eq.0) go to 117
      twka(macrow)=t-jobmat(jinbin,tem+33)+jobmat(jinbin,tem+21)
117   temp=jobmat(jinbin,tem+15)
      if(twka(macrow).gt.temp) twka(macrow)=temp
      twkq(macrow)=twkq(macrow)-twka(macrow)
89    opsum=opsum+numsum(macrow)
      twksum=twksum+ptsum(macrow)-twka(macrow)
90    continue
      do 91 i=1,nop
      tem=jobmat(jrow,i+9)
      jobmat(jrow,i+45)=twkq(tem)
      jobmat(jrow,i+51)=numsum(tem)
91    jobmat(jrow,i+57)=ptsum(tem)-twka(tem)
      do 92 i=1,nop
      tem=jobmat(jrow,i+9)
      numsum(tem)=numsum(tem)+1
      ptsum(tem)=ptsum(tem)+jobmat(jrow,i+15)
92    continue
      jobmat(jrow,64)=opsum
      jobmat(jrow,65)=twksum
c *************************************
c *                                   *
c *   due date setting goes here      *
c *                                   *
      tpt=0
      twiq=0
      twism=0
      zt=t
      ztwis=jobmat(jrow,65)
      do 94 i=1,nop
      ztwisi=ztwis*i
      tpt=tpt+jobmat(jrow,i+15)
      twiq=twiq+jobmat(jrow,i+45)
      twism=twism+jobmat(jrow,i+57)-jobmat(jrow,i+45)
      ztpt=tpt
      ztwiq=twiq
      ztwism=twism
      z=zt+zcoef(1)+zcoef(2)*ztpt+zcoef(6)*ztpt*ztpt
      z=z+zcoef(3)*ztwiq+zcoef(7)*ztwiq*ztwiq
      z=z+zcoef(4)*ztwism+zcoef(8)*ztwism*ztwism
```

```
      jobmat(jrow,i+27)=z+zcoef(5)*ztwisi+.5
      if (jobmat(jrow,i+27).lt.(t+tpt)) jobmat(jrow,i+27)=tpt+t
94    continue
c *                                    *
c ************************************
      jobmat(jrow,4)=t
      macrow=jobmat(jrow,10)
      macmat(macrow,1)=macmat(macrow,1)+1
      maccol=2
112   maccol=maccol+1
      if(macmat(macrow,maccol).gt.0) go to 112
      macmat(macrow,maccol)=jrow
c   poll machines; where unoccupied, select next job from queue and
    intiate
c ************************************
c *    select from queue goes here    *
c *               EDD                 *
c *                                   *
770   do 790 i=1,8
      if((macmat(i,2).gt.0).or.(macmat(i,1).eq.0)) go to 790
      tem=99999999
      maccol=2
      numq=0
795   maccol=maccol+1
      if(numq.ge.macmat(i,1)) go to 789
      if(macmat(i,maccol).eq.0) go to 795
      numq=numq+1
      jrow=macmat(i,maccol)
      numtsk=jobmat(jrow,2)
      if(jobmat(jrow,numtsk+27).ge.tem) go to 795
      tem=jobmat(jrow,numtsk+27)
      macmat(i,2)=maccol
      go to 795
789   newjob=macmat(i,macmat(i,2))
      curtsk=jobmat(newjob,3)
      jobmat(newjob,curtsk+33)=t+jobmat(newjob,curtsk+21)
790   continue
c ************************************
      go to 999
2222  do 618 i=1,8
618   print 1111, i, maxq(i)
      timexp=t-tbeg
      print 1112, maxj
      print 1113, timexp, msum, jbsm, jobnum
      stop
      end
```

Appendix C

Fortran Code for Uniform Random Number Generator

```
subroutine rndn(ix,iy,yfl)

