This is to certify that the

dissertation entitled

Multi-Objective Sequencing On A Single Processor With Sequence Dependent Setup Times: A Simulated Annealing Approach

presented by

Keah Choon Tan

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Business Administration

Major professor

Date

MULTI-OBJECTIVE SEQUENCING ON A SINGLE PROCESSOR WITH SEQUENCE DEPENDENT SETUP TIMES: A SIMULATED ANNEALING APPROACH

By

Keah Choon Tan

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Management

1995

ABSTRACT

MULTI-OBJECTIVE SEQUENCING ON A SINGLE PROCESSOR
WITH SEQUENCE DEPENDENT SETUP TIMES:
A SIMULATED ANNEALING APPROACH

BY

KEAH CHOON TAN

This research applies simulated annealing to minimize a weighted sum of tardiness and setup times on a single processor with sequence dependent setup times. The multiple objectives are combined by means of a tradeoff parameter. Schedulers can sequence jobs to minimize a weighted sum of multiple objectives by varying this parameter. New objectives can be added easily. The same formulation can be used to solve single objective minimization problems by simply setting the tradeoff parameters of the unwanted objectives to zero, thus eliminating them. Although it is not known if the proposed simulated annealing scheme identified any optimum solutions for the larger problems, it located the optimum solutions for 40% or 144 of the 360 10-job problems. The algorithm can retain the best incumbent solution at all time. Therefore, the best known solution is always readily available should there be a need to terminate the annealing process prematurely or unexpectedly. This research also exploits the Boltzmann distribution to provide some analytical guidelines on setting the proper annealing parameters.

The results of this dissertation suggest that simulated annealing holds promise as a viable solution technique for the type of scheduling problems considered, and the quality

of the solutions developed might be acceptably good for both the single and multiple objective formulations. Specifically, simulated annealing can handle the complex combinatorial nature of these problems which are difficult to solve by other methods. By making it possible to jointly consider multiple objectives, simulated annealing offers a promising alternative to solve scheduling problems where setup times are sequence dependent. The results of this research also suggest ways in which more complex (realistic) job shop environments can be modeled.

Research results indicate that: (1) the types of starting solution and the relative range of due date have no impact on the simulated annealing scheme, (2) 50 and 100 updates produced better solution than a single update between changes in temperature range, (3) setup problems were more difficult to solve than problems involving tardiness, and (4) the impact of selecting a 'bad' job with high setup was minimized with large problems.

DEDICATION

Dedicated in Memory of My Beloved Sister,

CHIEN, YOKE-CHEONG

(June 15, 1957 - August 27, 1980)

ACKNOWLEDGMENTS

I am deeply grateful to my dissertation committee, Professor Ram Narasimhan (Chair), Professor Robert B. Handfield, Professor Gary L. Ragatz, and Professor Michael A. Shanblatt for their support and invaluable contributions to this dissertation. They have made helpful suggestions and contributed greatly both with their inputs and their support. Their inputs and standards improved the final product immeasurably. I want to extend my very special thanks to these four very special individuals. I am very fortunate to have these four outstanding professors to serve on the dissertation committee.

Professor Narasimhan has provided invaluable suggestions on the experimental design, comments and guidance on the completion of this dissertation. He has performed his duties beyond that of a chair for a dissertation committee. Professor Handfield has patiently read and checked this dissertation for both form and content. His short turn around time played an important role in the timely completion of this dissertation. Professor Ragatz's knowledge and insights on scheduling problems played an important role in laying out the research problems. His comments on the research findings improved the quality of this dissertation. A special word of thanks to Professor Shanblatt, who has agreed to serve on the dissertation committe of a total stranger. His expertise and comments on simulated annealing were a major contributing factor in solving the research problem successfully.

I owe a special debt of gratitude to my first academic advisor, Professor Soumen Ghosh, who helped to plan my course work, and taught me to write my first conference proceeding. His thoughtful design of my course work has prepared me very well for the statistical analyses needed for this dissertation. I also want to thank Professor Paul A. Rubin for pointing out an error on my initial solution technique.

Many individuals also provided support and encouragement by calling and checking regularly to ensure that this dissertation was progressing on track. A special note of thanks to Dr. Vijay R. Kannan, Dr. Steve B. Lyman, and Jack Williams. I also owe a special word of thanks to Sue Polhamus who arranged Dr. Joel Litchfield to be my office mate, who in turn allowed me to work peacefully and quietly during all those days and nights in the office. Lonnie Herman and Kathy Mullins deserve a special word of thanks for their gracious support, encouragement and for providing invaluable secretarial support throughout my graduate studies at this fine university. Dr. Michael K. Moch and Dr. John H. Hoagland also deserve a special word of thanks for their continuing support and employment opportunities, without which I would not have the financial means to complete this study.

A very special word of thanks to my parents, Keah Soon Chien and Yoke Mui Sui, for investing a major part of their financial resources, time and hopes on me. Finally, I need to express my gratitude to my wife, Shaw Yun Chiu, and my daughter, Wen Hui Tan, for their encouragement, understanding and forbearance during all those days and nights spent working on my Ph.D. degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

θ	Tradeoff parameter
avgfin	Average objective function value
d	Due date
INIT	Type of initial solution
lnavgfin	Natural logorithm of average objective function value
n	Number of jobs in the sequence
p	Processing time
pctabopt	Percent above the optimum solution
PWS	Pair-wise swapped initial solution
RD	Relative range of due date
RND	Randomly generated initial solution
s	Setup time
STEP	Number of updates between changes in temperature range
TF	Tardiness factor

CHAPTER 1: INTRODUCTION

1.1 Introduction

A significant number of operations scheduling studies have been published over the last three decades. This research can be classified as either theoretical research dealing with optimizing procedures limited to static problems, or experimental research dealing with dispatching rules in both static and dynamic cases. In the static case, all relevant information about the jobs that are to be processed in the shop are known at the start of the scheduling horizon. In the dynamic case, jobs arrive intermittently and relevant information of all the jobs are not known at the start of the scheduling horizon. As a result of over thirty years of research, the literature dealing with operations scheduling is both large and diverse, as obvious from such extensive survey papers as Day and Hottenstein (1970), Panwalkar and Iskander (1977), Graves (1981), Blackstone, *et al.* (1982), and Hax and Candea (1984). The combinatorial nature of the scheduling problems and the potential benefits that exist in better scheduling continue to challenge operations researchers.

Though an enormous amount of research exists on operations scheduling problems, many either totally ignore setup times or assume that setup times on machines are independent of the job sequence (Corwin and Esogbue 1974, Ragatz 1993). Examples of sequence independent sequencing research include Potts and Van Wassenhove (1985, 1987, 1991), and Van Laarhoven, *et al.* (1992). In a survey of one hundred and fifteen industrial schedulers, Panwalkar, *et al.* (1973) reported that seventy percent of the

schedulers stated that sequence dependence of setup times occurred with at least twenty-five percent of the jobs scheduled. The magnitude of setup times depends on the similarity of the process technology requirements of two consecutive operations. Typically, large setup times are expected for two consecutive operations with significant differences in process technology requirements (White and Wilson 1977, Srikar and Ghosh 1986). In addition, most research has tended to focus on a single objective (for example, minimizing makespan, tardiness or setup times).

Panwalkar, *et al.* (1973) also reported that the single most important scheduling criterion was meeting due dates. A frequently cited due date related performance measure was minimizing the total tardiness of the jobs, $T = \sum_{i=1}^{n}$ [max(0, completion time$_i$ - due date$_i$)], or mean tardiness which differed only by the constant factor $1/n$ where n was the number of jobs. Minimizing total processing time, minimizing total setup times or costs, and minimizing in-process inventory costs were considered by industrial schedulers as secondary objectives. Despite the practical importance of operations scheduling to meeting due dates in the presence of sequence dependent setup times, research in this area tended to focus on minimizing total setup times. This priority is equivalent to minimizing the makespan of a set of jobs since the sum of the processing times is a constant. Total tardiness is a difficult criterion to work with, since it is a non-linear function. Furthermore, completing a job before the due date does not reduce total tardiness, but simply results in zero tardiness for that job. No simple rule is known to minimize tardiness, except in two special cases when the setup times are sequence independent: (1) Shortest Processing Time schedules minimize total tardiness if all jobs are tardy, and (2)

Earliest Due Date schedule minimizes total tardiness if at most one job in the sequence is tardy (Emmons 1969).

The simplest shop configuration in operations scheduling is the single processor (also called the single machine shop). Each job has a single operation that is to be performed on the single machine in the shop. Scheduling a single processor is equivalent to sequencing, in that it includes finding a sequence in which jobs have to be processed on a machine. Theoretically, the simpler single processor is the first step in investigating and understanding scheduling principles in more complex job shops. Practically speaking, many shops are limited by a bottleneck machine (Graves 1981, Hax and Candea 1984), and scheduling is often done by considering only a bottleneck machine. For this reason, dispatching rules have been studied extensively in the single processor environment. However, total tardiness remains a performance measure that cannot be optimized by any dispatching rule, even for this simple environment. Complete enumeration is infeasible for problems with a large number of jobs since there are $n!$ possible solutions for n jobs. If one wishes to minimize total tardiness regardless of sequence dependent setup times, the only known methods are either dynamic programming or branch and bound procedures (Blackstone, *et al.* 1982).

Dynamic programming (DP) is an approach for finding an optimal solution to a problem by breaking it into smaller subproblems, each labeled a stage. Although DP has tremendous potential due to its ability to solve difficult problems in which other optimization tools fail, it suffers from an exponential growth in the amount of computation as a function of problem size. If the problem doubles in size, the amount of computation

quadruples. In addition, every time a problem differs slightly, a new formulation must be designed. For problems with many jobs and constraints, DP is rendered inefficient or even infeasible.

Branch and bound (BB) procedures are an intelligent search procedure resulting in either an optimal or a close to optimal solution to mathematical programming problems. BB procedures include pure Integer Programming (IP) and Mixed Integer Programming (MIP) problems. The procedure divides a problem into two or more subproblems (branching) and sets two bounds on the value of the objective function. All subproblems whose objective functions are better than the established feasible bounds are used to modify the bound. These are then subdivided and investigated. The process is repeated until no further subdivision is possible, at which point the optimal or near optimal solution has been reached. BB can be efficiently coded into computer routine and works well in problems containing a few integer variables. However, the tightness of the bounding procedures remains a critical factor in determining how well BB works.

1.2 Problem Statement

The existing research in operations scheduling appears to provide little assistance to industrial schedulers. Sequencing jobs on a single processor with sequence dependent setup times to minimize makespan is analogous to the Traveling Salesmen Problem (TSP) which is an np-complete problem. The time it takes to solve an np-complete problem tends to increase exponentially with the size (n) of the problem (Hax and Candea 1984).

Minimizing total tardiness is a difficult and challenging task since tardiness is not linear in completion times, thus requiring combinatorial techniques for solution. In addition, schedulers do not always focus on a single objective. In many instances, schedulers may want to optimize the primary objective while keeping some secondary objectives at reasonable levels. For example, a scheduler may want to optimize tardiness while keeping total setup times at a reasonable level.

There are very few algorithms or solution approaches that can handle either due dates or setup times, and virtually no known efficient algorithm exists to handle both setup times and tardiness sequentially. This study investigates the multi-objective sequencing problem on a single processor with sequence dependent setup times to minimize a weighted sum of tardiness and setup times. The research focuses on investigating whether simulated annealing provides an efficient heuristic approach that can meet the needs of industrial schedulers.

The research question is: can simulated annealing provide an efficient solution approach to sequence jobs on a single processor with sequence dependent setup times to minimize a weighted sum of setup times and tardiness?

1.3 Organization of Dissertation

In chapter 2, the literature on operations scheduling and research addressing single processor or/and sequence dependent setup times will be discussed. In addition, the concepts and related research of simulated annealing will be described in detail (Simulation studies for the dynamic job shop problem will not be covered in depth since it is beyond the scope of this research). In chapter 3, the methodology, techniques, and the experiments used in this research will be discussed in detail. Chapter 4 provides a brief discussion on the data generated, including statistical tests to ensure that the data conforms to the intended statistical distributions. Chapter 5 discusses the results of this research, and chapter 6 summarizes the conclusions of this research.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

Operations scheduling is the final process associated with hierarchical production planning and inventory control systems. Its purpose is to make the most detailed scheduling decisions involving the assigning of operations to specific machines or/and operators during a given time interval. It is probably one of the most well-studied areas of production and inventory control systems. A number of basic definitions provided by Conway, *et al.* (1967) will be used throughout this research.

(i) *Operation* is an elementary task to be performed.
(ii) *Processing Time* is the amount of processing required by an operation.
(iii) *Setup Time* is the time for changing over a machine to process a different job.
(iv) *Job* is a set of operations that are interrelated by precedence restrictions derived from technological constraints.
(v) *Machine* is a facility that is capable of performing an operation.
(vi) *Due date* is the time in which the last operation of the job should be completed. In the case of single machine, it is the time in which the operation on the machine should be completed.
(vii) *Scheduling* is assigning each operation of each job a start time and a completion time onto machines.
(viii) *Sequencing* is establishing the order in which jobs waiting in the queue in front of a particular machine to be processed. Scheduling a single processor is equivalent to sequencing.
(xv) *Dispatching* indicates the single job to be performed first.

This chapter will briefly discuss the survey articles in operations scheduling. Scheduling research that addresses single processor or/and sequence dependent setup times will be discussed in greater detail. In addition, simulated annealing research will also be discussed.

2.2 Survey Literature In Operations Scheduling

Day and Hottenstein (1970) provided a schema for classifying machine-constrained operations scheduling research into three broad categories. The three categories were:

(1) Jobs arrival process.
- Static scheduling where n jobs arrive at time zero.
- Dynamic scheduling where an infinite number of jobs arrive continuously and according to some probability density function.

(2) The number of machines, m, involved.
- Single stage or single processor production system where $m = 1$.
- Multistage or multiprocessor production system where $m > 1$.

(3) The nature of the job route for multistage production system.
- parallel routing.
- flow shop.
- hybrid shop (static case) or job shop (dynamic case).

Day and Hottenstein noted that research in the single processor static sequencing problem has been extensive. Solution approaches include (1) combinatorial techniques, (2) mathematical programming, (3) heuristics, and (4) Monte Carlo sampling. Although optimal sequence techniques have been found to minimize some performance measures, there was a lack of research that addressed the issue of minimum tardiness in the presence of sequence dependent setup times.

Panwalkar and Iskander (1977) classified operations scheduling research into (1) theoretical research dealing with optimizing procedures limited to the static problems, and (2) experimental research dealing with dispatching rules in both static and dynamic cases.

The authors have focused on experimental research and classified over one hundred dispatching rules. However, they were unable to properly classify the results of the various researchers because of conflicting results reported in many cases, probably due to differences in experimental conditions. They further concluded that most operations scheduling research was based on hypothetical problems, and that there was a need for more research based on real problems.

Graves (1981) reviewed the theoretical developments for various scheduling problems and contrasted the available theory with the practice of production scheduling. Graves reported that a wide variety of results existed for the one stage, one processor problem, due to different problem specifications. However, the problem of minimizing weighted tardiness has not been satisfactorily solved. Although there seems to be a mismatch between scheduling theory and practice, there were encouraging signs for certain production settings (in particular, the single stage production shop) in which results of operations scheduling research have been either tested or adopted by practitioners.

Blackstone, *et al.* (1982) reviewed recent studies on dispatching rules but failed to identify a single rule as the best in all production settings. Sequencing procedures were not discussed in the study. The authors reported that mean tardiness remained a performance measure that could not be optimized by any dispatching rule, even for the single processor problem. Branch and bound procedures, and dynamic programming were the only alternatives for minimizing mean tardiness that guarantee optimality.

2.3 Scheduling Research With Sequence Dependent Setup Times

Lockett and Muhlemann (1972) emphasized that assumption of sequence independent setup times did not hold in a significant number of actual situations. They derived a branch and bound algorithm for scheduling jobs on a machine with sequence dependent setup times. The objective was to minimize the total number of tool changes. Tool changes depended not only on the previous job, but on prior jobs as well. However, this algorithm was computationally restrictive, except for the smallest problem. In addition, five scheduling heuristics comparable to those used to solve traveling salesman problems (TSP) were tested. The authors noted that the computational results of these heuristics were encouraging although they did not guarantee an optimum schedule. Processing times (which could be added or considered as part of the setup times) were completely ignored in this study.

Panwalkar, *et al.* (1973) mailed questionnaires and conducted personal plant visits in an attempt to survey actual industrial scheduling problems. Major discrepancies between scheduling researchers and practitioners were reported. Scheduling research tended to focus on minimizing total processing times, makespan, setup times, job lateness, and work in process. Such performance measures were considered as secondary goals by practitioners, who considered the meeting of due dates to be the single most important criterion. In addition, this study concluded that seventy percent of the practitioners reported that at least twenty five percent of their operations were subject to sequence dependent setup times. Unfortunately, there were few algorithms that could handle due dates or/and set up times efficiently. This clearly indicated a need for further research in

narrowing the gap between scheduling practitioners and researchers. In particular, there was a need to develop better solution approaches for meeting due dates (tardiness) in the presence of sequence dependent setup times.

Corwin and Esogbue (1974) presented two dynamic programming formulations for the two machine flow shop scheduling problems, with sequence dependent setup times on one of the machines. The objective was to determine a schedule that minimized makespan subject to meeting certain due date constraints. The authors indicated that the computational complexity of these two formulations was equivalent to the classical Traveling Salesman Problem, and the largest problem solvable on a computer was approximately fourteen to fifteen jobs.

Prabhakar (1974) formulated a two stage chemical production scheduling problem with sequencing considerations as a mixed integer programming (MIP) problem. The problem was solved using a branch and bound algorithm by first solving the MIP as a linear programming (LP) problem. The continuous variables established the upper/lower bound, and each non-integral integer variable originated from a node, from which two branches were created. The MIP model was able to efficiently solve a problem of moderate size. However, as the problem size grew larger, the model failed to handle the sequence dependent part of the problem. The author proposed to decompose the sequencing part as a subproblem and solve it separately.

White and Wilson (1977) used forward stepwise multiple regression to predict the setup times required in changing over machine tools. Independent variables considered include eleven dummy and three continuous variables. Five of the dummy and two of the

continuous variables were shown to be significant factors (α = 15%) in predicting setup times. The same regression equation was then used as a heuristic technique for sequencing jobs onto machines. Basically, the heuristic prioritized jobs with the highest regression coefficient to be processed as a group, thus preventing setup. The model ignored interaction effects among the independent variables.

Schrage and Baker (1978) presented a dynamic programming algorithm for solving one machine sequencing problems with precedence constraints. This algorithm was shown to be superior in performance compared to the "Chain" algorithm of Baker and Schrage (1978), and the branch and bound algorithms of Fisher (1976), Picard and Queyranne (1976), and Rinnooy Kan, *et al.* (1975). However, these algorithms were all coded in different computer languages and executed on different types of computers, thus making fair comparison of the performance of these different algorithms difficult.

Gupta (1982) proposed a branch and bound algorithm to minimize setup cost in an *n* jobs and *m* machines flow shop with sequence dependent setup times. The performance of this algorithm was not reported since no other similar technique could solve such a generalized problem proposed by the author. However, the author noted that this algorithm was suitable for solving small size problems only. Heuristic rules remained as the preferred techniques for solving large scheduling problems where computational efforts increased rapidly with problem size.

Srikar and Ghosh (1986) formulated a mixed integer linear program (MILP) for solving an *n* jobs and *m* machines flowshop with sequence dependent setup times. Their formulation required considerably reduced number of integer binary variables, than other

formulations. The formulation used binary variables (X_{ij} = 0 or 1) to represent the precedence relationship of job *i* and job *j*. The objective of their formulation was to minimize makespan, which was linear in job completion times. However, like any other MILP problems, the computation times of their formulation became prohibitive beyond the 6 jobs x 6 machines on a Prime 550 minicomputer.

2.4 Single Machine Scheduling Research

Gavett (1965) proposed three heuristic rules for sequencing jobs to a single processor with sequence dependent setup times. The goal was to minimize setup times, although the proposed heuristics did not guarantee optimum solution. Generally, with uniformly distributed setup times, the performance of the heuristic rules can be ranked as (1) Next Best Rule After Column Deductions, (2) Next Best Rule With Variable Origin, and (3) Next Best Rule. On average, the performance of these rules ranged from approximately twenty five percent to one hundred and fifty four percent when compared to the optimum solutions. The distribution and variance of the setup times, and the number of jobs in the batch has a significant effect on the performance of the heuristics. Nevertheless, these heuristics provided a simple and practical approach for sequencing jobs on a single processor that consistently outperformed random sequencing.

Emmons (1969) proposed an algorithm for sequencing jobs on a single machine to minimize total tardiness. Setup times of jobs were assumed to be sequence independent in this research. The author noted that there was no simple rule that would minimize total or

mean tardiness, except in two special cases: (1) the shortest processing time dispatching rule would minimize total or mean tardiness if all jobs in the sequence were tardy (in this case, tardiness became a linear function of completion times), and (2) the earliest due date schedule would minimize total or mean tardiness if at most one job was tardy. Basically, Emmons' proposed algorithm removed jobs with large processing times and long due dates to reduce the number of jobs that need to be sequenced.

Haynes, *et al.* (1973) examined the effectiveness of the three heuristic rules proposed by Gavett (1965) where the objective was to minimize setup times. These rules were evaluated under different job sizes (ranging from five to eleven jobs) and setup time distributions (normal, uniform and gamma). Three factor, fixed effects analysis of variance (similar to multiple regression analysis that considered the main effect and higher order interaction terms) was used to determine the effects of the factors. Job sizes, heuristic rules, and the interaction term of setup time distribution with job sizes were shown to be statistically significant factors (at $\alpha = 0.05$) affecting the effectiveness of the models. All three heuristics tended to be least effective when the setup times were uniformly distributed.