m1 = 65539

m2 = 4101

m3 = 261

iy = ix*m3

m4 = m1

if (iy.lt.0) m4 = m2

iy = iy*m4

if (iy.lt.0) iy = iy + 2147483647 + 1

yfl = iy

yfl = yfl*.4656613e-9

ix = iy

return
```

Note: ix is integer seed.
yfl is uniform random number ≥ 0 but < 1.

Appendix D

Independent Variables[1] Presented to
Phase 3 Evaluatory Stepwise
Multiple Regression Procedure

1. Mean Lateness
2. Job Stream 2
3. Job Stream 3
4. Job Stream 4
5. Job Stream 5
6. Job Stream 6
7. Job Stream 7
8. Job Stream 8
9. Job Stream 9
10. Job Stream 10
11. Job Stream 11
12. Job Stream 12
13. Job Stream 13
14. Job Stream 14
15. Job Stream 15
16. Job Stream 16
17. Job Stream 17
18. Job Stream 18
19. Job Stream 19
20. Job Stream 20
21. Global
22. High (90% Utilization)
23. Unknown (Actual Processing Times)
24. SLACK
25. CR
26. EOPDD
27. OPSLK
28. OPCR
29. Global*High
30. Global*Unknown
31. Global*SLACK
32. Global*CR
33. Global*EOPDD
34. Global*OPSLK
35. Global*OPCR
36. High*Unknown
37. High*SLACK
38. High*CR
39. High*EOPDD
40. High*OPSLK
41. High*OPCR
42. Unknown*SLACK
43. Unknown*CR
44. Unknown*EOPDD
45. Unknown*OPSLK
46. Unknown*OPCR
47. Global*High*Unknown

[1] All variables except Mean Lateness are dummy variables. Base case is EDD, Local, 75% Utilization, Job Stream 1, and Known Actual Processing Times.

Appendix E

Additional Tables

Table E.1

Penalty Comparisons Between Cycles Under EDD
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	1149	2	3	4	5	6	7	8	9	10
2	894		3	4	5	6	7	8	9	10
3	849			4	5	6	7	8	9	10
4	806					6	7	8	9	10
5	786					6	7	8	9	10
6	750									
7	744									
8	724									
9	695									
10	753									

Cycle	Median MSL									
1	1149	2	3	4	5	6	7	8	9	10
2	904		3	4	5	6	7	8	9	10
3	880			4	5	6	7	8	9	10
4	845					6	7		9	
5	795					6	7		9	
6	753									
7	755									
8	759									
9	703									
10	757									

Cycle	Median MAL									
1	24.5	2	3	4	5	6	7	8	9	10
2	21.5		3	4	5	6	7	8	9	10
3	20.7			4	5	6	7	8	9	10
4	20.3					6	7	8	9	10
5	20.2					6	7	8	9	
6	19.4									
7	19.4									
8	19.2									
9	18.8									
10	19.6								9	

Cycle	Median SQL							
1	542	3			6	7		9
2	487	3	4		6	7		9
3	366							
4	462	3			6	7		9
5	472	3			6	7		9
6	333							
7	410				6			9
8	435				6			9
9	362							
10	425				6	7		9

Table E.2

Penalty Comparisons Between Cycles Under SLACK
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	1006	2	3	4	5	6	7	8	9	10
2	761		3	4	5	6	7	8	9	10
3	715			4	5	6	7	8	9	10
4	684				5	6	7	8	9	10
5	630					6				
6	625									
7	633									
8	617									
9	590									
10	591									

Cycle	Median MSL									
1	1079	2	3	4	5	6	7	8	9	10
2	773		3	4	5	6	7	8	9	10
3	717			4	5	6	7	8	9	
4	706				5	6		8	9	
5	674					6				
6	656									
7	672					6				
8	656									
9	632									
10	675					6		8	9	

Cycle	Median MAL									
1	24.1	2	3	4	5	6	7	8	9	10
2	19.8		3	4	5	6	7	8	9	10
3	19.6			4	5	6		8		
4	19.4									
5	19.2					6				
6	19.0									
7	19.1					6				
8	18.9									
9	19.2									
10	19.7				5	6	7	8	9	

Cycle	Median SQL								
1	352								
2	344								
3	371								
4	401								
5	414								
6	404								
7	437	2	3	4	5	6			9
8	411					6			
9	414								
10	463	2	3	4	5	6	7	8	9

Table E.3

Penalty Comparisons Between Cycles Under CR
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	752	2	3	4	5	6	7	8	9	10
2	555		3	4	5	6	7	8	9	10
3	462			4	5	6	7	8	9	10
4	412				5	6	7	8	9	
5	387						7	8	9	
6	392						7	8	9	
7	385									
8	363									
9	371									
10	374									

Cycle	Median MSL									
1	792	2	3	4	5	6	7	8	9	10
2	603		3	4	5	6	7	8	9	10