Rinnooy Kan, *et al.* (1975) presented a branch and bound algorithm for sequencing jobs on a single processor. The goal was to minimize weighted tardiness of the jobs. The authors claimed that their algorithm was far superior to existing algorithms, particularly for up to fifteen or twenty jobs. However, when the tardiness factors were increased to 0.6 and 0.8 (6 problems each), their algorithm managed to solve only seven of the twelve 20-job problems within the allowable computation time of five minutes on a Control Data

Cyber 73-28 computer. They cited that sharper lower bounds were needed to reduce the size of the search tree. As a result, they concluded that minimizing weighted tardiness remained a difficult problem, and the issue would remain a challenge to researchers for a long time. The author emphasized that in order to solve the minimizing tardiness problem satisfactorily, a completely new approach other than the traditional algorithms or heuristics, or methodology from other disciplines should be investigated.

Driscoll and Emmons (1977) developed an efficient backward-time search procedure for scheduling jobs on one machine that minimized the total setup costs while meeting the due dates of the jobs. They claimed the monotonicity property of the forward-time dynamic program could be applied for developing an efficient backward-time search procedure for solving the problem.

Barnes and Vanston (1981) discussed the problem of scheduling jobs with linear delay penalties and sequence dependent setup costs. They also discussed and compared the application of various branch and bound algorithms for solving this class of scheduling problems. A hybrid algorithm analogous to Morin and Marsten's dynamic programming/ branch and bound approach was shown to be superior in yielding optimal solution.

Potts and Van Wassenhove (1985) presented a branch and bound algorithm for the single machine total weighted tardiness problem where setups were assumed to be sequence independent. The algorithm was tested on problems with 20, 30, 40, and 50 jobs. Processing times (p) were generated from uniform distribution between 1 and 100 ($p \sim U(1, 100)$). Relative range of due dates (RD) and average tardiness factor (TF) were chosen at 0.2, 0.4, 0.6, 0.8, and 1.0. Each job was given a due date generated from a

uniform distribution between (p[1-TF-RD/2]) and (p[1-TF+RD/2]). The computation generated due dates between the ranges -0.5*p (TF = 1 and RD = 1) to 1.3*p (TF = 0.2 and RD = 1). Although no explanation was given for the negative due date, it probably meant overdue jobs. The proposed algorithm was reported to have successfully solved problems with up to 40 jobs. The authors also reported that the problems were most difficult to solve when the tardiness factors were between 0.6 and 0.8.

Potts and Van Wassenhove (1987) discussed and compared three general precedence constrained dynamic programming algorithms and seven special purpose decomposition algorithms for minimizing total tardiness on a single machine where setups were assumed to be sequence independent. The authors concluded that general precedence constrained dynamic programming algorithms relied heavily upon Emmons' (1969) dominance rules. Experimental results indicated that the special purpose decomposition algorithms were far superior to the general precedence constrained DP algorithms. The most effective decomposition algorithm solved all the 100-job test problems in the authors' experiment.

Potts and Van Wassenhove (1991) presented a collection of heuristics for the single machine total tardiness problem. The authors tested all these heuristics against the simulated annealing method on a large set of test problems. Experimental results of 20-job, 40-job, and 50-job test problems indicated that the decomposition heuristic outperformed simulated annealing method in minimizing total tardiness. However, simulated annealing method was clearly a viable approach for minimizing total weighted tardiness. All these heuristics assumed setups were sequence independent. The authors

concluded that the initial solution was a significant factor in determining the quality of the final solution if the simulated annealing process was truncated after a relatively small number of iterations. In a particular weighted tardiness problem, the difference in best solution of two simulated annealing runs on the same problem was reported to be as high as 100% after 10,000 iterations.

Ragatz (1993) proposed a branch and bound approach for minimum tardiness sequencing on a single processor with sequence dependent setup times. Experimental results of eight to sixteen jobs indicated that the algorithm was fairly effective in solving the smaller problems. However, the algorithm has difficulty with larger problems, particularly those with a narrow range of due dates or/and a high variance in processing times. In more than half the research problems, the author managed to find the optimal sequence within the first ten thousand nodes explored.

Rubin and Ragatz (1995) applied a genetic search algorithm (GS) to a subset of test problems similar to Ragatz (1993). The performance of GS on a set of thirty two test problems was compared to (1) a pure random search (RS) and (2) Ragatz's branch-and-bound algorithm (RBB). Moderate tardiness factor and narrow due date range were considered as the hard levels, whereas low tardiness factor and wide due date range were considered as the easy levels. The authors concluded that:

(1) processing time variance was not a determinant of comparative performance,

(2) GS and RS produced "good" solutions faster than RBB with test problems having moderate tardiness factor and wide due date range,

(3) RBB was superior to GS and RS when the tardiness factor was low and due date range was narrow,

(4) GS, RS and RBB located an optimal schedule almost immediately when the tardiness factor was low and due date range was wide. The authors classified that category as a trivial problem and stated that sequencing jobs according to their due dates appeared to work in general.

(5) Neither GS, RS nor RBB appeared to be consistently superior when the tardiness factor was moderate and due date was narrow.

2.5 Summary of Current Scheduling Research

This review of current scheduling research suggests that despite its practical importance, sequencing jobs with sequence dependent setup times to minimize tardiness has not been given due consideration. Most researchers have focused on a single objective, and have either attempted to minimize total setup costs or makespan, or assumed sequence independent setup times. Setup times were either totally ignored or treated as sequence independent and thus absorbed as part of the processing times. Dynamic programming and branch and bound algorithms were the general solution approaches applied so far. Unfortunately, these solution approaches have proved to be neither efficient nor feasible for problems with large number of jobs.

2.6 Simulated Annealing

Simulated annealing offers a powerful search heuristic for obtaining excellent solutions for problems in the np-complete category. Unlike local optimization or iterative improvements which repeatedly improve the initial solution by making small local

alterations until no such alteration yields a better solution, simulated annealing randomizes this procedure in a way that allows for occasional changes that worsen the solution, in an attempt to reduce the probability of becoming stuck in a locally optimal solution. The only requirement for solving a combinatorial optimization problem with simulated annealing is that there be a set of moves that can generate a new solution from the current solution. Therefore, it is possible to represent the combinatorial optimization problem of interest in this research as a simulated annealing problem.

The basic steps in casting a combinatorial problem into an annealing problem are: (1) describe the state space, (2) the move set, and (3) the objective function. In this research, the state space is the set of feasible solutions, that is, all possible sequences. The move set involves randomly switching the sequence of two jobs. The objective function consists of a weighted sum of tardiness and setup times. In this research, the proposed simulated annealing schedule will accept with some probability changes in the sequencing of two jobs, despite poorer performance in an attempt to allow the problem to avoid a local minimum in favor of a better solution. The procedure uses the Boltzmann distribution to determine whether to accept a change that leads to a worse sequence.

Simulated annealing makes a random switch in the sequence of two jobs (called state change) and determines the resultant objective value created after the change. The procedure slowly decreases the probability of accepting the random switch in the sequence that results in poorer performance. Rejection or acceptance of the state change is made according to the following criteria:

(1) If the objective value is lower after the random state change (i.e. system performed better), accept the change.

(2) If the objective value is not lower after the random state change (i.e. system performed worse), accept or reject the change according to the Boltzmann Distribution.

2.7 Simulated Annealing Research

Kirkpatrick, *et al.* (1983) discussed some successful applications of the simulated annealing techniques to a number of practical problems, including optimization of the traveling salesman problem. The technique proposed involved finding a function to be minimized, choosing a starting temperature, and then dropping the temperature according to a useful annealing schedule. The authors exploited the annealing analogy of solids to provide a framework for combinatorial optimization problems. They claimed to have annealed into local optimum the traveling salesman problem for up to 6,000 cities.

Johnson, *et al.* (1989) reported on an extended empirical study of the use of simulated annealing to the combinatorial optimization problem proposed by Kirkpatrick, *et al.* (1983). The authors investigated how to adapt simulated annealing to particular problems and compared its performance to traditional algorithms. They applied simulated annealing to a graph partitioning problem and reported that simulated annealing clearly out-performed traditional local optimization and the algorithm due to Kernighan-Lin (1970). They further concluded that longer annealing runs produced better results, and starting the annealing schedule at a good initial solution produced better solutions than starting at a randomly generated initial solution. However, Suresh and Sahu (1993) reported a contradictory finding which stated that a very important feature of simulated annealing was the nondependence of the final solution on the initial solution.

Brown, *et al.* (1992) compared the performance of simulated annealing and genetic algorithm to routing and scheduling traffic over a freight rail network. The authors concluded that simulated annealing was computationally more efficient and superior to genetic algorithm for their application. Genetic algorithms required several hours to converge on a set of good answers, whereas simulated annealing could often find a fairly good answer in a few minutes.

Van Laarhoven, *et al.* (1992) used simulated annealing to find the minimum makespan in a job shop where setups were assumed to be sequence independent. The algorithms were tested on problems varying from six jobs on six machines to thirty jobs on ten machines. Experimental results indicated that the proposed simulated annealing heuristic was able to find shorter makespans than the iterative improvement approach and the shifting bottleneck procedure, at the expense of large computation times. However, the authors considered the disadvantage of large computation times to be compensated for by: (1) the simplicity of the algorithm, (2) its ease of implementation, (3) the fact that it required no deep insight into the combinatorial structure of the problem, and (4) the high 'quality' of the final solution.

Laursen (1993) investigated the optimal tradeoff between simulation time and solution quality of simulated annealing algorithms applied to the quadratic assignment problem. The author confirmed Johnson's, *et al.* (1989) findings and reported that the simulation length was indeed optimizable, although a rather broad range of simulation lengths produced near optimal solution quality. The author found that the optimal simulation time (about 25,000 iterations) was rather short for the quadratic assignment

problem. Therefore, several short simulation runs of the same problem should be the preferable choice rather than using a single long simulation run.

Shang (1993) used the simulated annealing heuristic to find a near optimal solution for the proposed multicriteria facility layout problem. The multiple objective quadratic assignment problem combined both qualitative and quantitative objectives. Analytic hierarchy process was used to find the relative weight for each qualitative aspect of the facility layout problem. The author tested the simulated annealing heuristic on three different initial solutions and reported that the initial solution had a significant effect on the quality of the final solution.

2.8 Summary of Simulated Annealing Research

Simulated annealing was first applied to combinatorial problems by Kirkpatrick, *et al.* (1983). Many researchers have subsequently reported successful applications of simulated annealing in obtaining good solutions to NP-hard combinatorial problems. Simulated annealing has been successfully used to find good solutions to minimizing makespan and tardiness problems in job shop scheduling (where setups were assumed to be sequence independent). It has also been successfully applied to solve multi-objective facility layout problem. A general conclusion that can be reached is that a well tuned simulated annealing search heuristic could be a viable and attractive alternative job shop scheduling approach when the problem is too complex to be solved through optimization methods.

CHAPTER 3: RESEARCH DESIGN

3.1 Introduction

This research investigated the problem of scheduling a single processor to minimize a weighted sum of setup times and tardiness when setup times are sequence dependent. A tradeoff parameter, θ, was used to combine the multi-objective (setup times and tardiness) into a single objective function. An initial solution was generated, and then simulated annealing was used to 'lower' the combined objective function into a local or global minimum. All jobs were sequence dependent and generated according to pre-defined distributions specified in section 3.6.

3.2 Research Statement

This study applied simulated annealing to solve the multi-objective sequencing problem on a single processor with sequence dependent setup times. The objective was to minimize a weighted sum of tardiness and setup times. The processing time (p_i), setup time (s_{ij}), and the due date (d_i), for each job were assumed to be known with certainty when jobs arrived at the shop, and setup times were sequence dependent. In this environment, minimizing makespan is equivalent to minimizing setups since the total processing times of all the jobs remained the same regardless of the processing sequence.

Simulated annealing was used to locate the local or global optimal sequence that minimized a weighted sum of setup times and tardiness. This approach allows minimizing tardiness and setup times sequentially or considering them together using a tradeoff parameter. In addition, this research investigated the effect of varying the parameters affecting the annealing schedule.

The specific issues investigated by this research are as follows:

(a) Most of the existing operations scheduling research has either totally ignored setup times or assumed that setup times on each machine are independent of the job sequence despite the fact that at least twenty-five percent of the jobs scheduled by practitioners are sequence dependent. In addition, past research tends to focus on a single objective. This research focuses on sequencing jobs on a single processor with sequence dependent setup times to minimize a weighted sum of tardiness and setup times. Simulated annealing is used to solve the research problem.

(b) In order to derive an appropriate cost function to investigate the effect of multiple objectives (i.e. setup times and tardiness), the following objective function which is a linear combination of the two objectives is used:

$$\textit{Objective function, } H = \theta\,(\textit{tardiness}) + (1-\theta) * (\textit{setup times})$$

In this representation, θ is used to represent the tradeoff between the two objective functions, tardiness and setup times. When $\theta = 1$, the formulation reduces to the single objective of minimizing tardiness, and when $\theta = 0$, it reduces to the single objective of minimizing setup times. When $0 < \theta < 1$, the simulated annealing heuristic needs to consider a weighted sum of tardiness and setup times. Therefore, it should be expected that the problems are easier to solve when $\theta = 0.0$ or $\theta = 1.0$.

(c) This research investigates the effect of the initial solutions on the final solution. Experimental results on the effect of the initial solution on the performance of simulated annealing is inconclusive.

(d) Johnson, *et al.* (1989) concluded that longer annealing runs produced better results, whereas Laursen (1993) reported that the iterations needed to obtain a good solution in their experiment are very small. However, this factor was expected to be a significant factor in determining the quality of

the final solution. Longer simulation runs are expected to produce better solutions, but at the expense of higher computation cost.

(e) Ragatz (1993) and Rubin and Ragatz (1995) indicated that tightness of due dates and the number of jobs to be scheduled are major factors affecting the effectiveness of their solution approaches. Tightness of due dates indicates how closely the due dates of individual jobs are related to each other. This research investigates the effects of the number of jobs to be scheduled and tightness of due dates on the final solutions.

(f) This research also investigates various parameters affecting the simulated annealing schedule, including:

(i) The starting temperature.
(ii) The final temperature at which the system was considered 'frozen'.
(iii) The temperature decay rate.

3.3 Problem Description

Minimizing tardiness or minimizing setup times separately is very similar to the classic Traveling Salesman Problem (TSP), except that the objective function of tardiness is not linear in job completion times whereas the objective function of TSP is linear in distance traveled. This subtle difference makes minimizing tardiness as an objective a more challenging problem. The actual best solution to either of the problems separately is an np-complete problem, and the time required to solved it on a computer grows exponentially with the number of jobs to be sequenced. The solution approach becomes more complex when the scheduling objectives involve both minimizing setup times and minimizing tardiness sequentially.

Suppose there are n jobs $\{1, 2, 3, ..., n\}$ available at time zero for processing on a continuously available machine (i.e., no machine breakdowns) that can process only one job at a time. For each job i, the processing time (p_i), due date (d_i), and the sequence dependent setup times (s_{ij}) are assumed known with certainty when jobs arrive at the shop. The variable s_{ij} is the sequence dependent setup time if job j follows job i in the processing sequence. The machine is assumed to be idle at time zero. The tardiness of job i can be defined as $T_i = \max\{0, \text{completion time}_i - \text{due date}_i\}$, and the setup times if job j follows job i in the sequence is s_{ij}. Completing a job before its due date does not affect the tardiness objective. This research addressed how to schedule the set of n jobs to minimize a weighted sum of setup times and tardiness.

3.4 Simulated Annealing

As discussed in chapter 2, simulated annealing was used to solve the class of scheduling problems considered in this research. Although the general approach to solving problems using simulated annealing is well known, the issues to be addressed in this research required careful consideration of ways to implement the simulated annealing approach. Specifically, simulated annealing requires generating the move set, specifying the probability of accepting a potential move and setting the values for parameters that control the cooling schedule. These aspects of simulated annealing are discussed in the following sections.

3.4.1. The Boltzmann Distribution

The Boltzmann probability function is $p(c) = \frac{1}{1 + e^{\Delta H/T}}$

where,

T	=	Temperature
ΔH	=	$H' - H$
H	=	Value of the objective function before the state changes, and
H'	=	Value of the objective function after the state changes

As the parameter T decreases, the equation makes it clear that changes that decrease the function are more likely than those that increase it. At very high T compared to ΔH, $p(c)$ is approximately 0.5. That means at high temperature, the simulated annealing schedule has equal probability of accepting or rejecting a worse sequence. At moderate T, the higher the ΔH, the less likely the simulated annealing schedule will accept a change that leads to a worse sequence. At very small T, the random behavior of the Boltzmann distribution is eliminated, and $p(c)$ approaches zero. Thus, the probability of accepting a worse sequence approaches zero.

3.4.2. Perturbation Scheme - the move set

The first step in applying the simulated annealing schedule is to generate an initial feasible schedule. Then, the move set or the perturbation scheme must be defined. The random perturbation scheme used in this research involved: (1) generate two integer random numbers from uniform distribution between 1 and n, where n is the total number of jobs in the sequence, (2) let these two random numbers represent the job numbers, (3) exchange the position of these two jobs in the sequence.

3.4.3. Simulated Annealing Scheme

The simulated annealing scheme can be summarized as:

1. Set the control parameters:

   1.1 Initial temp (T_{max})
   1.2 Final temp (T_0) at which the system is considered *frozen*,
   1.3 Temperature decay rate, (r)

2. While not yet *frozen*, do the following

   2.1 Perform the following loop *STEP* times
   (*STEP* = 1, 50, or 100)

      2.1.1 Generate 2 different integer random #'s between 1 and *n*.
      Let these 2 random #'s represent job #'s.
      Exchange the position of these 2 jobs.
      2.1.2 Let $\Delta H = H' - H$
      2.1.3 If $\Delta H \leq 0$ (downhill move), accept the new sequence.
      2.1.4 If $\Delta H > 0$ (uphill move),

         (i) calculate the probability of accepting that change from the Boltzmann Distribution, $P(c) = \frac{1}{1+e^{\Delta H/T}}$,

         (ii) Select *h*, a uniformly distributed random number between 0 and 1.
         2.1.4.1 If $P(c) > h$, accept the new sequence.
         2.1.4.2 If $P(c) \leq h$, reject the new sequence, and retain the previous sequence prior to step 2.1.1.

      2.1.5 Return to step 2.1.1

   2.2 Set T = rT (reduced temperature).

3. Return H.

4. End of Program.

3.4.4. Setting The Control Parameters

The three control parameters (T_{max}, T_0, r) must be determined *a priori* before the simulated annealing scheme can be applied to solve the research problem. Pilot runs of the research problems and the probability function of the Boltzmann distribution are used to derive a good estimate for these three parameters. The fundamental principle of simulated annealing is to allow an equal chance of rejecting or accepting a worse solution initially. The chance of rejection is slowly decreased to zero by means of a control parameter known as 'temperature'. This is achieved by slowly decreasing T_{max} to T_0 at the rate of (1 - r) percent. These three control parameters were derived as followed:

(I) The Initial Temperature (T_{max})

At very high temperature (T), the simulated annealing scheme should have approximately 50 percent chance of rejecting or accepting a worse solution. Therefore, it is essential to conduct pilot runs to estimate the worst deterioration of the objective function value if the sequence of two jobs were switched. The worst case scenario is derived from the 50-job case with Tradeoff Parameter (θ) equals to 1. The deterioration of the objective function ΔH did not exceed 1,000. Therefore, the highest ΔH was fixed at 1,000. Once the highest ΔH was fixed, T_{max} was calculated easily from the Boltzmann probability function, p(c). The question was to determine the value of T_{max} so that $p(c) \approx 0.50$. Theoretically, p(c) approaches 0.50 only if T_{max} is infinitely large. For the purpose

of this research, the value of T_{max} was computed by arbitrarily setting p(c) = 0.475, a value sufficiently close to 0.50.

Boltzmann Distribution, $P(c) = \frac{1}{1+e^{\Delta H/T}} \approx 0.5 \; ; \; \Delta H = 1{,}000$

$$\begin{aligned}
\text{Let} \quad & \frac{1}{1+e^{\Delta H/Tmax}} = 0.475 \\
\Rightarrow \quad & e^{\Delta H/Tmax} = 1/0.475 - 1 \\
\Rightarrow \quad & e^{\Delta H/Tmax} = 1.105 \\
\Rightarrow \quad & (\Delta H/T_{max}) \ln e = \ln 1.105 \\
\Rightarrow \quad & T_{max} = (\Delta H * \ln e) / \ln 1.105 \\
& \approx 10{,}000
\end{aligned}$$

(II) The Final Temperature (T_0)

T_0 must be selected such that the probability of accepting a worse sequence approaches zero, that is, p(c) ≈ 0. The smallest possible incremental value of the objective function ΔH was 0.5 (when θ = 0.5). Theoretically, p(c) approaches 0.00 only if T_0 is close to zero. For the purpose of this research, the value of T_0 was computed by arbitrarily setting p(c) = 0.007, a value sufficiently close to 0.00. Therefore, the value of T_0 was derived by:

$$\begin{aligned}
\text{Let} \quad & \frac{1}{1+e^{\Delta H/To}} = 0.007 \\
\Rightarrow \quad & e^{\Delta H/To} = 1/0.007 - 1 \\
\Rightarrow \quad & e^{\Delta H/To} = 141.857 \\
\Rightarrow \quad & (\Delta H/T_0) \ln e = \ln 141.857 \\
\Rightarrow \quad & T_0 = (\Delta H * \ln e) / \ln 141.857 \\
& \approx 0.10
\end{aligned}$$

(III) The Temperature Decay Rate (r)

The Temperature Decay Rate, r, should be as small as possible so as to provide a better probability for the annealing schedule to escape local optimal. However, the annealing time would be unacceptably long if r was too small. With T_{max} and T_0 fixed, the temperature decay rate was derived by considering the additional computation time required at each 0.5 percent increment. Figure 1 plotted the number of iterations required to bring T_{max} to T_0 at temperature decay rate from 0.90 to 1.00. Theoretically, r = 1.00 is impossible because that will require indefinite annealing time. The graph clearly indicates that the marginal computation time required for r = 0.90 to r = 0.99 is rather insignificant. However, the marginal time increased significantly when r > 0.995. Therefore, r was selected at 0.995.

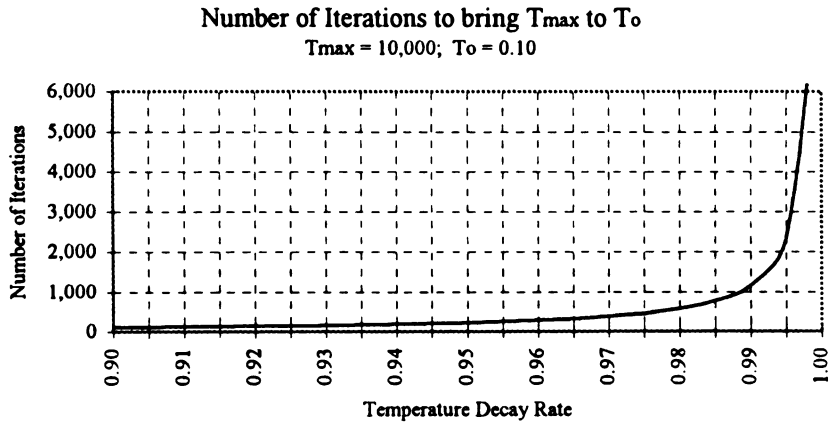


Figure 1: Number of Iterations versus Temperature Decay Rate

3.5 Experimental Factors

This research investigated a total of five experimental factors. Three of the experimental factors were set at 3 levels each, and the other two experimental factors were set at two levels each. The full factorial experimental design consisted of a total of 108 (3×3×3×2×2) treatments with ten test problems in each treatment cell. The factors were: (1) the types of initial solution, *INIT*, (2) the average number of updates per temperature range, *STEP*, (3) the number of jobs in the sequence, n, (4) tightness of due dates, *RD*, and (5) the tradeoff parameter, θ, used to determine the tradeoff between tardiness and setup times.

3.5.1. Means of Generating the Initial Solution (INIT)

Theoretically, simulated annealing tends to yield a near optimal solution independent of the initial solution. Johnson, *et al.* (1989), Potts and Wassenhove (1991), and Shang (1993) reported that starting the annealing schedule at a good initial solution might produce better final solution than starting at a randomly generated initial solution. Intuitively, this seems to be a valid claim since the initial solution partially determines how the final solution settles into a local or global minimum, and also the ability of the simulated annealing schedule to avoid a local minimum. However, Suresh and Sahu (1993) report that a very important feature of simulated annealing is the independence of the final solution from the initial solution.

This research investigates the effects of *better than randomly generated* initial solutions compared to *randomly generated* (*RND*) initial solutions. The *randomly generated* initial solutions are generated by the computer using pseudo random numbers. The *randomly generated* initial solutions are improved by performing pair-wise 'swaps' of two jobs to obtain the *better than randomly generated* initial solutions. Therefore, in an n-job sequence, the pairs (1, 2), (1, 3), (1, 4) ... (1, n), (2, 3), (2, 4) ... (2, n), (3, 4) ... (n-1, n) were checked, and adopted if it lead to a better solution. The *better than randomly generated* initial solution is called the pairwise swapped (*PWS*) initial solution. This factor relates to the simulated annealing heuristic and average objective value will be used as the performance measure.

3.5.2. Number of Updates Between Changes in Temperature (STEP)

In each temperature range, a larger number of updates was expected to produce better solutions than a lower number of updates. This is due to the fact that a larger average number of updates increase the probability escaping a local optimum. However, using an unduly large number of updates will lead to wasted computation time. In order to investigate the influence of this factor on the quality of the final solution and the computation cost of the heuristic, this factor was set at three levels, 1, 50 and 100 updates. One update per temperature range was considered 'low', fifty updates were considered 'medium', whereas one hundred updates were estimated to be large enough for the perturbation scheme to get out of local minimum. This procedure is known as

metropolis loop and the step is called the *metropolis step* (Otten and Ginneken, 1989). This factor relates to the simulated annealing heuristic. Average objective value will be used as the performance measure.

3.5.3. Number of Jobs In The Sequence (n)

The number of jobs in the sequence was expected to be an important factor affecting the effectiveness of simulated annealing. In order to investigate how the solutions generated by simulated annealing depended on problem size, this factor was set at n = 10, 30, 50 jobs. However, prior expectation was that the actual global optimum solutions could not be identified for larger problems. Therefore, the quality of the final solution was compared based on the average objective function value per job to investigate if higher number of jobs lead to larger averages. Averaging the total objective value works well for the setup problems, but not for problems involving tardiness, since tardiness is not a linear function of the number of jobs in the sequence. Investigation of this factor will be limited to the setup problems only (i.e. $\theta = 0.0$).

3.5.4. Tightness of Due Dates - Relative Range of Due Date (RD)

The approach for generating the uniformly distributed due date was consistent with Rinnooy Kan, *et al.* (1975) and Ragatz (1993). Potts and Van Wassenhove (1985) used a different method for generating due dates. Their approach generates negative due dates when both the tardiness factor and the relative range equal to 1. For this research, the

mean of due dates was set equal to $(1\text{-}TF)\times(n)\times$(mean processing time) = $(1\text{-}\ 0.5)\times(n)\times(100) = 50n$, and its range was set equal to $(RD)\times(n)\times$(mean processing time). *RD* is the relative range of due dates and *n* is the number of jobs. Rinnooy Kan, *et al.* (1975), Ragatz (1993), and Rubin and Ragatz (1995) reported that the problem was more difficult to solve with a narrow relative range of due date. The relative range of due date, *RD*, is an experimental factor in this study. *RD* was set at 0.2 and 0.5. Levels of *RD* lower than 0.2 are not very realistic, since they indicate a need to look for better ways of promising due dates to customers. In summary, due dates have a mean of $50n$ and a range of:

(i) $20n = (0.2 \times n \times 100)$ when $RD = 0.2$. That is, $d \sim Uniform(40n, 60n)$.
(ii) $50n = (0.5 \times n \times 100)$ when $RD = 0.5$. That is, $d \sim Uniform(25n, 75n)$.

Investigation of this factor will be limited to the 10-job problems only since the optimum solutions for the larger problems are not known. Final solutions will be compared to the optimum solutions. The performance measure, 'percent above optimum', *pctabopt* = (final solution - optimum solution) / optimum solution, will be used.

3.5.5. Tradeoff Parameter (θ)

In the formulation of the objective function, θ was used to determined the tradeoff between the two objective functions, tardiness and setup times. If $\theta = 1$, the formulation represents the single objective of minimizing tardiness, and if $\theta = 0$, it represents the single objective of minimizing setup times. If $0 < \theta < 1$, the objective to be considered is a weighted sum of tardiness and setup times. Therefore, it was expected that the problems

are more difficult to solve, or simply stated, the simulated annealing heuristic was expected to be less efficient in identifying the optimum solutions for multi-objective problems. In order to test this, the tradeoff parameter was investigated at three levels:

(1) $\theta = 0.0$ (single objective minimizing setup times)
(2) $\theta = 0.5$ (minimizing a weighted sum of setup times and tardiness)
(3) $\theta = 1.0$ (single objective minimizing tardiness)

3.5.6. Summary of Experimental Factors

The performance measures used (Section 3.7), factors and levels can be summarized as:

Table 1: Summary of Experimental Factors

Factor	Level 1	Level 2	Level 3	Performance Measure
Types of Initial Solution (*INIT*)	(*RND*)	(*PWS*)	___	*avgfin* *[1]
Number of Updates (*STEP*)	1	50	100	*avgfin*
Number of Jobs (n)	10	30	50	*avgfin* ($\theta = 0$ only) *[2]
Due Date Tightness (*RD*)	0.2	0.5	___	*pctabopt* *[3] ($n = 10$ only)
Tradeoff Parameter (θ)	0.0 (100% Setup)	0.5	1.0 (100% Tardy)	*pctabopt* ($n = 10$ only)

*[1] *avgfin* = average objective function value
*[2] For setup problems only since tardiness is not a linear function of n.
*[3] *pctabopt* = percent of the final solution above the optimum solution (10-job problems only)

3.6 Test Problems

The test problems were adapted from Ragatz (1993). Ragatz (1993) used thirty two test problems for each level of the number of jobs (n = 8, 10, 12, 14, 16) to be scheduled, whereas Rubin and Ragatz (1995) used eight test problems for each level of the number of jobs (n = 15, 25, 35, 45) to be scheduled. In this research, twenty independent sets of jobs (10 each for RD = 0.2 and 0.5) were generated for each level of the number of jobs (n = 10, 30, 50). In other words, ten problems per experimental cell were solved. The approach, distributions and levels of the various parameters chosen were consistent with those considered by Rinnooy Kan, *et al.* (1975) and Ragatz (1993), despite the fact that these two past research involved only the single objective of minimizing tardiness. The following sections describe how each job was generated.

3.6.1. Distribution of Job Processing Time (p)

Job processing time is the time required for a job to be completely processed on the single processor or machine. Ragatz (1993) used normally distributed job processing times with an arbitrarily chosen mean of 100, whereas the variance was an experimental factor chosen at 5^2 and 25^2. The author reported that his branch and bound algorithm has particular difficulty in solving large problem with a narrow due date range and/or a high variance in processing time. However, Rubin and Ragatz (1995) reported processing time variance was not a determinant of comparative performance among genetic search, random search, and Ragatz's branch and bound. This research used a normally distributed processing times with a mean of 100 and variance of 25^2. That is, $p \sim$ Normally(100, 25^2).

3.6.2. Distribution of Setup Times (s)

Ragatz (1993) used uniformly distributed setup times at two levels with a similar mean of 9.5 as an experimental factor, that is, $s \sim$ Uniform(7, 12) and $s \sim$ Uniform(0, 19). The author reported the range of setup times distribution was not a significant factor affecting solution difficulty, possibly due to the fact that the average setup time of 9.5 was too small relative to the mean processing time of 100. Setup times for this research was set at $s \sim$ Uniform(0, 19).

3.6.3. Tardiness Factor (TF)

The mean of the due dates was set equal to (1-*TF*)×(*n*)×(mean processing time), where *n* is the total number of jobs to be processed. This approach is consistent with past research. The tardiness factor determines the approximate percentage of jobs in a random sequence that will be tardy. For example, $TF = 0.50$ means approximately fifty percent of all the jobs are expected to be tardy in a random sequence. Rinnooy Kan, *et al.* (1975) reported that solution difficulty increased steadily from $TF = 0.20$ to $TF = 0.80$, whereas, Ragatz (1993) reported that solution difficulty increased from $TF = 0.40$ to $TF = 0.60$. Rubin and Ragatz (1995) reported that the scheduling problem was rather trivial with a low *TF* and wide due date range. They reported that the problem was more difficult to solve when the tardiness factor was at the moderate level with a narrow due date range. Due date range is an experimental factor in this research. Tardiness Factor (*TF*) was set at a moderate level of 0.5 in this study.

3.7 Performance Measures

The performance measure for testing the experimental factors specific to the simulated annealing heuristic (i.e. *INIT* and *STEP*) is the average objective value per job. By using the average instead of the total objective value, it is possible to compare problems with different number of jobs. Data from 10, 30 and 50-job problems will be used of this analysis. The effect of the number of job (n = 10, 30 and 50) will also be tested. However, this will be limited to the pure setup problems (i.e. $\theta = 0.0$) only, because tardiness is not a linear function of the number of jobs in the sequence.

The performance measure for testing the 'quality' of the solution is the percent of the final solution above the optimum solution, *pctabopt* = (final solution - optimum solution) / optimum solution. Analysis of solution quality will be limited to the 10-job problems. The impact of θ, *RD* and n on the quality of the 10-job problems will be investigated.

CHAPTER 4: DATA VALIDATION

4.1 Introduction

The job generator was coded using the standard version of Microsoft Visual Basic 3.0. Twenty sets of jobs each were generated for n = 10, 30 and 50 according to the predefined distribution discussed in section 3.5. Kolmogorov-Smirnov and χ^2 test were used to check whether the due dates, setup and processing times of each set of jobs conformed to the desired distributions. Wald-Wolfowitz Runs test was also used to check if prior data generated influenced the values of subsequent data.

4.2 Due Date: $d \sim U(40n, 60n)$ or $d \sim U(25n, 75n)$

H_0: Sample Distribution = Population Distribution
H_1: Sample Distribution ≠ Population Distribution

Kolmogorov-Smirnov (K-S) test was used to check whether due dates of jobs generated came from a uniform distribution between $40n$ and $60n$ when RD = 0.2, and between $25n$ and $75n$ when RD = 0.5. The K-S test was more appropriate than the χ^2 test because of the wide range of due dates which could be assumed to be a continuous variable. The sampling distribution of the K-S statistics was based on the assumption of a continuous population variable. However, it should be noted that the actual due dates distribution was discrete rather than continuous. Using K-S test for discrete population variable is conservative, in the sense that the actual p-value does not exceed that given by the continuous population variable. It could be much smaller (Berry and Lindgren, 1990).

Large 2-tailed *p*-values of the K-S test shown in Tables 2 and 3 (except in three cases) implied that the null hypothesis H_0 could not be rejected at $\alpha = 0.05$. Therefore, there was no reason to believe that due dates were not uniformly distributed between $40n$ and $60n$ when $RD = 0.2$, and $25n$ and $75n$ when $RD = 0.5$.

Table 2: $RD = 0.2$; Due Date, $d \sim U(40n, 60n)$

Problem #	# of Jobs	Mean	Std Dev	Min	Max	Runs Test	K-S
1	10	535.00	52.69	412	598	0.0950	0.1712
2	10	525.00	51.82	422	595	0.3143	0.4038
3	10	521.20	61.26	418	592	0.3143	0.4059
4	10	507.10	51.07	409	587	0.7373	0.8239
5	10	491.50	60.09	402	586	1.0000	0.9995
6	10	503.60	54.85	408	568	1.0000	0.3802
7	10	487.00	41.83	446	578	0.3143	0.2037
8	10	483.60	57.45	408	588	0.3143	0.7067
9	10	538.50	59.28	407	600	0.7373	0.0536
10	10	500.10	76.30	406	594	0.3143	0.3546
1	30	1503.67	167.35	1213	1784	0.8526	0.7596
2	30	1510.87	181.90	1233	1800	0.3529	0.8052
3	30	1542.10	193.66	1202	1799	0.0945	0.1152
4	30	1478.67	181.82	1202	1754	0.5772	0.7528
5	30	1520.93	150.13	1245	1779	0.8526	0.4448
6	30	1475.73	178.64	1207	1799	0.3529	0.3018
7	30	1512.13	178.85	1203	1784	0.5772	0.2024
8	30	1500.10	171.84	1208	1793	0.3529	1.0000
9	30	1480.93	167.15	1236	1775	0.3529	0.4533
10	30	1423.77	158.74	1232	1702	0.3529	0.0763
1	50	2482.38	292.63	2021	2955	0.2530	0.8766
2	50	2573.94	306.42	2056	2994	0.3913	0.1004
3	50	2471.16	279.70	2028	2971	0.7751	0.8864
4	50	2476.74	284.32	2006	2999	0.5676	0.6318
5	50	2568.74	242.47	2019	2941	0.2530	0.0311
6	50	2496.76	291.92	2023	2996	0.7751	0.9105
7	50	2508.54	298.84	2018	2999	0.0864	0.8839
8	50	2545.44	287.00	2050	2955	0.5676	0.5076
9	50	2425.82	258.40	2005	2984	1.0000	0.1011
10	50	2518.02	298.48	2000	2989	0.3913	0.3683

Table 3: *RD* = 0.5; Due Date, *d* ~ U(25*n*, 75*n*)

Problem #	# of Jobs	Mean	Std Dev	Min	Max	Runs Test	K-S
1	10	447.40	163.13	258	734	0.7373	0.4852
2	10	470.20	147.01	268	654	0.3143	0.8856
3	10	494.50	137.22	289	669	0.3143	0.8654
4	10	454.30	92.40	331	586	1.0000	0.9330
5	10	533.60	138.66	326	708	0.7373	0.8540
6	10	506.40	146.96	267	666	0.7373	0.2267
7	10	428.30	107.80	314	654	0.7373	0.2027
8	10	505.20	143.85	297	712	1.0000	0.9620
9	10	482.70	170.30	255	750	0.7373	0.5310
10	10	512.40	141.51	330	742	1.0000	0.6830
1	30	1368.97	452.34	752	2223	0.8526	0.3180
2	30	1584.07	452.66	841	2213	0.8526	0.3463
3	30	1555.17	477.38	770	2231	0.8526	0.5175
4	30	1458.77	471.24	765	2153	0.5772	0.5527
5	30	1413.03	431.14	778	2182	0.5772	0.6578
6	30	1442.10	458.42	752	2222	0.0157 *	0.4273
7	30	1484.67	473.13	798	2243	0.8526	0.4074
8	30	1522.73	433.69	818	2174	0.3529	0.4271
9	30	1468.80	435.15	794	2222	0.8526	0.6733
10	30	1494.60	461.37	792	2230	0.0945	0.6941
1	50	2467.98	709.84	1276	3734	0.5676	0.9273
2	50	2579.82	621.76	1283	3596	0.2530	0.0121 *
3	50	2409.40	684.03	1260	3674	0.1530	0.6570
4	50	2602.76	604.36	1374	3739	0.0222 *	0.5779
5	50	2567.94	782.58	1273	3738	0.7751	0.2561
6	50	2504.68	688.33	1418	3683	0.7751	0.5592
7	50	2488.68	750.79	1270	3741	0.3913	0.9495
8	50	2239.76	651.30	1323	3747	0.5676	0.0011 *
9	50	2556.82	683.44	1263	3731	0.7751	0.6566
10	50	2641.90	627.15	1459	3710	0.5748	0.7071

* significance at $\alpha = 0.05$

4.3 Processing Time: $p \sim N(100, 25^2)$

H_0: Sample Distribution = Population Distribution
H_1: Sample Distribution ≠ Population Distribution

The Kolmogorov-Smirnov (K-S) test was used to test the sampling distribution of processing time. Tables 4, 5 and 6 showed large 2-tailed *p*-values of K-S statistics for all the 10-job, 30-job, and 50-job problem sets. The null hypothesis H_0 could not be rejected at $\alpha = 0.05$. There was no reason to believe that processing time was not normally distributed with a mean of 100, and a standard deviation of 25.