3	483			4	5	6	7	8	9	10
4	414				5	6	7	8	9	10
5	387							8	9	
6	394						7		9	
7	398				5					
8	366									
9	379									
10	378									

Cycle	Median MAL									
1	17.9	2	3	4	5	6	7	8	9	10
2	15.9		3	4	5	6	7	8	9	10
3	14.4			4	5	6	7	8	9	10
4	13.8				5	6	7	8	9	10
5	13.4						7		9	
6	13.5						7		9	
7	13.3									
8	13.4									
9	13.2									
10	13.4						7		9	

Cycle	Median SQL					
1	305	2		4	5	6	7
2	212						
3	207				5		
4	221						
5	200						
6	217						
7	212		3				
8	223				5	6	7
9	236				5	6	7
10	242			4	5	6	7

Table E.4

Penalty Comparisons Between Cycles Under EOPDD
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	1001	2	3	4	5	6	7	8	9	10
2	750		3	4	5	6	7	8	9	10
3	616			4	5	6	7	8	9	10
4	556				5	6	7	8	9	10
5	509						7	8	9	10
6	484									
7	489								9	
8	497					6			9	10
9	486					6				
10	497					6	7			

Cycle	Median MSL									
1	1158	2	3	4	5	6	7	8	9	10
2	851		3	4	5	6	7	8	9	10
3	648			4	5	6	7	8	9	10
4	586				5	6	7	8	9	10
5	521						7	8	9	10
6	512						7		9	10
7	504								9	
8	513					6			9	10
9	490									
10	507						7			

Cycle	Median MAL									
1	24.5	2	3	4	5	6	7	8	9	10
2	20.5		3	4	5	6	7	8	9	10
3	17.6			4	5	6	7	8	9	10
4	16.8				5	6	7	8	9	10
5	16.2						7	8	9	10
6	16.1						7		9	10
7	16.0								9	
8	16.2					6			9	
9	15.7									
10	15.9									

Cycle	Median SQL									
1	308	2	3	4	5	6	7	8	9	10
2	243			4	5	6	7	8	9	10
3	227			4	5	6	7	8	9	10
4	203									
5	215					6	7	8		
6	191									
7	197									
8	198					6	7			
9	215							8		
10	213							8		

Table E.5

Penalty Comparisons Between Cycles Under OPSLK
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	1016	2	3	4	5	6	7	8	9	10
2	732		3	4	5	6	7	8	9	10
3	576			4	5	6	7	8	9	10
4	522					6			9	10
5	544			4		6	7		9	
6	502									
7	523			4					9	10
8	524			4					9	10
9	517									10
10	514									

Cycle	Median MSL									
1	1138	2	3	4	5	6	7	8	9	10
2	858		3	4	5	6	7	8	9	10
3	596			4	5	6	7	8	9	10
4	537					6	7	8	9	10
5	545			4		6	7		9	
6	507									
7	525									10
8	533								9	10
9	528					6	7			10
10	514					6				

Cycle	Median MAL									
1	24.7	2	3	4	5	6	7	8	9	10
2	21.0		3	4	5	6	7	8	9	10
3	17.5			4	5	6	7	8	9	10
4	16.6				5	6	7	8	9	10
5	16.1									
6	16.2				5					
7	16.3									
8	16.3									
9	16.4				5		7	8		
10	16.4					6	7	8		

Cycle	Median SQL									
1	396	2	3	4	5	6	7	8	9	10
2	236									
3	277			4			7	8		10
4	236									
5	253							8		
6	257							8		
7	269				5	6		8		10
8	216									
9	216									
10	260	2			5	6		8	9	

Table E.6

Penalty Comparisons Between Cycles Under OPCR
Global Allowance Policy Forms
(90% Utilization, Known Actual Processing Times)

Cycle	Median VAR									
1	897	2	3	4	5	6	7	8	9	10
2	727		3	4	5	6	7	8	9	10
3	599			4	5	6	7	8	9	10
4	549				5	6	7	8	9	10
5	510					6	7	8	9	10
6	480								9	10
7	471									
8	488					6				
9	463									
10	466									

Cycle	Median MSL									
1	952	2	3	4	5	6	7	8	9	10
2	741		3	4	5	6	7	8	9	10
3	612			4	5	6	7	8	9	10
4	554				5	6	7	8	9	10
5	514					6	7	8	9	10
6	486						7		9	10
7	474									
8	489					6				
9	467									
10	467									

Cycle	Median MAL									
1	19.8	2	3	4	5	6	7	8	9	10
2	17.1		3	4	5	6	7	8	9	10
3	16.2			4	5	6	7	8	9	10
4	15.6				5	6	7	8	9	10
5	14.9					6	7	8	9	10
6	14.7							8	9	10
7	14.8							8	9	
8	14.6									
9	14.4									
10	14.5									

Cycle	Median SQL									
1	314	2	3	4	5	6	7	8	9	10
2	291		3	4	5	6	7	8	9	10
3	284			4	5	6	7	8	9	10
4	249					6	7	8	9	10
5	235					6	7	8	9	10
6	224								9	10
7	227					6				
8	234								9	10
9	209									
10	219									

Vita

[redacted]