Table 4: 10-job Problem; Processing Time, $p \sim N(100, 25^2)$

Problem #	RD	Mean	Std Dev	Min	Max	Runs Test	K-S
1	0.2	105.60	37.83	44	173	0.3143	0.9950
2	0.2	91.60	21.85	63	136	0.7373	0.9891
3	0.2	104.90	24.02	71	140	0.3143	0.7545
4	0.2	100.90	22.24	52	127	0.3143	0.6755
5	0.2	103.90	19.92	71	148	0.3143	0.8687
6	0.2	107.20	22.79	72	138	0.3143	0.8941
7	0.2	105.90	21.19	68	130	1.0000	0.7394
8	0.2	94.60	21.30	54	118	0.3143	0.9586
9	0.2	106.20	18.77	75	135	0.7373	0.9882
10	0.2	97.80	26.43	46	151	0.7373	0.6288
1	0.5	99.00	18.51	64	120	1.0000	0.6033
2	0.5	103.70	23.16	55	126	0.7373	0.5288
3	0.5	86.60	15.65	47	103	1.0000	0.2952
4	0.5	97.40	35.07	36	161	1.0000	0.9567
5	0.5	105.20	29.27	59	165	0.7373	0.3734
6	0.5	97.70	28.02	33	128	0.7373	0.8731
7	0.5	99.60	18.63	75	130	0.7373	0.8571
8	0.5	102.00	26.18	47	126	0.7373	0.5049
9	0.5	100.00	27.01	63	149	0.3143	0.9333
10	0.5	93.50	19.35	65	115	0.7373	0.6979

Table 5: 30-job Problem; Processing Time, $p \sim N(100, 25^2)$

Problem #	RD	Mean	Std Dev	Min	Max	Runs Test	K-S
1	0.2	97.27	24.30	42	145	0.1934	0.9079
2	0.2	101.93	22.80	60	158	0.8526	0.6436
3	0.2	96.00	28.40	27	168	0.8325	0.9908
4	0.2	102.37	25.05	50	151	0.5772	0.8682
5	0.2	102.80	21.03	57	147	0.5926	0.8315
6	0.2	101.63	26.92	48	157	0.5772	0.8847
7	0.2	96.23	25.04	23	137	0.5772	0.9921
8	0.2	104.47	30.88	40	155	1.0000	0.8990
9	0.2	100.70	21.36	53	134	0.8526	0.7932
10	0.2	89.67	22.47	39	130	0.3529	0.9800
1	0.5	96.27	25.10	45	154	0.8526	0.9800
2	0.5	97.93	28.01	53	156	0.8526	0.6304
3	0.5	97.03	23.65	50	144	0.3529	0.6610
4	0.5	98.97	21.34	51	142	0.5772	0.9029
5	0.5	103.93	23.92	62	155	0.5772	0.5905
6	0.5	99.37	23.82	51	140	0.5772	0.9969
7	0.5	103.13	25.37	56	153	0.8715	0.9877
8	0.5	103.93	21.08	56	142	0.0945	0.3347
9	0.5	107.53	25.30	65	160	0.0157 *	0.6264
10	0.5	101.43	23.67	43	157	0.3529	0.8091

* significance at $\alpha = 0.05$

Table 6: 50-job Problem; Processing Time, $p \sim N(100, 25^2)$

Problem #	RD	Mean	Std Dev	Min	Max	Runs Test	K-S
1	0.2	97.04	23.87	54	167	0.0864	0.8193
2	0.2	100.54	24.09	36	144	0.2530	0.3894
3	0.2	96.50	29.57	38	187	0.5676	0.7985
4	0.2	99.62	23.14	38	152	0.7835	0.9994
5	0.2	93.88	28.83	38	193	0.5676	0.7808
6	0.2	101.34	27.93	36	170	0.0097 *	0.8706
7	0.2	99.68	28.21	34	165	0.5748	0.9968
8	0.2	93.64	27.56	36	171	0.9168	0.8451
9	0.2	100.40	25.70	33	154	0.7751	0.9085
10	0.2	99.60	27.27	48	155	0.3913	0.9556
1	0.5	99.22	26.23	25	144	0.3913	0.9306
2	0.5	106.06	24.04	45	184	0.5676	0.8491
3	0.5	105.56	23.33	61	182	0.1530	0.9887
4	0.5	101.90	23.13	49	154	0.3913	0.9855
5	0.5	95.16	24.16	51	156	0.9909	0.9316
6	0.5	97.44	20.75	57	144	0.0541	0.9384
7	0.5	94.38	22.39	48	153	0.3968	0.9290
8	0.5	100.80	28.79	54	188	0.5676	0.8580
9	0.5	100.80	27.65	36	172	0.7751	0.9339
10	0.5	98.22	24.72	55	157	0.5593	0.7466

* significance at $\alpha = 0.05$

4.4 Setup Time: $s \sim U(0, 19)$

H_0: Sample Distribution = Population Distribution
H_1: Sample Distribution ≠ Population Distribution

χ^2 test was used to test the sampling distribution of setup time due to its narrow range. Kolmogorov-Smirnov test was inappropriate here because setup time must be an integer value from 0 to 19. The range of 0 to 19 was too small to be treated as continuous variable. Therefore, χ^2 test was used to test the hypotheses about the relative proportion of cases falling into the 20 mutually exclusive groups (i.e. 0, 1, 2, ..., 19). χ^2 statistics from Tables 7, 8 and 9 showed that the *p*-values were too large to reject the null hypotheses (except four cases) at $\alpha = 0.05$. There was no reason to suspect that setup time was not uniformly distributed between 0 and 19.

4.5 Wald-Wolfowitz Runs Test

H_0: Observations were random
H_1: Observations were not random

Wald-Wolfowitz Runs (Norusis, 1993a, pp382) test was used to check if prior observation affected the values of subsequent observations. Tables 2 to 9 showed large 2-tailed Runs test statistics (except 6 cases). Therefore, the hypotheses of randomness was not rejected at $\alpha = 0.05$. Since majority of the test problems passed the randomness test, no additional investigation of the 6 cases were conducted. These tests suggested that the test problems were indeed generated according to the intended distributions.

Table 7: 10-job (90 setups per case); Setup, $s \sim U(0, 19)$

Problem #	RD	Mean	Std Dev	Min	Max	Runs Test	χ^2
1	0.2	8.61	5.70	0	19	0.3850	0.2048
2	0.2	9.07	5.51	0	19	0.3964	0.6720
3	0.2	9.76	5.82	0	19	0.5363	0.6424
4	0.2	9.94	5.50	0	19	0.5276	0.7837
5	0.2	9.22	5.93	0	19	0.6715	0.6424
6	0.2	10.31	6.04	0	19	0.6394	0.1431
7	0.2	10.12	5.78	0	19	0.7252	0.9529
8	0.2	9.36	5.50	0	19	0.9394	0.9888
9	0.2	10.00	5.91	0	19	0.7565	0.4081
10	0.2	9.06	5.91	0	19	0.9962	0.6124
1	0.5	9.38	5.72	0	19	0.1476	0.4640
2	0.5	9.69	5.94	0	19	0.0269 *	0.4640
3	0.5	10.09	5.44	0	19	0.2891	0.0184 *
4	0.5	9.23	5.98	0	19	0.9051	0.5823
5	0.5	9.20	5.49	1	19	0.6394	0.2048
6	0.5	8.47	5.44	0	19	0.6572	0.7296
7	0.5	10.21	5.54	0	19	0.2681	0.8949
8	0.5	10.44	6.05	0	19	0.6715	0.6124
9	0.5	10.48	5.69	0	19	0.4453	0.4356
10	0.5	10.58	5.49	0	19	0.5720	0.3069

* significance at $\alpha = 0.05$

Table 8: 30-job (870 setups per case); Setup, $s \sim U(0, 19)$

Problem #	RD	Mean	Std Dev	Min	Max	Runs Test	χ^2
1	0.2	9.24	5.91	0	19	0.3738	0.0380 *
2	0.2	9.56	5.90	0	19	0.0786	0.1167
3	0.2	9.41	5.68	0	19	0.9350	0.2912
4	0.2	9.55	5.82	0	19	0.3713	0.8073
5	0.2	9.44	5.85	0	19	0.1224	0.9433
6	0.2	9.47	5.91	0	19	0.8640	0.3006
7	0.2	9.77	5.80	0	19	0.0012 *	0.2288
8	0.2	9.55	5.80	0	19	0.0573	0.0676
9	0.2	9.48	5.64	0	19	0.7581	0.2430
10	0.2	9.69	5.76	0	19	0.7226	0.6124
1	0.5	9.58	5.89	0	19	0.3764	0.5409
2	0.5	9.76	5.80	0	19	0.6765	0.6311
3	0.5	9.64	5.75	0	19	0.9642	0.7364
4	0.5	9.25	5.74	0	19	0.6613	0.2430
5	0.5	9.69	5.82	0	19	0.0613	0.5440
6	0.5	9.68	5.75	0	19	0.0134 *	0.4043
7	0.5	9.36	5.93	0	19	0.6335	0.0884
8	0.5	9.51	5.81	0	19	0.4853	0.9104
9	0.5	9.49	5.95	0	19	0.2359	0.6187
10	0.5	9.58	5.67	0	19	0.1335	0.4980

* significance at $\alpha = 0.05$

Table 9: 50-job (2450 setups per case); Setup, $s \sim U(0, 19)$

Problem #	RD	Mean	Std Dev	Min	Max	Runs Test	χ^2
1	0.2	9.55	5.76	0	19	0.5181	0.2892
2	0.2	9.49	5.70	0	19	0.9356	0.8988
3	0.2	9.27	5.77	0	19	0.4524	0.9125
4	0.2	9.49	5.86	0	19	0.6440	0.8195
5	0.2	9.55	5.84	0	19	0.4922	0.0842
6	0.2	9.34	5.77	0	19	0.7117	0.5356
7	0.2	9.50	5.76	0	19	0.6857	0.1175
8	0.2	9.58	5.80	0	19	0.2577	0.3970
9	0.2	9.32	5.72	0	19	0.6640	0.7444
10	0.2	9.60	5.80	0	19	0.4436	0.4405
1	0.5	9.54	5.74	0	19	0.2809	0.8562
2	0.5	9.27	5.82	0	19	0.9788	0.0381 *
3	0.5	9.50	5.75	0	19	0.1264	0.8133
4	0.5	9.38	5.75	0	19	0.1022	0.5918
5	0.5	9.42	5.87	0	19	0.0711	0.3941
6	0.5	9.31	5.78	0	19	0.0894	0.2560
7	0.5	9.53	5.81	0	19	0.8398	0.9801
8	0.5	9.62	5.82	0	19	0.5495	0.0439 *
9	0.5	9.47	5.72	0	19	0.0896	0.1952
10	0.5	9.47	5.86	0	19	0.5193	0.9379

* significance at $\alpha = 0.05$

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Introduction

This chapter presents the results of the experiments carried out. First, steps taken to ensure the correctness of simulated annealing solution technique are discussed. This is followed by a detailed discussion of the results for the 10, 30 and 50-job problems.

5.2 Programming Code Verification

The simulated annealing scheme of section 3.4.3 was coded using the standard version of Microsoft Visual Basic 3.0 (APPENDIX 4). All the problems were solved on an Intel 486-33 MHz personal computer. Due dates, processing and setup times of jobs were read in as ASCII format from the data files. The following steps were taken to verify the programming codes:

(1) Due dates, processing and setup times read from the data files were immediately written to backup files. Fifteen backup files (5 each for the 10, 30 and 50-job problems) were selected at random to check against the original data files to ensure that the information read was correct. No discrepancy was found.

(2) The initial objective function value of each problem and its sequence were recorded in an output file. Fifteen sequences (5 each for the 10, 30 and 50-job problems) were selected at random to verify the correctness of the initial objective function value. No discrepancy was found.

(3) The final objective function value of each problem and its sequence were also recorded in the output file. Fifteen sequences were selected at random to verify the correctness of the final objective function value. No discrepancy was found.

(4) All subsequent objective function values between changes in temperature range during the entire perturbation for each problem were recorded. Charts of the objective values versus temperature were plotted for some of the problems (e.g., Figures 2 to 5, APPENDICES 2 and 3). No discrepancy was noted.

5.3 Optimum Solutions of 10-Job Problems

The means, variances, optimum and worst solutions for all the 10-job problems were obtained by explicitly enumerating all the 10! possible sequences. The purpose was to establish a benchmark (the optimum solution) for comparing solutions generated by simulated annealing, at least for the 10-job problems. This is not possible for the 30 and 50-job problems due to the huge number of feasible solutions (i.e. 30! and 50!). The results are shown in Tables 10 to 15. Each problem consists of ten jobs to be sequenced, and there are ten problems in each group.

Figures 2 to 6 show the frequency distribution of the objective value for all feasible solutions for 5 problems, two of which are considered as easy problems where simulated annealing identified the optimum solution in all the six combinations of solution

approaches (*STEP* = 1,2,3 and *INIT* = *RND* & *PWS*). Simulated annealing failed to identify the optimum solution for the two problems in Figures 2 and 4. Two of the simulated annealing solution approaches (*INIT* = *RND* with *STEP* = 100 and *INIT* = *PWS* with *STEP* = 50) identified the optimum solution for the problem in Figure 5. All the five problems have unique optimum solution (i.e. frequency for optimum solution = 1). The feasible solutions for all five problems conform to normal distribution, with the worst solution about 8.5 standard deviations away.

Tables 10 and 11 show the optimum and worst solutions for the 10-job problems when *RD* = 0.2 and 0.5 respectively. The objective is to minimize tardiness (i.e. $\theta = 1.0$). The *mean* and *variance* are the average solution and the variance of all the 10! feasible solutions for each problem. The variance of problem #1 (58,671.67) is more than four times larger than the variance of problem #9 (14,629.00), although both the problems have a mean of approximately 1,900. The large variation of the feasible solutions for problem #1 may indicate that it is easier to identify the optimum solution compared to one that has small variation. The rationale is simple. That is, it is always easier to find the best sequence from a group if the differences among them are very wide. However, it is more difficult to select the best sequence if the differences are minor. The optimum solutions when *RD* = 0.2 range from 835 to 1,645, with an average of 1,298.40 and a standard deviation of 258.71 for all the 10 problems. When *RD* = 0.5, the optimum solutions range from 523 to 1,524, with an average of 1,064.70 and a standard deviation of 281.06 for all the 10 problems.

Tables 12 and 13 show the optimum and worst solutions when $\theta = 0.0$. The objective is to minimize setup. The means and variances of these sets of problems are more consistent. The means and variances in Tables 12 very closely matched those in Table 13 because *RD* has no effect on setup objective. The optimum solutions when *RD* = 0.2 range from 11 to 22, with an average of 15.50 for all the 10 problems. When *RD* = 0.5, the optimum solutions are slightly higher and range from 12 to 25, with an average of 20.40.

Table 10: Distribution of Tardiness for all Feasible Solutions (n = 10), RD = 0.2

Problem #	Optimum	Worst	Mean	Variance
1	1143	2693	1900.4168	58671.67
2	835	1970	1370.7949	21719.25
3	1350	2728	1991.2698	31846.58
4	1299	2523	1898.1343	28455.70
5	1485	2635	2103.9255	16128.05
6	1608	2900	2221.5039	21423.85
7 *[1]	1645	2949	2247.7596	25459.63
8	1114	2319	1740.9244	21555.12
9	1442	2534	1956.1603	14629.00
10	1063	2508	1786.1794	35356.82
Average	1298.40	2575.90	1921.7069	27524.57
Std. Dev.	258.71	284.19	256.2510	12728.04

*[1] The frequency distribution of the 7[th] problem (J10RD02G) is shown in Figure 2.

StdDev	Frequency
0.0	1
0.5	253
1.0	4343
1.5	28334
2.0	107581
2.5	256858
3.0	462414
3.5	624758
4.0	670022
4.5	583550
5.0	423874
5.5	265606
6.0	134100
6.5	51467
7.0	13400
7.5	2109
8.0	127
8.5	3
9.0	0
9.5	0
	3,628,800

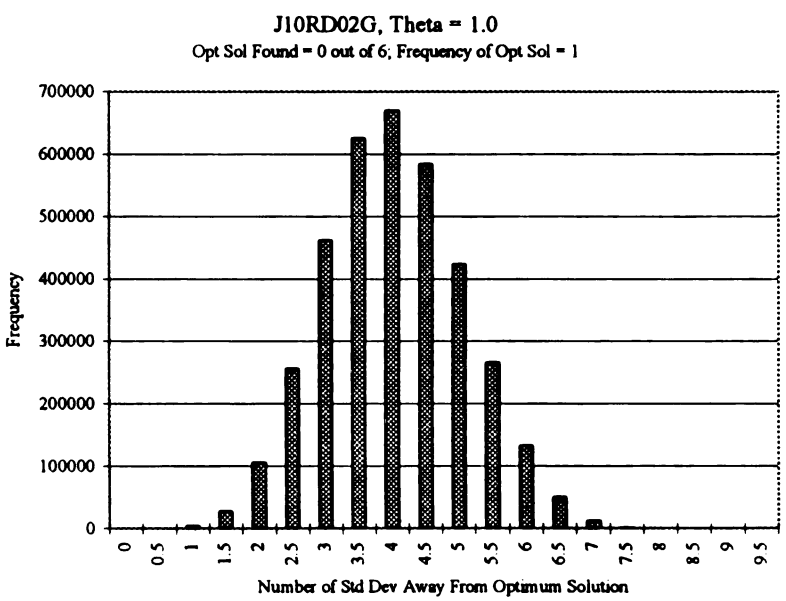


Figure 2: Frequency Distribution of Tardiness for All Feasible Solutions (J10RD02G)

Table 11: Distribution of Tardiness for all Feasible Solutions (n = 10), RD = 0.5

Problem #	Optimum	Worst	Mean	Variance
1	1171	3242	2241.9184	92479.84
2	1319	3340	2311.1629	76031.18
3	523	2236	1430.3046	54369.79
4	1220	2988	2064.4789	67340.54
5	1001	2874	1981.0228	53492.09
6 *[2]	951	2719	1800.8868	44730.39
7	1524	3143	2369.8550	48696.38
8	1084	3061	2049.3896	72288.02
9	1087	3230	2138.0429	70571.11
10	767	2446	1645.7220	48947.87
Average	1064.70	2927.90	2003.2784	62894.72
Std. Dev.	281.06	362.83	299.5843	15282.18

*[2] The frequency distribution of the 6th problem (J10RD05F) is shown in Figure 3.

StdDev	Frequency
0.0	1
0.5	94
1.0	1875
1.5	15448
2.0	65304
2.5	168296
3.0	334584
3.5	523543
4.0	662518
4.5	691921
5.0	556026
5.5	354183
6.0	168521
6.5	64241
7.0	18296
7.5	3519
8.0	418
8.5	12
9.0	0
9.5	0
	3628800

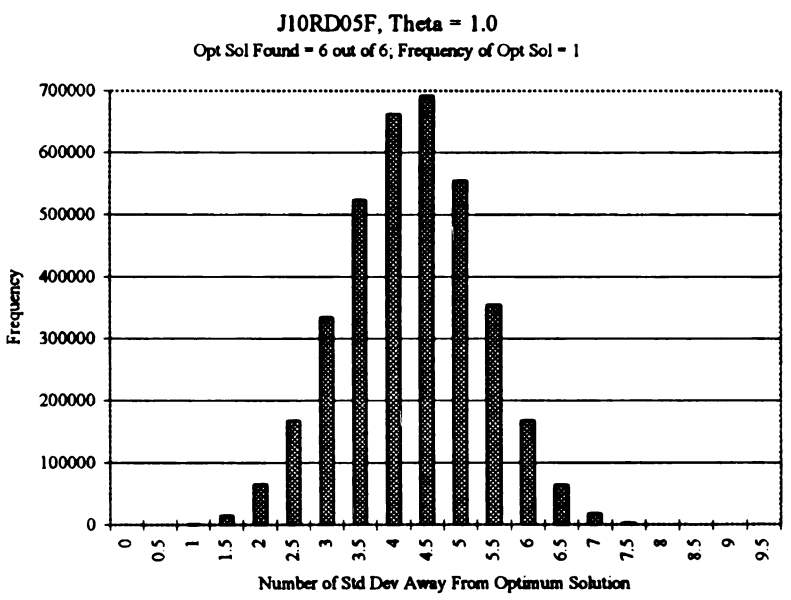


Figure 3: Frequency Distribution of Tardiness for All Feasible Solutions (J10RD05F)

Table 12: Distribution of Total Setup Time for all Feasible Solutions (n = 10), RD = 0.2

Problem #	Optimum	Worst	Mean	Variance
1	13	146	77.5	243.1322
2	12	151	81.6	260.7318
3	14	157	87.8	284.1675
4	22	155	89.5	246.7626
5	13	155	83.0	315.0128
6	14	163	92.8	329.0397
7	19	156	91.1	305.2628
8	11	145	84.2	273.1718
9	22	166	90.0	305.8402
10 *[3]	15	152	81.5	261.4442
Average	15.50	154.60	85.90	282.4566
Std. Dev.	4.03	6.62	5.03	30.0368

*[3] The frequency distribution of the 10[th] problem (J10RD02J) is shown in Figure 4.

StdDev	Frequency
0.0	1
0.5	143
1.0	1407
1.5	10110
2.0	43517
2.5	131544
3.0	292846
3.5	500623
4.0	664658
4.5	698447
5.0	583326
5.5	386961
6.0	218015
6.5	71410
7.0	20769
7.5	4395
8.0	586
8.5	42
9.0	0
9.5	0
	3,628,800

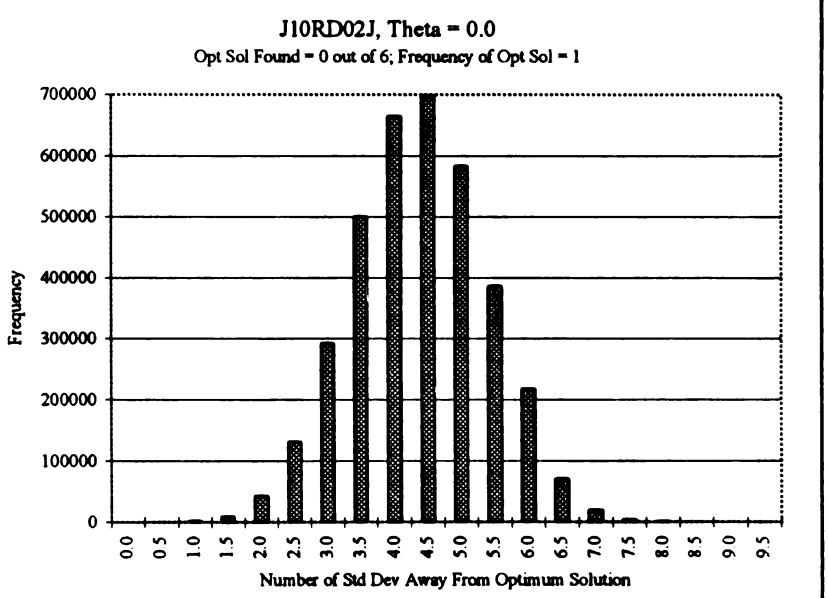


Figure 4: Frequency Distribution of Total Setup Time for All Feasible Solutions (J10RD02J)

Table 13: Distribution of Total Setup Time for all Feasible Solutions (n=10), RD = 0.5

Problem #	Optimum	Worst	Mean	Variance
1 *[4]	17	157	84.4	294.7655
2	22	159	87.2	312.9578
3	27	152	90.8	260.1807
4	12	158	83.1	309.6457
5	24	143	82.8	209.5998
6	14	150	76.2	261.7754
7	21	160	91.9	232.1879
8	18	161	94.0	310.7514
9	25	164	94.3	277.3686
10	24	151	95.2	229.7934
Average	20.40	155.50	87.99	269.9026
Std. Dev.	4.97	6.35	6.28	37.4224

*[4] The frequency distribution of the 1st problem (J10RD05A) is shown in Figure 5.

StdDev	Frequency
0.0	1
0.5	131
1.0	3134
1.5	16895
2.0	73323
2.5	172248
3.0	386156
3.5	605112
4.0	650412
4.5	704917
5.0	481828
5.5	328637
6.0	146341
6.5	44728
7.0	12820
7.5	1862
8.0	247
8.5	8
9.0	0
9.5	0
	3,628,800

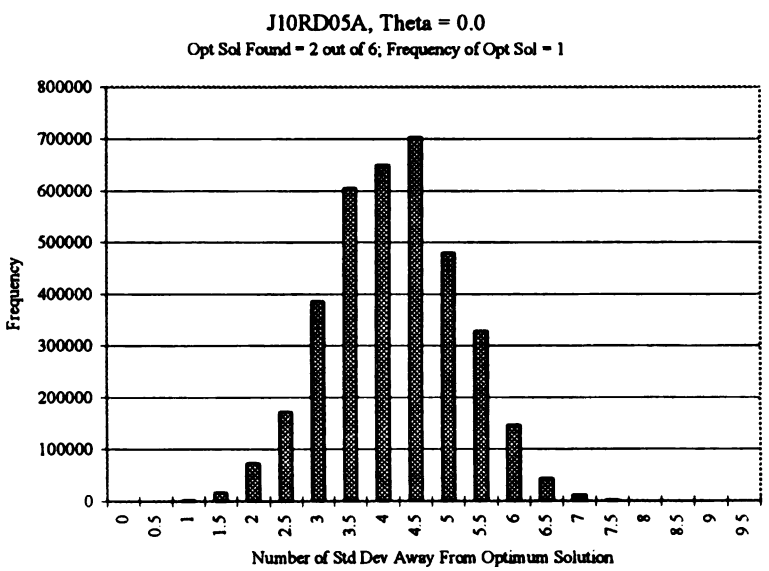


Figure 5: Frequency Distribution of Total Setup Time for All Feasible Solutions (J10RD05A)

Tables 14 and 15 showed the optimum and worst solutions when $\theta = 0.5$. The objective is to minimize an equally weighted combination of setup and tardiness (0.5*{setup+tardiness}). When *RD* = 0.2, the average optimum solutions for all the 10 problems is higher than when *RD* = 0.5. However, its standard deviation is smaller due to the fact that due dates are closer than when *RD* = 0.5. The optimum solutions when *RD* = 0.2 range from 438 to 837, with an average of 665.80 for all the 10 problems. When *RD* = 0.5, the optimum solutions are slightly lower and range from 284.5 to 787.0, with an average of 554.70.

Table 14: Distribution of Objective Value ($\theta = 0.5$) for all Feasible Solutions, *RD* = 0.2

Problem #	Optimum	Worst	Mean	Variance
1	586.5	1409.0	988.9584	15161.2700
2	438.0	1049.5	726.1975	5995.0270
3	695.5	1437.5	1039.5349	8612.5200
4	668.0	1330.5	993.8171	8256.3940
5	754.0	1388.5	1093.4627	5059.5870
6	823.0	1526.5	1157.1519	6548.1160
7	837.0	1544.5	1169.4298	7450.9890
8	569.0	1221.5	912.5622	5729.5100
9	737.5	1342.0	1023.0801	4813.0590
10	549.5	1308.5	933.8397	9520.4180
Average	665.80	1355.80	1003.8034	7714.6890
Std. Dev.	128.63	145.39	129.6320	3047.9391

Table 15: Distribution of Objective Value ($\theta = 0.5$) for all Feasible Solutions, $RD = 0.5$

Problem #	Optimum	Worst	Mean	Variance
1	604.5	1689.5	1163.1592	24211.7900
2	678.0	1737.0	1199.1815	19686.4700
3	284.5	1189.5	760.5523	14336.8300
4	630.5	1559.0	1073.7894	17636.7200
5 *[5]	527.5	1500.0	1031.9114	14033.8000
6	492.5	1423.5	938.5434	11923.6000
7	787.0	1643.0	1230.8775	14139.0000
8	565.5	1604.5	1071.6948	18834.0800
9	574.5	1692.5	1116.1715	18340.1400
10	402.5	1289.5	870.4610	12673.2200
Average	554.70	1532.80	1045.6342	16581.5650
Std. Dev.	140.90	182.51	149.7277	3824.0234

*[5] The frequency distribution of the 5th problem (J10RD05E) is shown in Figure 6.

StdDev	Frequency
0.0	1
0.5	27
1.0	566
1.5	5500
2.0	29818
2.5	103796
3.0	248524
3.5	447208
4.0	621714
4.5	696307
5.0	624742
5.5	450704
6.0	252899
6.5	108155
7.0	31954
7.5	6298
8.0	577
8.5	10
9.0	0
9.5	0
	3,628,800

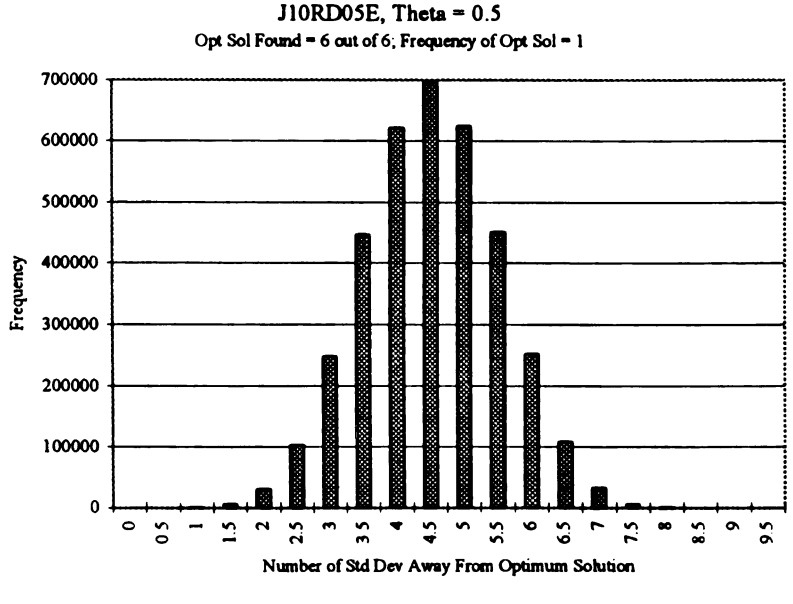


Figure 6: Frequency Distribution of Objective Value for All Feasible Solutions (J10RD05E)

5.4 Number Of Optimum Solutions Identified In The 10-Job Cases

Table 16: Optimum Solution Identified - 10 job Problem

Factors	Level 1	Level 2	Level 3
Initial Solution	*INIT = RND*	*INIT = PWS*	
(180 problems/cell)	36.1 %	43.9 %	
Quality of Solution *			—
Average	12.56	11.75	
Median	0.91	0.53	
Worst % Suboptimal	171.43	227.27	
95% Confidence Interval	8.90 to 16.22	7.87 to 15.63	
TradeOff Parameter	$\theta = 0.0$	$\theta = 0.5$	$\theta = 1.0$
(120 problems/cell)	14.2 %	52.5 %	53.3 %
Quality of Solution *			
Average	33.77	1.62	1.07
Median	22.88	0.00	0.00
Worst % Suboptimal	227.27	20.74	13.39
95% Confidence Interval	27.37 to 40.17	0.97 to 2.27	0.69 to 1.46
Number of Updates	*STEP* = 1	*STEP* = 50	*STEP* = 100
(120 problems/cell)	19.2 %	49.2 %	51.7 %
Quality of Solution *			
Average	22.26	7.71	6.50
Median	5.00	0.09	0.00
Worst % Suboptimal	227.27	100.00	69.23
95% Confidence Interval	15.55 to 28.97	4.77 to 10.64	4.07 to 8.93
Due Date Tightness	*RD* = 0.2	*RD* = 0.5	
(180 problems/cell)	32.2 %	47.8 %	
Quality of Solution *			—
Average	14.27	10.04	
Median	0.96	0.21	
Worst % Suboptimal	227.27	133.33	
95% Confidence Interval	9.79 to 18.75	7.19 to 12.90	

* *Quality of Solution* indicates percent above the optimum solution, computed by:
pctabopt = (Final Solution - Optimum Solution)/Optimum Solution.

Table 16 shows the percent of the 10-job problems solved to optimality using simulated annealing (final solution = optimum solution). The 30 and 50-job problems were excluded from this analysis because the optimum solutions were unknown. This analysis ignored any interaction effects among the experimental factors. It simply expressed the number of problems solved to optimality as a ratio to the total number of problems. There are a total of 360 problems (2 *INIT* × 3 θ × 3 *STEP* × 2 *RD* × 10 Problems each). Therefore, each cell consists of 180 problems if the experimental factor is at two levels, and 120 problems if the experimental factor is at 3 levels.

'Quality of Solution' in Table 16 indicates the percent of the final solution above the optimum solution. The performance measure, *pctabopt* = (Final solution - Optimum solution) / (Optimum solution). This performance measure will be used to test the experimental factors *RD*, θ and *n*, based on the 10-job problems. *pctabopt* cannot be computed for the 30 and 50-job problems because the optimum solution was not known. 'Average' shows the average percent above optimum solution for all the problems in that cell, and 'Worst % Suboptimal' shows the percent above optimum solution for the worst problem.

Table 16 shows that slightly less percent of the 10-job problems were solved to optimality with *random initial solution* (*RND* = 36.1%) than *better than random initial solution* (*PWS* = 43.9%). *PWS* did not seem to have an advantage in locating the optimum solution. Tradeoff Parameter (θ) indicates that as θ increased from 0 to 0.5, the percent of problems solved to optimality increased substantially from 14.2% to 53.3%. When θ = 1, 53.3% of the problems were solved to optimality. It appears that simulated

annealing and the annealing parameters used in this research are more effective in solving tardiness objective ($\theta = 1$) than setup objective ($\theta = 0$). This could be due to the magnitude of the objective values compared to the annealing parameters used. When $\theta = 0$, the objective was to minimize setup with an average magnitude of approximately 85 (Tables 12 & 13). When $\theta = 0.5$, the objective was to consider a tradeoff between setup and tardiness, with an average magnitude of approximately 1,000 (Tables 14 & 15). When $\theta = 1$, the objective was to minimize tardiness with an average magnitude of approximately 2,000 (Tables 10 & 11). The annealing parameters (temperature decay rate, starting and frozen temperatures) were not experimental factors in this research. They were calculated using the Boltzmann distribution as explained in chapter 3.

Table 16 also indicates that higher number of updates (*STEP*) between changes in temperature range, coincided with higher percent of problems in which optimum solution was identified. When *STEP* was increased from 1 to 50, optimum solutions identified increased almost three fold from 19.2% to 49.2%. However, when *STEP* was increased from 50 to 100, the percent increased to 51.7%, an additional 2.5%. *Due Date Tightness* indicates that 32.2% and 47.8% of the optimum solutions were identified when *RD* = 0.2, and 0.5 respectively. More optimum solutions were identified when *RD* = 0.5, probably due to the fact that it was easier to select or discriminate jobs with wider due dates (i.e. when *RD* = 0.5 instead of 0.2).

Other possibilities for the simulated annealing scheme to perform better on tardiness problems than setup problems include: (1) the perturbation scheme may be structured so that it does a better job of exploring the 'neighborhood' of the optimum

solution for tardiness problems, and (2) there may be more alternate optimum solutions for the tardiness criteria, making it easier to run across one during the simulated annealing process. However, Figures 2 to 6 do not support this rationale. All five problems shown have unique optimum solution, that is, one optimum solution.

5.5 The Objective Values During The Annealing Process

This section briefly discusses the movement of the objective values during the entire annealing process, that is, how the objective values changed as the temperature was decreased. The objective values are plotted on the y-axis, and the annealing temperatures are plotted on the x-axis. The highest temperature used for the annealing process was 10,000. The system was considered 'frozen' when the temperature reached 0.1. Among a total of 1,080 problems (2 *INIT* $\times$ 3 θ $\times$ 3 n $\times$ 3 *STEP* $\times$ 2 *RD* $\times$ 10 Problems each), 12 problems were selected for the purpose of examining the annealing process in detail. Three problems (or graphs) each are plotted in Figures 7 to 10. The only difference among the three graphs in each figure is the number of updates between changes in temperature. That is, the first, second and third graphs used *STEP* = 1, 50 and 100 respectively. These four figures provide insight into how the objective values changed as the temperature was decreased in the simulated annealing process.

The first, second and third graphs in Figure 7 shows the movements of the objective values for a 10-job problem with *STEP* = 1, 50 and 100 respectively. Other job parameters (*RD*, θ, *INIT*, and n) are the same for these three graphs. When *STEP* = 1, it

is obvious that the movement was not as erratic as when *STEP* = 50 or 100. The first graph (*STEP* = 1) shows that at certain points during the annealing process, there was no change in the objective value (horizontal movement of setup) as the temperature was reduced especially between the temperature range 24.4 and 1.2. This phenomenon is not observed in the second and the third graphs, except toward the ending part of the annealing process because when *STEP* = 1, the annealing process has only one chance of accepting a worse solution in each temperature range. Therefore, when it failed to accept a worse solution, the annealing scheme reverted to its previous objective value before the state change, thus resulting in a horizontal movement of setup.

All three graphs in Figure 7 represent solution of the same 10-job problem. The initial solution is 64, and the optimum solution is 14. The final solutions for the first, second and third graphs are 38, 16, and 21 respectively. A close examination of the first graph reveals that the annealing process actually found a better solution than the final solution when the temperature was approximately between 24 and 5 (the three troughs that are lower than the final solution). However, the annealing process accepted a worse solution later on and was unable to return to the better solution found earlier. The same problem was observed in the third graph. This is due to the fact that simulated annealing is a "memoryless" system. It does not store a previously found solution in memory, and thus will not be able to return to a previous solution. Future research will be conducted to study the effect of adding a "memory loop" in the simulated annealing scheme in an attempt to improve the annealing process.

Figure 7 also confirmed that the final temperature (T_0) of 0.1 determined in *Section 3.4.4. Setting The Control Parameter* was small enough to ensure a near zero probability of accepting a worse solution toward the end of the annealing process. Indeed the annealing process could be considered 'frozen' when the temperature was approximately equal to 1. Setting T_0 smaller than 0.1 is unlikely to lead to a better final solution.

Figure 8 shows the annealing process of another 10-job problem, where the objective is to minimize tardiness. The first, second and third graphs used *STEP* = 1, 50 and 100 respectively. The initial solution for these three graphs is 2,190 and the optimum solution is 1,350. Figure 8 demonstrates the same erratic behavior observed in Figure 7, when *STEP* were set at 50 and 100. The third graphs reveals that the final solution of 1,363 is not the lowest sampled during the annealing process. The objective value (Tardiness) was actually lower when the annealing temperature was around 24.4. Indeed, it found the optimum solution but was unable to return to that solution as temperature was reduced. Since its final solution is not the optimum solution, this problem was not counted as solved to optimality in Table 16. The first and second graphs report final solutions of 1,406 and 1,350 respectively (optimum solution = 1,350).

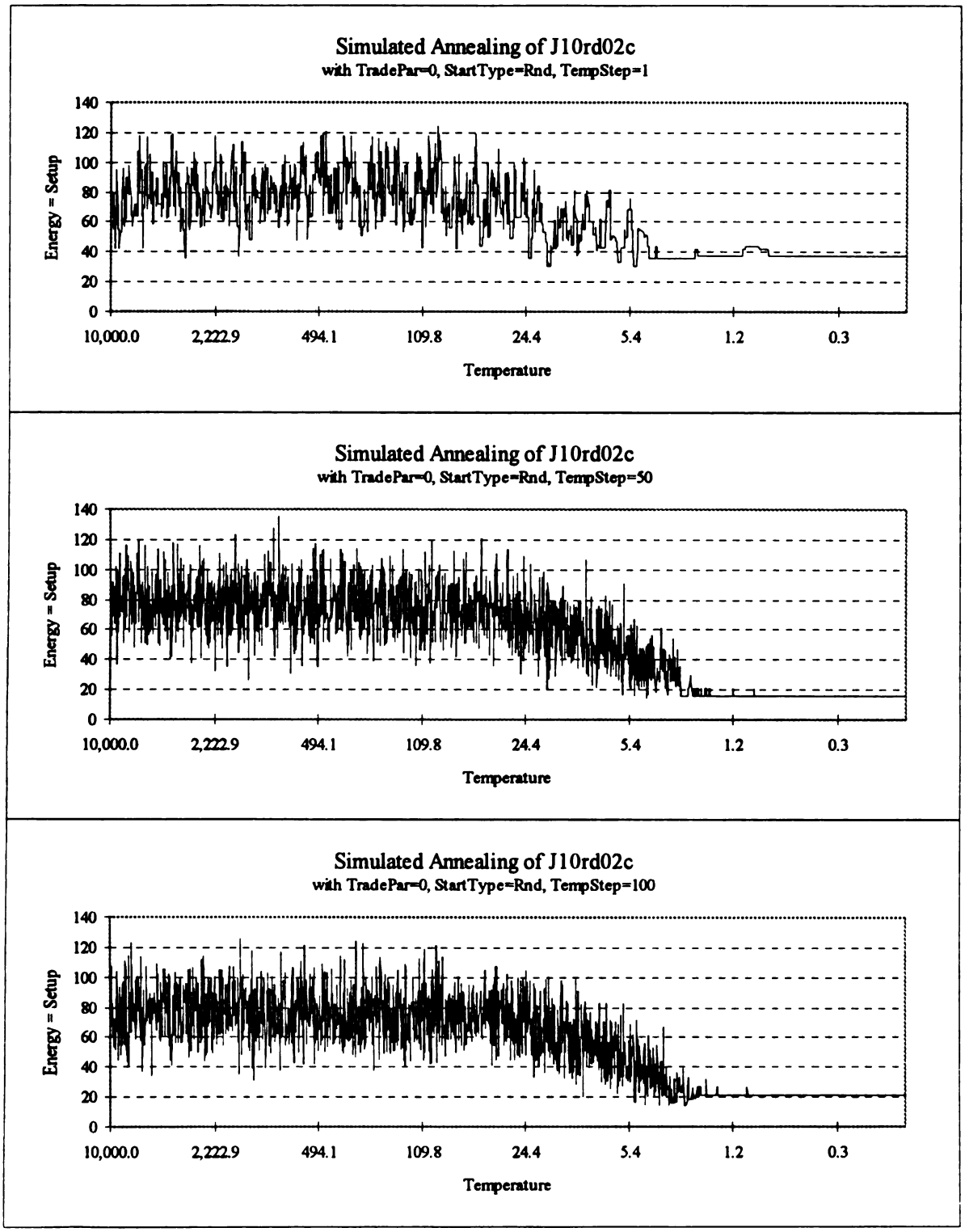


Figure 7: Objective Values as Temperature Decreased (10-job, $\theta = 0$, *RND*)

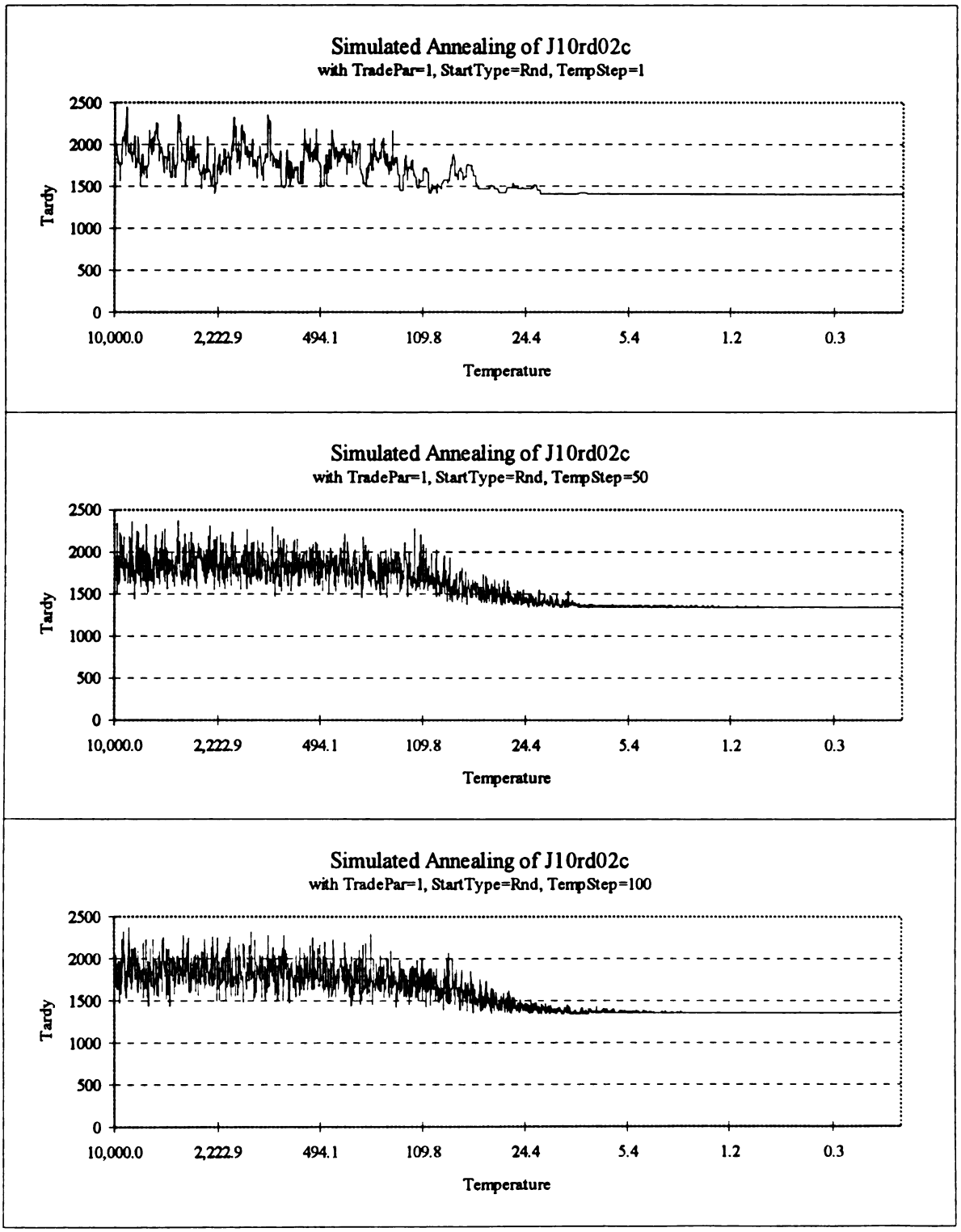


Figure 8: Objective Values as Temperature Decreased (10-job, $\theta = 1$, *RND*)

Figure 9 shows the objective values of a 30-job problem with $\theta = 0$, $RD = 0.5$, and $INIT = PWS$. The "memoryless" aspect of simulated annealing is not a problem for the 30-job problems. Unlike the 10-job problems, there is no indication that the annealing process was unable to return to a better solution found previously. The ratio of the objective value to the temperature decay rate might play a role in whether simulated annealing is able to return to a better solution found previously. All three graphs show that the objective values (setup) were moving around 250 until the temperature reached approximately 24.4, at which point, the objective values slowly decreased and settled into the final solutions. The pairwise swapped initial solution (*PWS*) for these three graphs is 94, and the final solution is 46, 41 and 51 respectively. The best known solution for this problem is 41. Although the initial solution was 94, setup actually increased to more than 300 before it settled down into its final solution.

The three graphs in Figure 9 also indicates that the objective values did not decrease until the temperature reached approximately 24.4. This might be an indication that using a starting temperature of 10,000 was too high for the 30-job problems, thus resulting in wasted annealing time. It might be more beneficial to use a smaller temperature decay rate rather than starting at a very high temperature. However, the starting temperature of 10,000 was selected for all the 10, 30, and 50-job problems for this experiment for ease of comparison. Two additional runs of the 30-job problem with lower starting temperature and smaller temperature decay rate did not produce better results. (See APPENDIX 3). Future research will be conducted to investigate the effect of the ratio of the objective value to the annealing parameters on the solution.

Figure 10 shows the objective values of a 50-job problem, with $\theta = 1$, $RD = 0.5$, and $INIT = PWS$. The initial pairwise swapped solution is 20,192. The first, second and third graphs report final solutions of 18545, 16562 and 16661 respectively. The second graph reports the best known solution for this problem. The first graph visually indicates that using a better than randomly generated initial solution may not reduce the annealing time. The objective value (tardiness) actually increased steadily from 20,192 to more than 40,000 before it settled down into its local optimum solution. This is consistent with the analogy of simulated annealing where one heats up the system and let it slowly cools until it settles into a steady state (thus the local or optimum solution). Figure 10 also shows a distinct difference from Figures 7, 8 and 9. Its objective values started to decrease steadily at a very early stage of the annealing process.

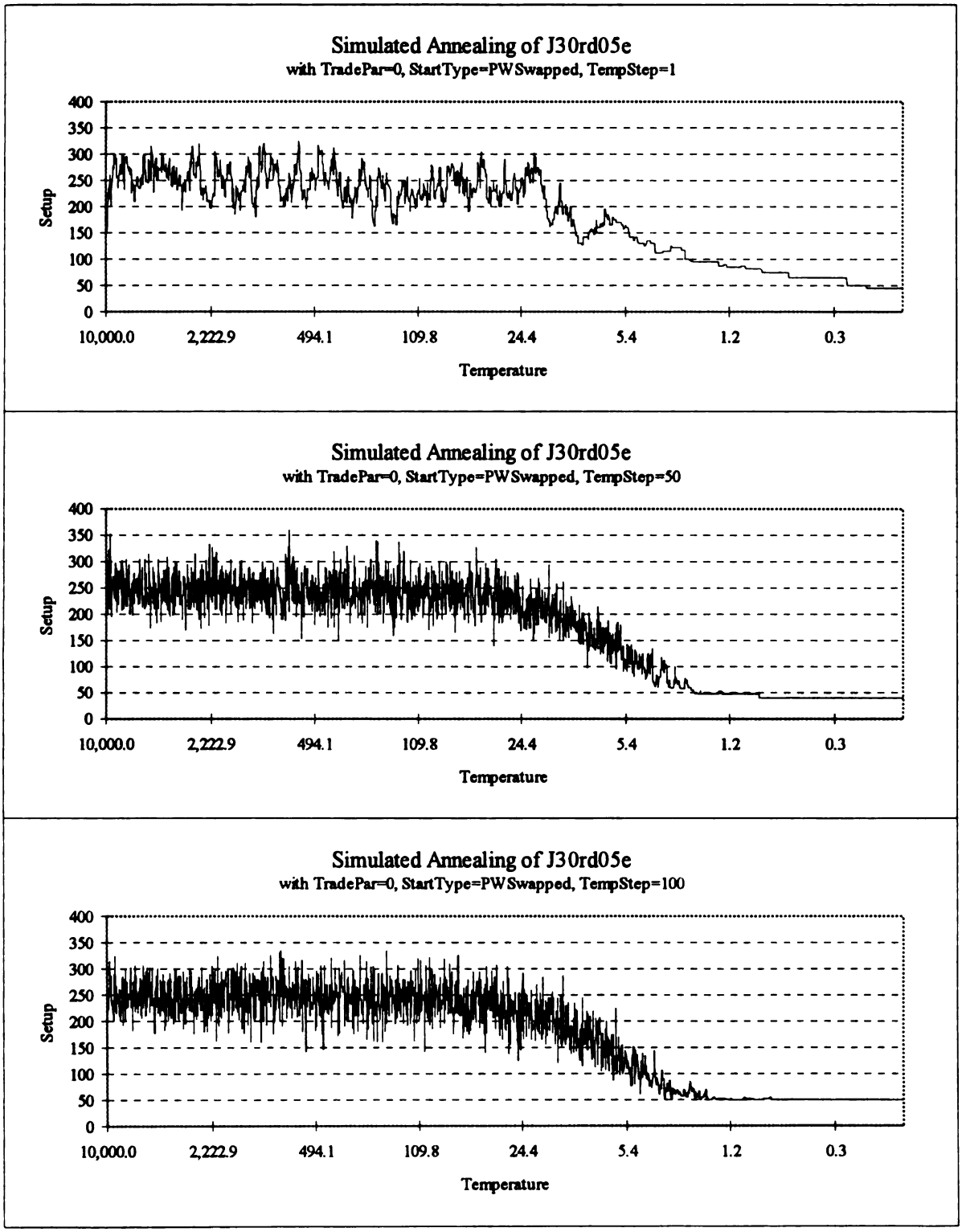


Figure 9: Objective Values as Temperature Decreased (30-job, $\theta = 0$, *PWS*)

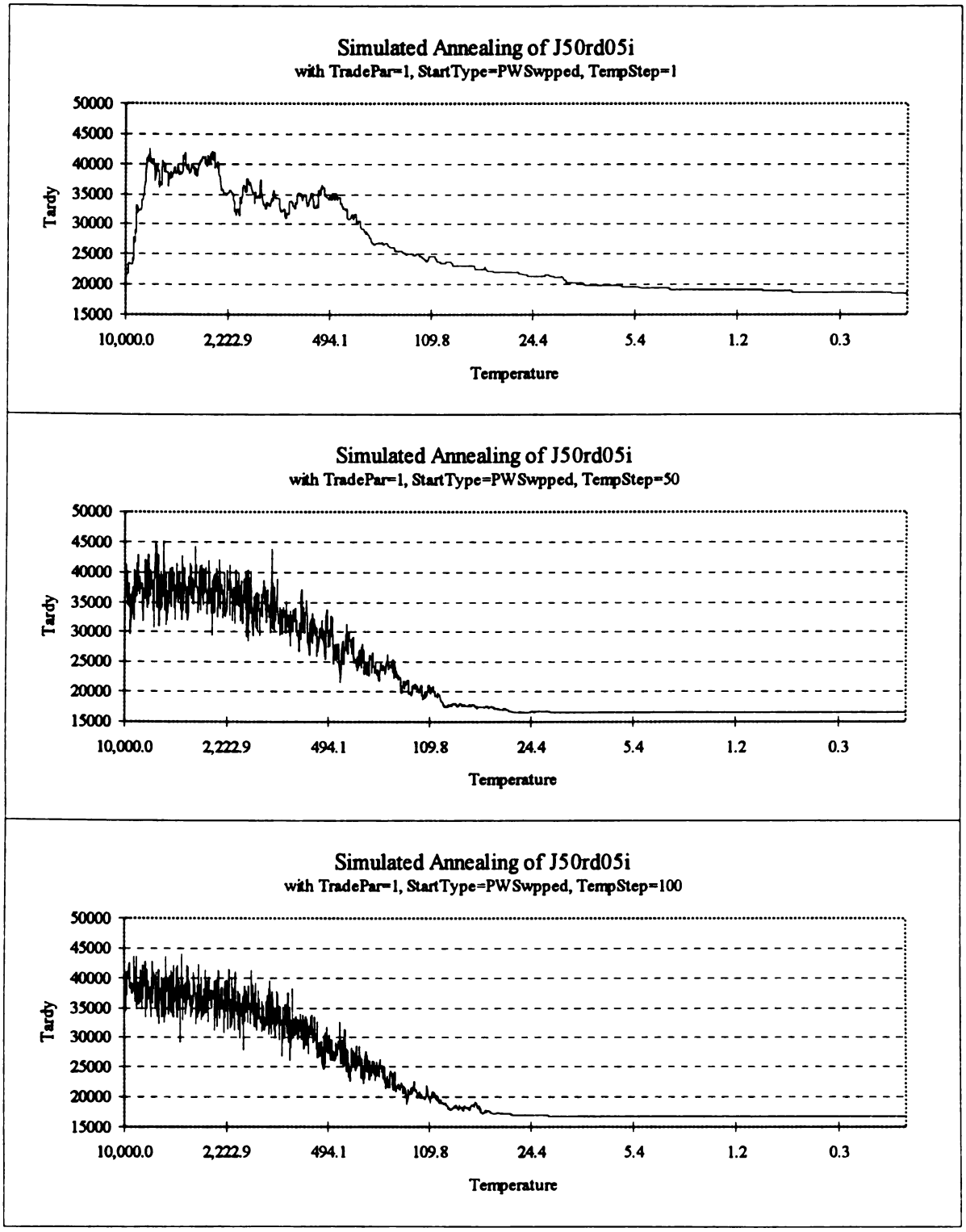


Figure 10: Objective Values as Temperature Decreased (50-job, $\theta = 1$, *RND*)

5.6 Cases With Worse Final Solution Than The Initial Solution

Table 17 shows the problems where the final solutions at the end of the annealing process are actually worse than the pairwise swapped initial solutions. However, none of these initial or final solutions are the best known solutions. *BestSol*, *InitSol*, and *FinalSol* are the best known, initial and final solutions respectively. The best known solutions for the 10-job problems are the optimum solutions. *Diff* is the difference between *InitSol* and *FinalSol*, and *% Diff* is expressed as a percentage of the initial solution. *n*, *RD*, θ, *INIT*, *STEP* are the experimental factors.

Table 17: Problems With Worse Final Solution Than Initial Solution

Jobno	*n*	*RD*	θ	*INIT*	*STEP*	BestSol	InitSol	FinalSol	Diff	% Diff
J10RD02H	10	0.2	0	PWS	1	11.0	32.0	36.0	-4.0	12.5
J10RD05G	10	0.5	0.5	PWS	1	787.0	844.0	854.0	-10.0	1.2
J30RD02B	30	0.2	0	PWS	1	36.0	74.0	91.0	-17.0	23.0
J30RD02F	30	0.2	0	PWS	1	33.0	75.0	86.0	-11.0	14.7
J30RD02G	30	0.2	0	PWS	1	26.0	80.0	94.0	-14.0	17.5
J30RD05C	30	0.5	0	PWS	1	41.0	80.0	84.0	-4.0	5.0
J30RD05H	30	0.5	0	PWS	1	38.0	94.0	95.0	-1.0	1.1
J50RD02A	50	0.2	0	PWS	1	60.0	115.0	123.0	-8.0	7.0
J50RD02D	50	0.2	0	PWS	1	52.0	100.0	151.0	-51.0	51.0
J50RD02G	50	0.2	0	PWS	1	49.0	133.0	147.0	-14.0	10.5
J50RD02H	50	0.2	0	PWS	1	61.0	126.0	132.0	-6.0	4.8
J50RD02I	50	0.2	0	PWS	1	51.0	132.0	136.0	-4.0	3.0
J50RD02J	50	0.2	0	PWS	1	56.0	117.0	128.0	-11.0	9.4
J50RD02F	50	0.2	0.5	PWS	1	12,080.0	13,267.5	13,302.5	-35.0	0.3
J50RD05B	50	0.5	0	PWS	1	60.0	133.0	140.0	-7.0	5.3
J50RD05C	50	0.5	0	PWS	1	65.0	120.0	153.0	-33.0	27.5
J50RD05J	50	0.5	0	PWS	1	48.0	103.0	112.0	-9.0	8.7
J50RD05A	50	0.5	0.5	PWS	1	9,647.5	10,986.0	11,140.5	-154.5	1.4
J50RD05G	50	0.5	0.5	PWS	1	5,794.5	7,556.0	7,707.5	-151.5	2.0
J50RD05A	50	0.5	1.0	PWS	1	19,446.0	22,016.0	22,323.0	-307.0	1.4

There are a total of 20 problems where the final solutions are worse than the initial solutions, or 1.85% from a total of 1,080 problems. Thirteen of the 20 problems are 50-job problems ($n = 50$), and 15 problems (75%) are setup problems ($\theta = 0$). Approximately equal number of problems are from $RD = 0.2$ and 0.5. All the 20 problems involved pairwise swapped initial solution ($INIT = PWS$), and one update between changes in temperature range ($STEP = 1$). Table 17 indicates that the deterioration of the objective values range from 0.3% to 51.0%. A closer analysis of the output file reveals that, only three of the 20 problems actually sampled a better sequence (lower objective value) during the annealing process, but were unable to return to that solution at the end of the annealing process. The lowest objective values sampled by the first, the second, and the eighth problems are 25, 802.5 and 111 respectively.

At this point, it should not be concluded that good final solutions can be generated by simply swapping jobs pairwise, and simulated annealing is ineffective. However, the results probably show that simulated annealing may have more difficulty in getting out of local optima when $STEP = 1$. Charts 1 to 7 of Appendix 1 show the objective values of these 20 problems plotted against the temperature. A general conclusion from these charts is that the objective values increased steadily at high temperature (initially). This suggests that the simulated annealing algorithm had no trouble escaping the initial solution neighborhood. However, it could not get back to the initial solution, or a better solution at the end of the simulated annealing process. Practically, this is not a problem because anyone using a simulated annealing approach to scheduling would record the best incumbent. In the event of a poor final solution, the best incumbent can be used.

5.7 Analysis of Variance - *AVGFIN*

Standard Analysis of Variance (ANOVA) procedure was used to test the hypotheses at $\alpha = 0.05$. All statistical analysis was conducted using SPSS for Windows version 6.0 on an Intel 486-25 MHz personal computer. The performance measure for testing the two experimental factors related to simulated annealing (*INIT* and *STEP*) is the average objective value per job, *avgfin*. A general full factorial analysis of variance (using *unique* sums of squares) was conducted with all the five experimental factors.

Table 18 shows the ANOVA for *avgfin*. The model is statistically significant with a very high R^2 value of 0.935. Four of the five experimental factors are statistically significant at the main effect level, except *INIT*. Three 2-factor interaction and a 3-factor interaction are statistically significant. However, the ANOVA model failed both the Cochrans C and the Bartlett-Box F tests, indicating that it has violated the homogeneity of variance assumption. The variances (Figure 11) and the standard deviations (Figure 12) of *avgfin* are plotted against the cell means to investigate the possible reason of the violation. Figures 11 to 13 clearly indicate that the variances increased with the magnitude of the cell means, which is a common cause of constant variance violation. Under such circumstances, a logarithmic transformation would normally stabilize the variance. However, Box-Cox transformation was used to determine the proper transformation to stabilize the variance. The normal Q-Q plot of the residuals of *avgfin* (Figure 14) indicates that the model has over-estimated for smaller *avgfin* and under-estimated for larger *avgfin*.

Table 18: Full Factorial ANOVA - *avgfin*

Source of Variation	SS	DF	MS	F	Sig of F
WITHIN+RESIDUAL	1305394.72	972	1343.00		
RD	198717.75	1	198717.75	147.97	.000 *
INIT	66.64	1	66.64	.05	.824
n	3599523.68	2	1799761.80	1340.11	.000 *
θ	12413022.67	2	6206511.30	4621.38	.000 *
STEP	19065.79	2	9532.89	7.10	.001 *
RD × INIT	81.88	1	81.88	.06	.805
RD × *n*	34532.00	2	17266.00	12.86	.000 *
RD × θ	136505.28	2	68252.64	50.82	.000 *
RD × STEP	208.09	2	104.05	.08	.925
INIT × *n*	4.38	2	2.19	.00	.998
INIT × θ	60.39	2	30.20	.02	.978
INIT × STEP	21.65	2	10.82	.01	.992
n × θ	2419973.46	4	604993.36	450.48	.000 *
n × STEP	10271.55	4	2567.89	1.91	.106
θ × STEP	9246.70	4	2311.68	1.72	.143
RD × INIT × *n*	.16	2	.08	.00	1.000
RD × INIT × θ	149.31	2	74.65	.06	.946
RD × INIT × STEP	52.83	2	26.42	.02	.981
RD × *n* × θ	22303.49	4	5575.87	4.15	.002 *
RD × *n* × STEP	497.76	4	124.44	.09	.985
RD × θ × STEP	120.20	4	30.05	.02	.999
INIT × *n* × θ	5.51	4	1.38	.00	1.000
INIT × *n* × STEP	6.00	4	1.50	.00	1.000
INIT × θ × STEP	23.94	4	5.98	.00	1.000
n × θ × STEP	6071.38	8	758.92	.57	.807
RD × INIT × *n* × θ	40.98	4	10.24	.01	1.000
RD ×INIT ×*n* ×STEP	22.44	4	5.61	.00	1.000
RD ×INIT ×θ ×STEP	77.69	4	19.42	.01	1.000
RD × *n* × θ × STEP	607.58	8	75.95	.06	1.000
INIT ×*n* ×θ × STEP	58.39	8	7.30	.01	1.000
RD×INIT×*n*×θ×STEP	113.33	8	14.17	.01	1.000
(Model)	18871452.92	107	176368.72	131.32	.000 *
(Total)	20176847.65	1079	18699.58		
R-Squared =	0.935				
Adj. R-Squared =	0.928				

Univariate Homogeneity of Variance Tests
Cochrans C (9, 108) = .08133, p = .000 *(approx.)
Bartlett-Box F(107, 78010) = 30.43430, p = .000 *

* significance at $\alpha = 0.05$

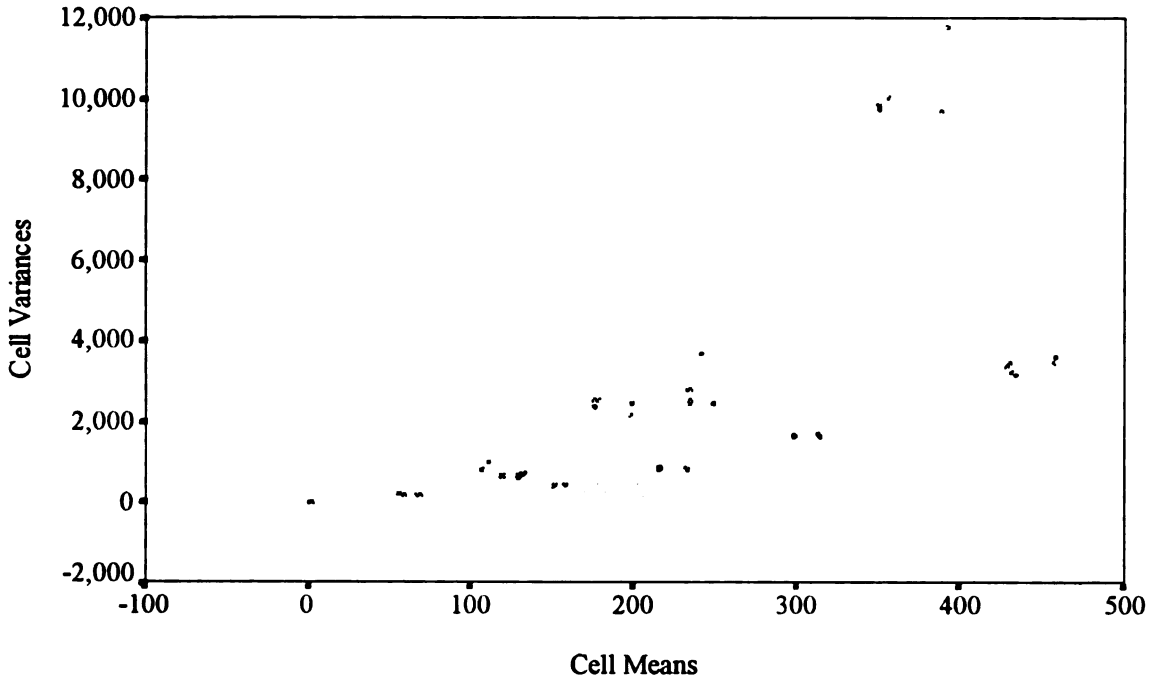


Figure 11: Mean versus Variance for *avgfin*

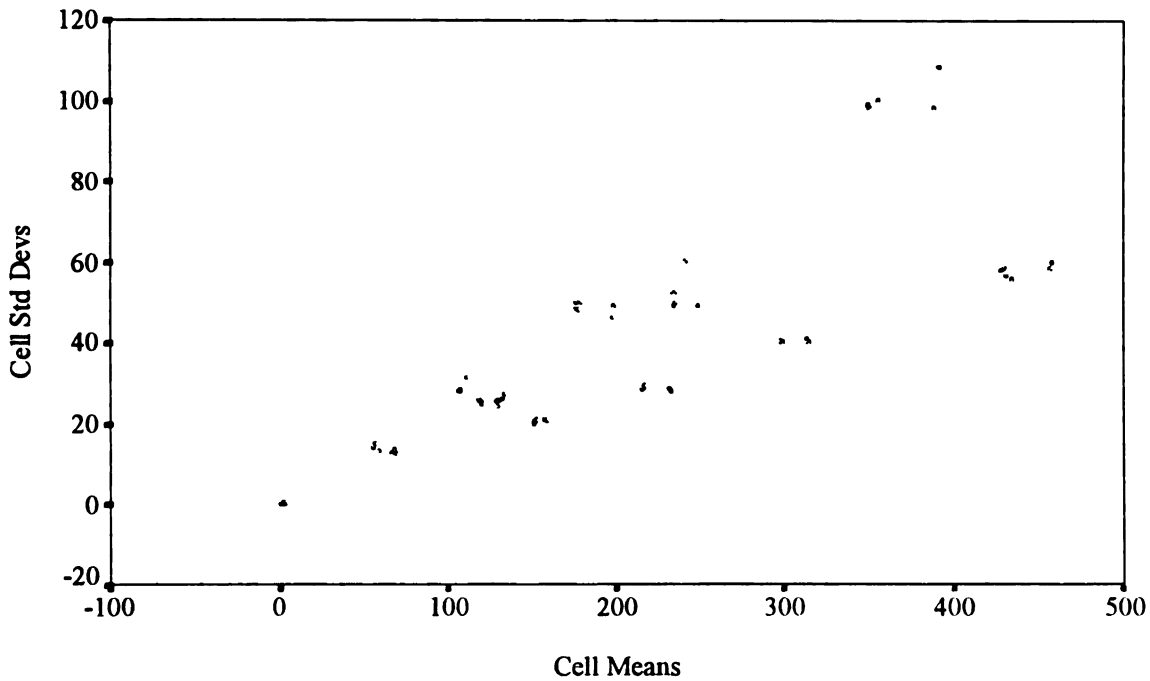


Figure 12: Mean versus Standard Deviation for *avgfin*

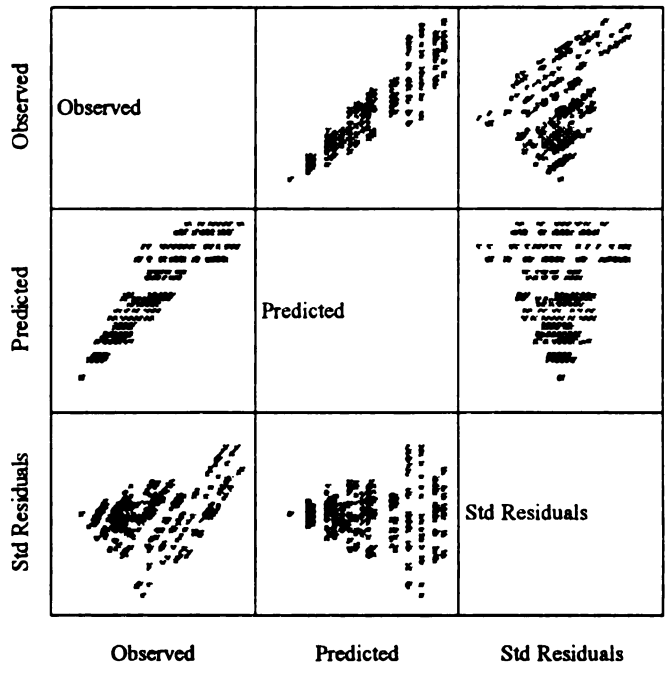


Figure 13: Residual Plots - *avgfin*

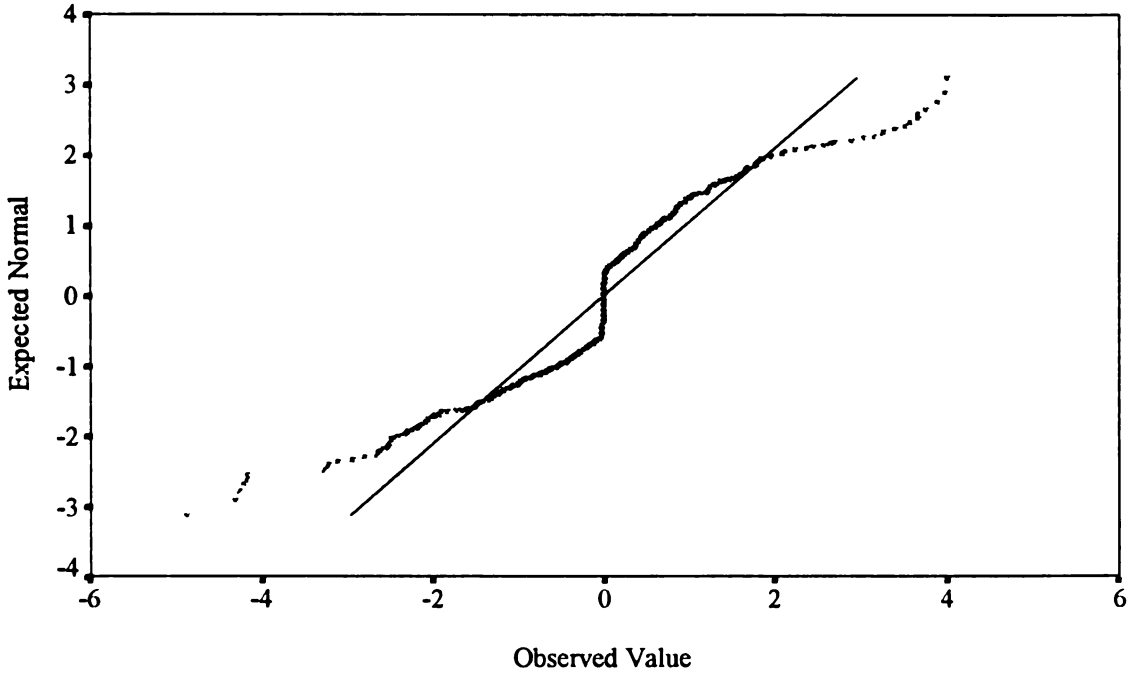


Figure 14: Normal Q-Q Plot of Residuals of *avgfin*

5.8 Box-Cox Transformations

Box and Cox (1964) developed a procedure for choosing a transformation from the family of power transformations on the dependent variable (Y = *avgfin*). This procedure is useful for correcting skewness of the distributions of the residuals, unequal variances, and non-linearity of the regression function. The family of power transformations can be expressed as $Y' = Y^{\lambda}$ where λ is a parameter to be determined from the data. The criterion for determining the appropriate λ of the transformation of Y is to find the value of λ that minimizes the sum squares error (SSE) based on the transformation. However, the power of the transformation affects the magnitude of the SSE term. Therefore the geometric mean of Y was used as the standardized variable to neutralize the effect so that SSE is unaffected by the value of λ (Neter *et. al*, 1989, pp 149-150). The transformation procedure can be expressed as:

$$W = K_1(Y^{\lambda} - 1) \quad \text{if } \lambda \neq 0$$

$$= K_2(\ln Y) \quad \text{if } \lambda = 0$$

where: $K_1 = \dfrac{1}{\lambda K_2^{\lambda-1}}$

$$K_2 = \left(\prod_{i=1}^{n} Y_i\right)^{\frac{1}{n}} = \text{geometric mean of } \textit{avgfin} = 36.55629$$

Table 19 contains the Box-Cox transformation results for *avgfin* with selected values of λ ranging from -4 to + 4. The Box-Cox procedure suggests a power value of 0 is most appropriate for *avgfin*. It results in the lowest SSE of 61,823.61. For $\lambda = 0$, the

proper transformation is W = 36.55629 * ln(*avgfin*). It should be reminded that the geometric mean of 36.55629 was used to neutralize the effect of the power transformation so that SSE could be compared on the same magnitude independent of the value of λ. Natural logarithm, ln, was used for the transformation. The R^2 and the adjusted R^2 of Table 19 were included for comparison purpose only. They have no significance in the selection of the appropriate value of λ. A new ANOVA model was fitted on the transformed dependent variable, *lnavgfin* = ln(*avgfin*), and presents in *Section 5.9: Analysis of Variance - LNAVGFIN.*

Table 19: Box-Cox Results for *avgfin*

λ	SSE	R^2	Adj R^2
-4	1.0810 E+16	0.390	0.323
-3	7.8897 E+12	0.597	0.552
-2	7,184,911,117	0.804	0.782
-1	9,002,598.87	0.946	0.941
-0.5	385,930.77	0.983	0.981
0	**61,823.61**	**0.991**	**0.990**
0.5	203,651.26	0.971	0.968
1	1,305,394.72	0.935	0.928
2	98,602,470.48	0.858	0.842
3	10,748,813,925	0.780	0.756
4	1.3764 E+12	0.706	0.674

5.9 Data Exploration - *lnavgfin*

A brief examination of the transformed dependent variable, *lnavgfin*, is presented in this section. Descriptive statistics are presented in Table 20. The mean is 3.599 and the most frequently occurring value of *lnavgfin* (mode) is 0.336. The median, which is the average of the 540th and the 541st observation is 4.705. The large difference between the mode and the other two central tendency measures indicate that the distribution of *lnavgfin* is not symmetric, that is, the distribution is skewed. The *Skewness* measure of -0.556 indicates that the distribution of *lnavgfin* is negatively skewed, or the 'tail' (more cases) is toward smaller values. The *Kurtosis* measure of -1.437 indicates that observations cluster around the central point less than those in the normal distribution (i.e. it is flatter or platykurtic). The standard deviation is 2.197.

Table 20: Descriptive Statistics - *lnavgfin*

Mean	3.599	Mode	0.336	Median	4.705
Std error	0.067	Std dev	2.197	Variance	4.826
Range	6.675	Minimum	-0.357	Maximum	6.319
Kurtosis	-1.437	S E Kurt	0.149		
Skewness	-0.556	S E Skew	0.074		

A *Boxplot* displays summary statistics for the distribution, and it can further summarize and confirm the information in Tables 20. It plots the median (horizontal line inside the box), the 25th (lower boundary of the box) and the 75th (upper boundary of the box) percentiles, and values that are far removed from the rest. These 25th and the 75th percentiles are sometimes called the Tukey's hinges (Norusis 1989a, pp 185). The range between these two inter-quartiles is called the box-length. Figures 10 to 14 show the *Boxplots* for *lnavgfin* versus the experimental factors, *RD*, *INIT*, *n*, θ, and *STEP* respectively. The two horizontal lines drawn from the ends of the box represent the largest and the smallest observed values that are not outliers. Cases with values that are more than 3 box-lengths from the upper or lower edge of the box are known as *extreme values* (indicated as * in the *Boxplot*), and those that are between 1.5 to 3 box-lengths are called *outliers* (indicates as *o*). Figures 15 to 19 clearly indicate that *lnavgfin* has no extreme value or outlier. However, all the medians (except Figure 18) are very close to the 75th percentile, confirming the negative skewness measure that more cases skewed toward smaller values.

Figures 15 and 16 show the *Boxplots* of *lnavgfin* versus *RD* and *INIT* respectively. Both the figures show a very wide and similar dispersion (range) of *lnavgfin* over the two levels of the experimental factors. Based on the similarity of the dispersions and the medians of these two plots, *RD* and *INIT* do not seem to have an effect on *lnavgfin*. Figure 17 shows that *lnavgfin* is not as dispersed and the median of *lnavgfin* is smaller when n = 10 (level 1). The medians and the dispersions increase as n increased. Figure 18 clearly indicates that *lnavgfin* is smallest with θ = 0 (level 1) when compared to θ = 0.5 or

1.0 (levels 2 or 3). This is because when $\theta = 0$, it is a pure setup problem where the objective function value is much smaller than a combined setup and tardiness or a pure tardiness problem. The dispersion is small compared to the other plots. Figure 19 indicates that the dispersion of *lnavgfin* is slightly smaller with *STEP* = 1 when compared to *STEP* = 50 or 100 (levels 2 or 3). It should be noted that these analysis and comparisons are made without considering any interaction effects among the experimental factors. The two experimental factors of interested are *INIT* and *STEP*.

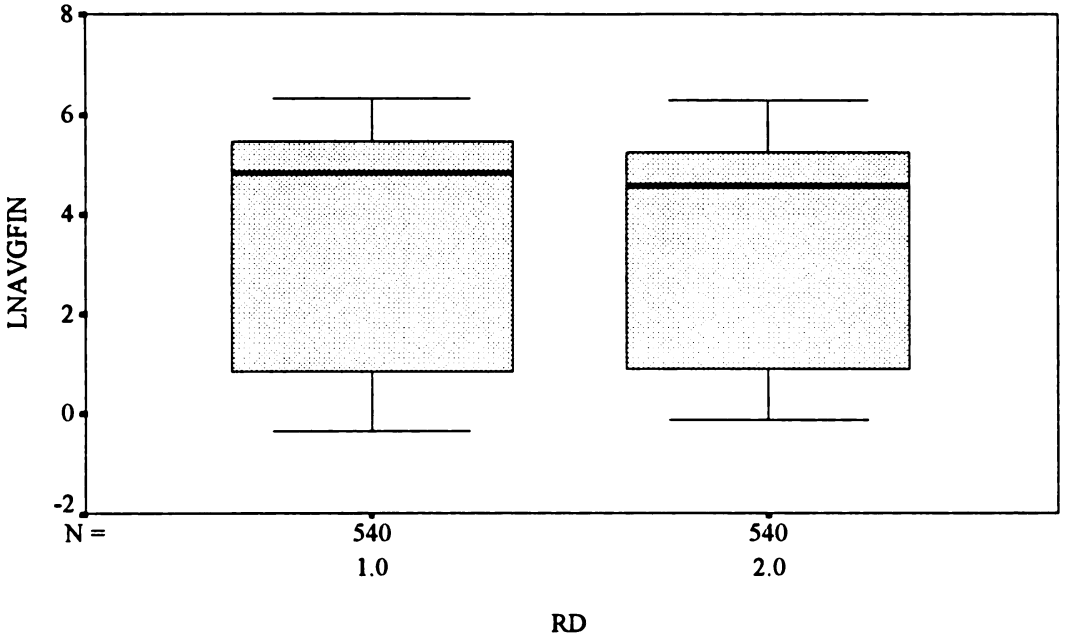


Figure 15: Boxplots of *lnavgfin* versus *RD*

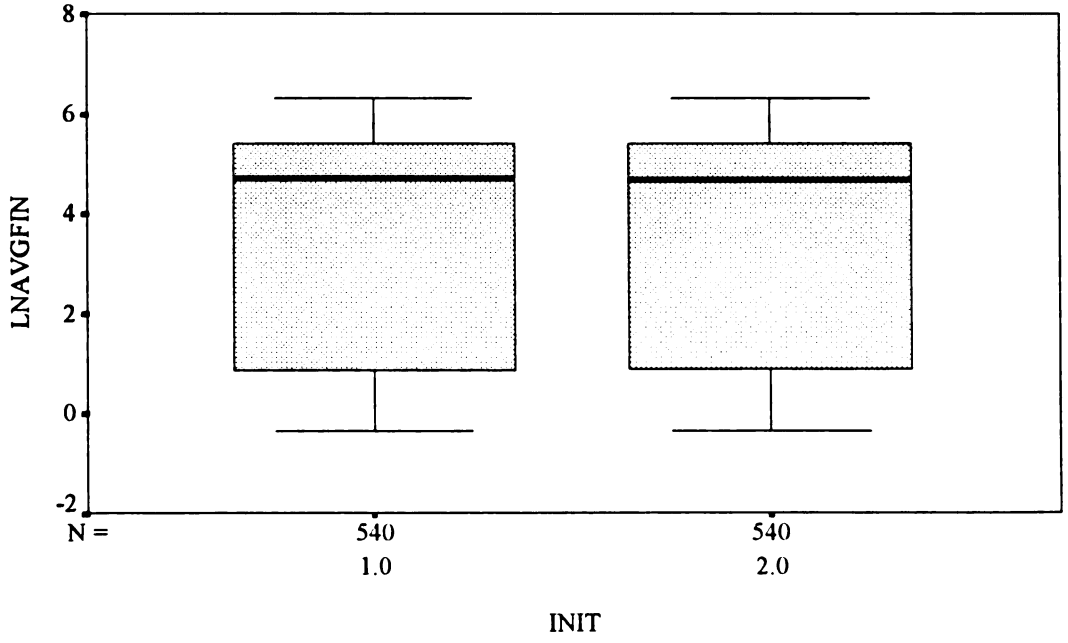


Figure 16: Boxplots of *lnavgfin* versus *INIT*

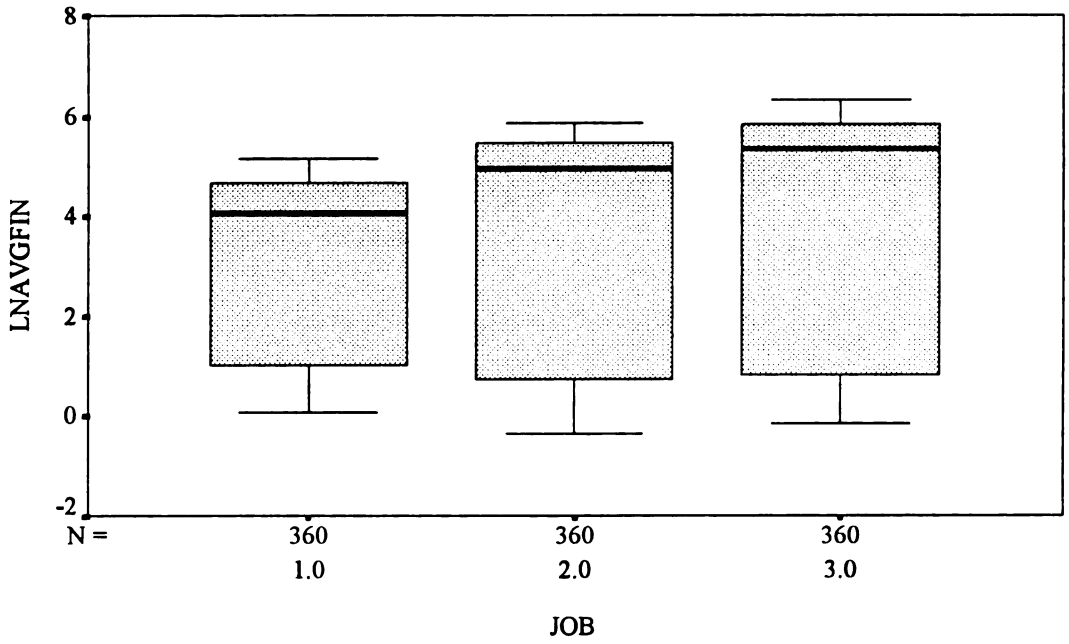


Figure 17: Boxplots of *lnavgfin* versus *n*

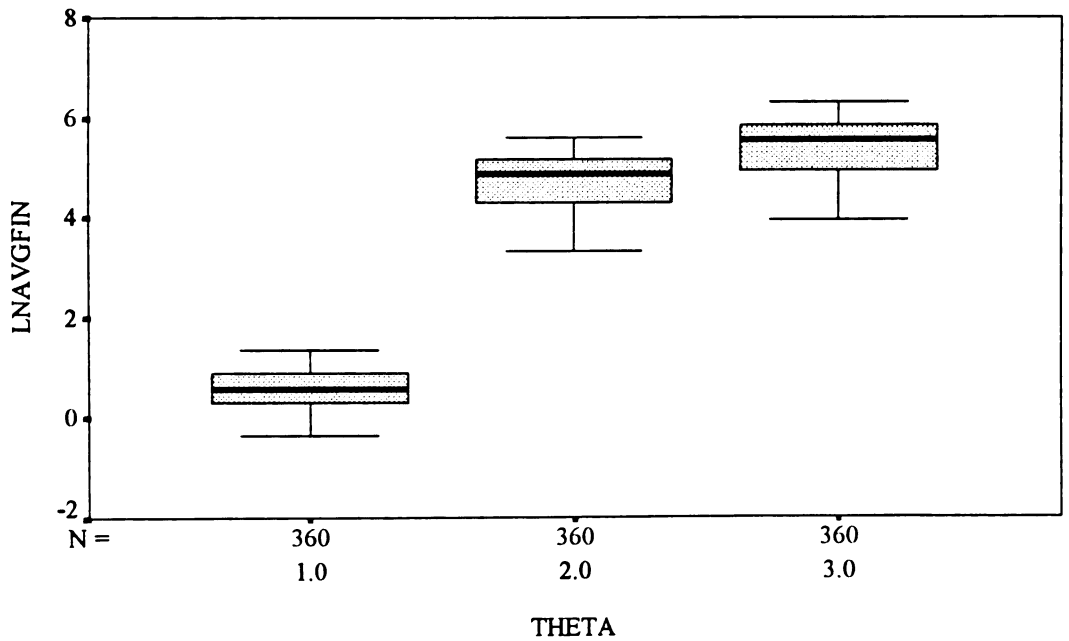


Figure 18: Boxplots of *lnavgfin* versus θ

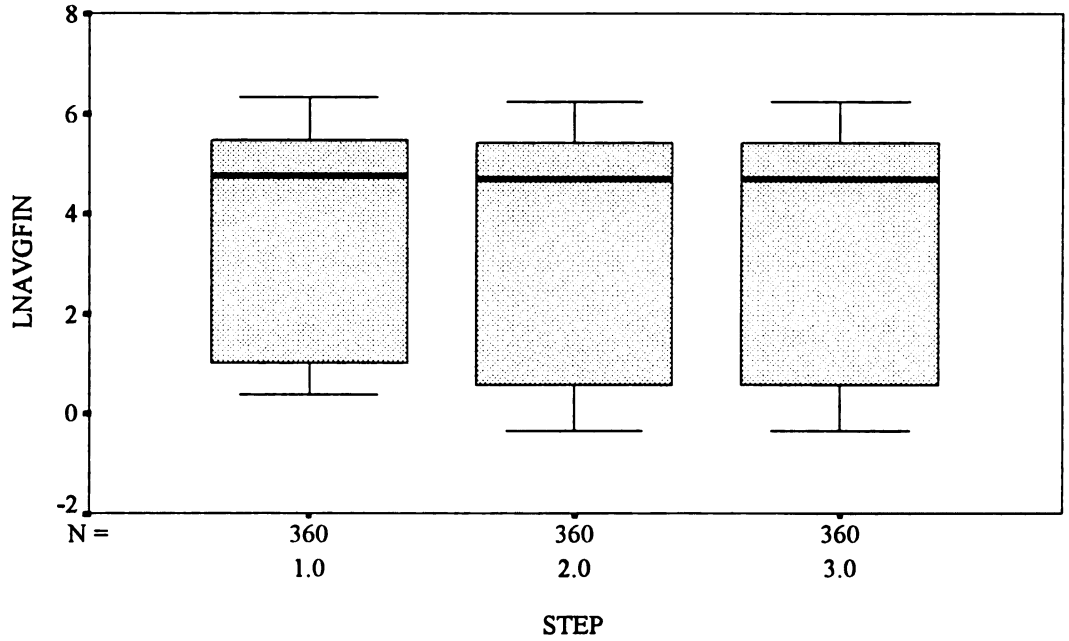


Figure 19: Boxplots of *lnavgfin* versus *STEP*

5.10 Analysis of Variance - *lnavgfin*

A general full factorial analysis of variance (using *unique* sums of squares) was fitted onto the new transformed dependent variable, *lnavgfin* with all the five experimental factors. Since the two experimental factors of interest are *INIT* and *STEP*, the factors *RD*, θ, and *n* can be combined into one factor. However, the three factors *RD*, θ, and *n* were separated to check for interaction among the three factors themselves (both analyses produced similar results - APPENDIX 2). The results are presented in Table 22. The model fitted the data very well, with a significant F-statistic of 0.000 and an R^2 of 0.991. Four main factor effects, 5 two-factor interactions and a three-factor interaction effects are significant at $\alpha = 0.05$. In an ANOVA, the residuals are assumed to be independently and normally distributed random variables with mean zero and variance σ^2. The variance σ^2 is assumed constant for all the experimental cells.

5.10.1. Model Adequacy Checking

Wald-Wolfowitz Runs test was used to check if the residuals are randomly distributed. Table 21 shows the runs test of both the raw and standardized residuals. The runs were tested against the true mean of 0.0000, which was calculated from the residuals. The zero mean conformed to the assumption that the residuals are distributed with a mean of zero. Two-tailed test statistics indicate that the hypotheses of randomness could not be

rejected at $\alpha = 0.05$. Therefore, there is no reason to suspect that the residuals are not independently distributed.

Table 21: Runs Test of Residuals

		Raw Residuals	
Runs:	559	Test Value =	.0000 (Mean)
Cases:	457	LT Mean	
	623	GE Mean	Z = 1.9181
	1080	Total	2-Tailed P = .0551
		Standardized Residuals	
Runs:	559	Test Value =	.0000 (Mean)
	457	LT Mean	
	623	GE Mean	Z = 1.9181
	1080	Total	2-Tailed P = .0551

The next assumption tested is the homogeneity of variance. Table 22 shows that the Bartlett-Box F test rejected the null hypothesis that all population cell variances are equal, but the Cochrans C test did not reject the null hypothesis, which indicates that there is no reason to suspect that the variances are not equal. Unfortunately, both of these tests are sensitive to departures from normality. In order to further investigate this assumption, the cell means were plotted against the cells variances (Figure 20) and the standard deviations (Figure 21). The observed, predicted and standardized residuals were also plotted against each others (Figure 22). Visual inspection of these figures do not indicate any serious violation of the homogeneity of variance assumption, except that there is a gap

in the observations between cell means 1 to 4. This is because there is no observation in this range. Fortunately, the F test is only slightly affected in the balanced, fixed-effect model if the constant variance assumption is violated (Montgomery, 1984, pp 91). A fixed effect model is defined as a design in which the factors and levels under investigation are the only ones that the researchers are interested in drawing conclusions about.

Figure 23 shows the normal probability plot of the residuals where each observed value was paired with its expected value from the normal distribution. The points clustered very closely around the straight line, indicating that the residuals are reasonably normally distributed. The histograms of the raw residuals (Figure 24) and the standardized residuals (Figure 25) conformed quite closely to the normal curves superimposed on the charts, confirming that there is no reason to suspect that the residuals are not normally distributed. However, the Kolmogorov-Smirnov test rejected the normality assumption. Whenever the sample size is large, almost any goodness-of-fit test will reject the null hypothesis. It is almost impossible to find research data that are exactly normally distributed. In addition, moderate departures from normality are of little concern in the fixed effects analysis of variance. It usually causes both the true significance level and power to differ slightly from the computed values, with the power generally being lower (Montgomery, 1984, pp 87). For analysis of variance (and indeed most statistical tests), it is adequate that the data are approximately normally distributed. In summary, the diagnostic checks on the assumptions did not reveal any major departures.

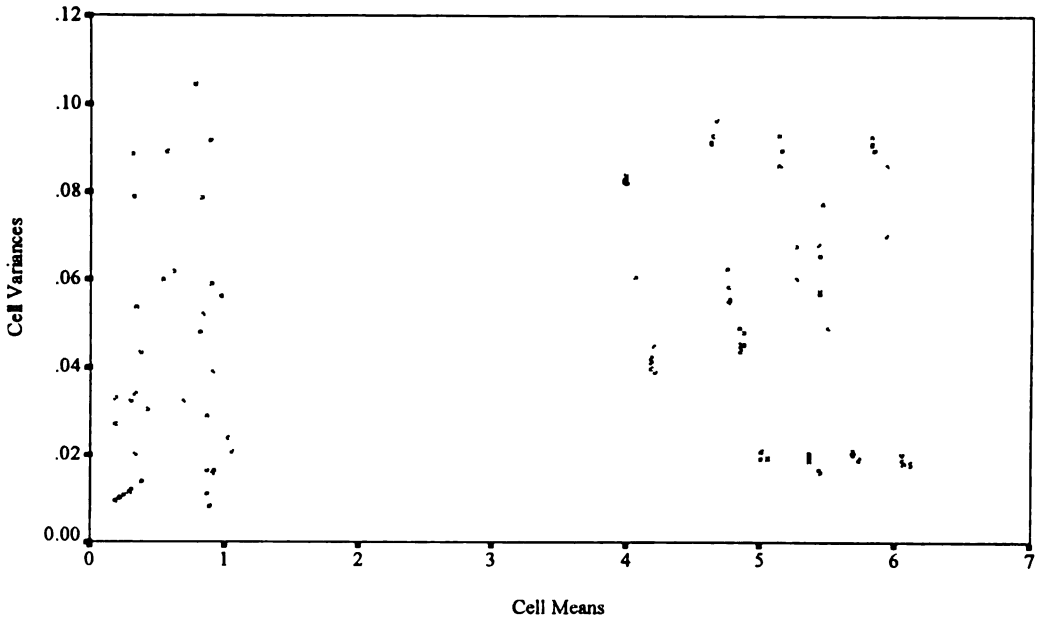


Figure 20: Mean versus Variance for *lnavgfin*

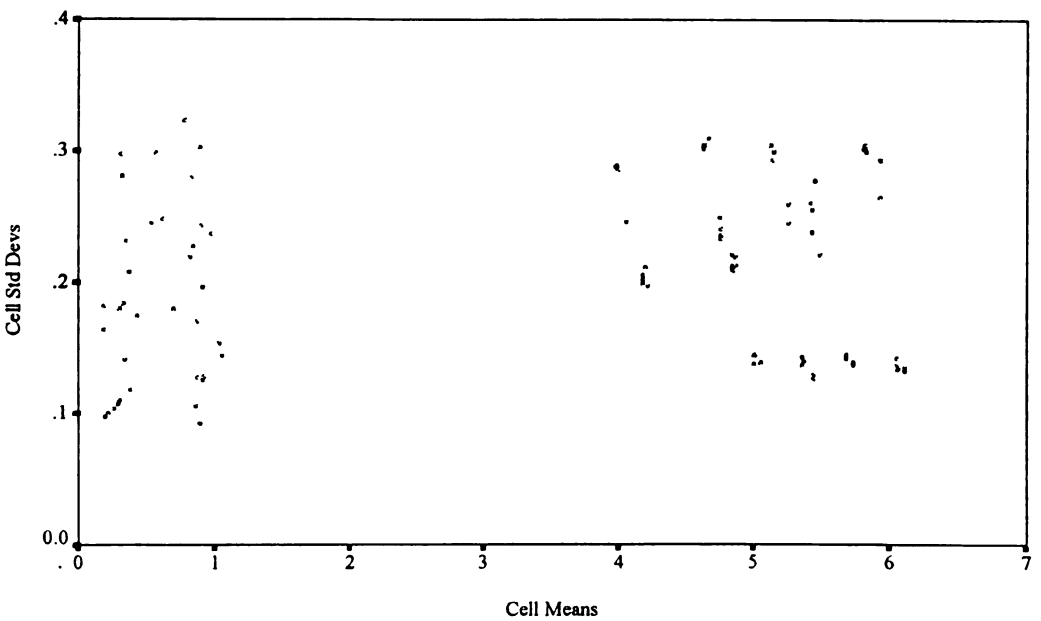


Figure 21: Mean versus Standard Deviation for *lnavgfin*

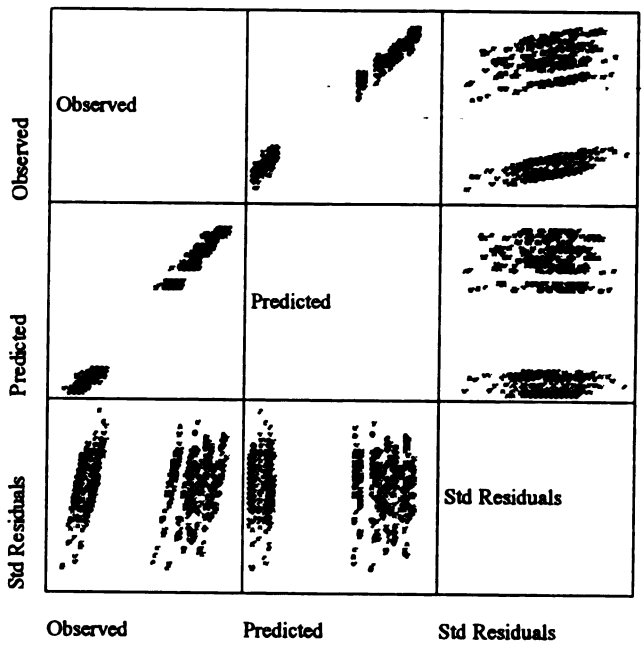


Figure 22: Residual Plots - *lnavgfin*

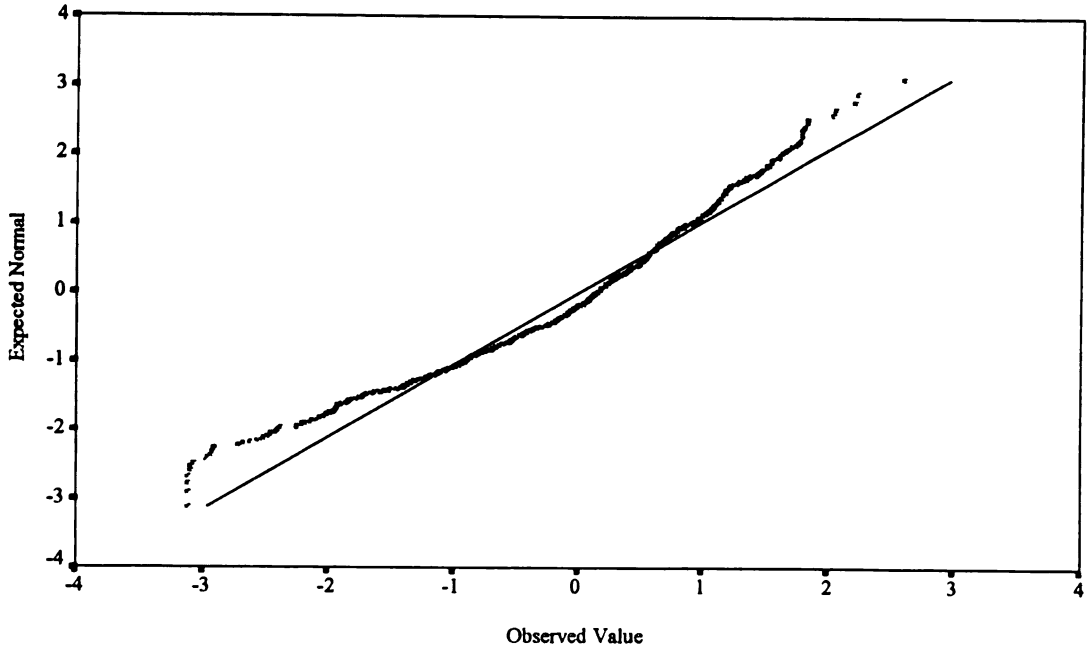


Figure 23: Normal Q-Q Plot of Residuals of *lnavgfin*

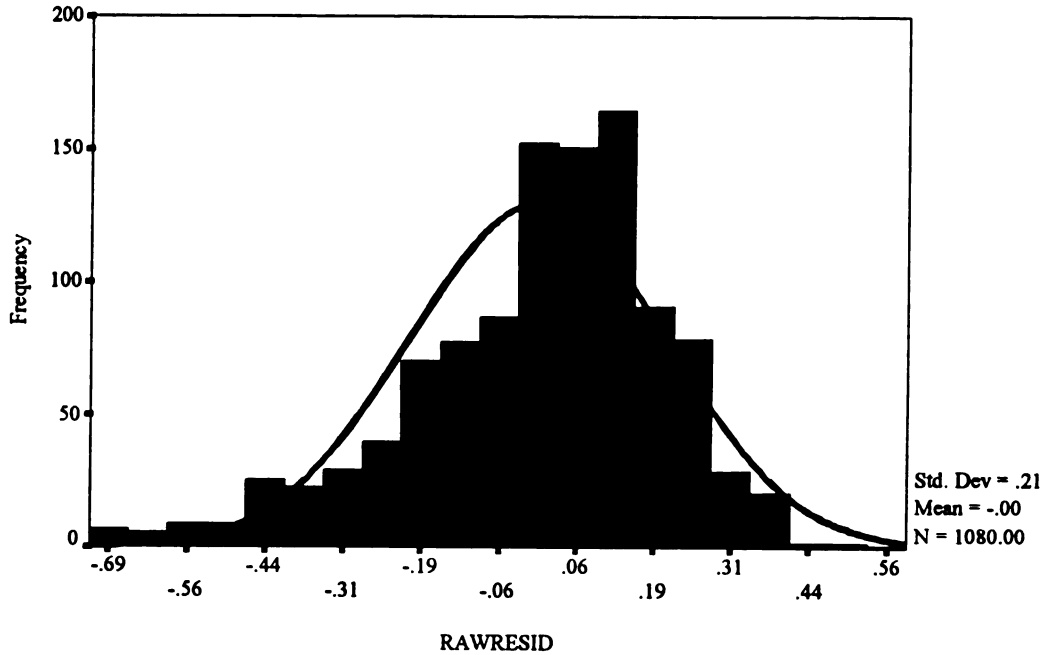


Figure 24: Histogram of the Raw Residuals of *lnavgfin*

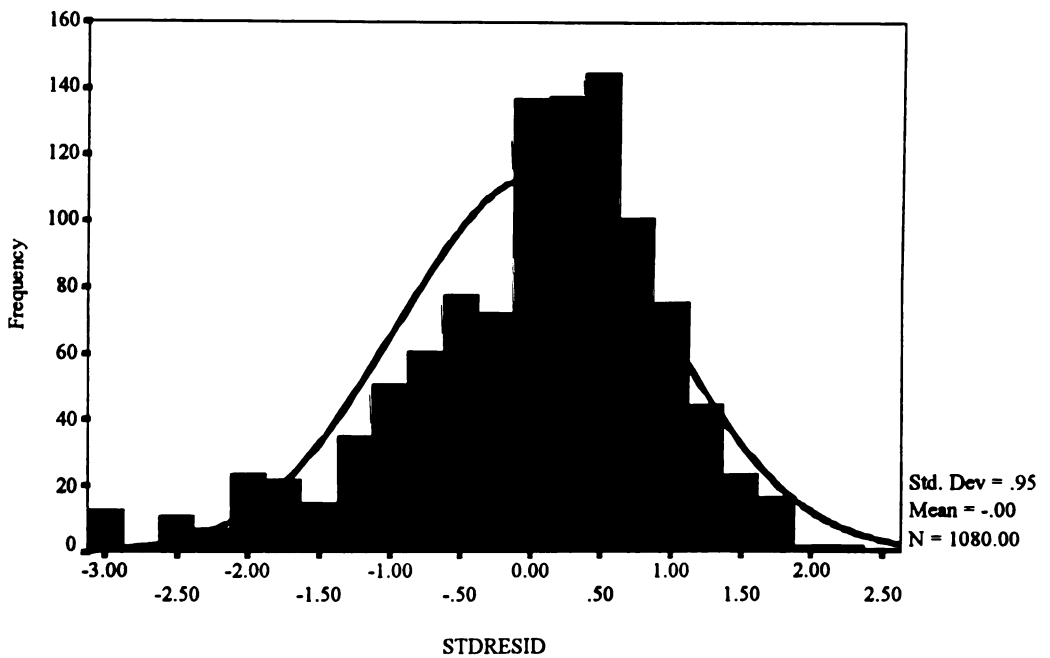


Figure 25: Histogram of the Standardized Residuals of *lnavgfin*

5.10.2. The Results of the Full Factorial ANOVA

In a full factorial analysis of variance, the *within* source of variation for the error term is the same as the *within+residual* source of variation because all the possible main and interaction terms are tested (therefore, *residual* error term is zero). The ANOVA in Table 22 shows that there are five main effect terms, of which four are highly significant. Five of the two-term interactions, and one three-term interaction are significant as well. None of the four-term nor the five-term interaction is significant. The F-statistic shows that the model is significant with an R^2 of 0.991 (adjusted $R^2 = 0.990$).

The row labeled *INIT* tested whether the two different means of generating the initial solution were equally effective in solving the sequencing problem. It was the only experimental factor that has no interaction effect. The other four factors (including *STEP*) have significance interaction terms. Therefore, it made no sense trying to conclude that, overall, the main effect of *STEP* on *lnavgfin*. Simultaneous confidence interval of the parameters are not presented due to these problems. Where interaction is significant, comparisons between the means of one factor may be obscured by the interaction. The approach to this situation in this research is to fix *STEP* at a specific level (e.g. $n = 1$, 2 or 3), and apply the Duncan's multiple range test to the means of *STEP*. However, from Table 22, it is able to conclude that the types of initial solution (*INIT*) has no effect on *lnavgfin* due to the lack of interaction effect involving *INIT*. The effect size measures and the observed power of the ANOVA are presented in the next section.

5.10.3. Effect Size and Power of the Tests

Table 23 presents the effect size measures and the observed power of the analysis of variance at $\alpha = 0.05$. Partial eta-squared statistics (effect-size measure) describe the proportion of the total variability explained by the grouping or factor variable. A value close to one indicates that all of the total variability is attributable to differences between the groups, while a value close to zero indicates that the grouping variable explains little of the total variability. Table 23 shows that *RD* explains very little of the total variability. Only two (i.e. *RD* * $\theta = 0.688$, and θ * *STEP* = 0.168) of the five significant two-factor interaction terms have partial eta-squared value of more than 0.10. In summary, only five grouping variables (i.e. *n*, θ, *STEP*, *n* * θ, and θ * *STEP*) explained more than 0.10 of the variability in its respective groups.

The observed power is the ability to reject the null hypothesis when it is false, assuming $\alpha = 0.05$. In general, power depends on the magnitude of the true differences between the means and the sample sizes used in the experiment. If the true differences are very small, large sample sizes are needed to be able to correctly reject the null hypotheses. Therefore, it is very useful and important to evaluate the power of the test after the experiment has been completed. Table 23 clearly indicates that the sample size of 10 selected for each experimental cell is large enough to detect a true difference between cell means as the power of those significance terms are at least 0.794 and above.

Table 22: Full Factorial ANOVA - *lnavgfin*

Source of Variation	SS	DF	MS	F	Sig of F
WITHIN+RESIDUAL	46.26	972	.05		
RD	4.38	1	4.38	92.04	.000 *
INIT	.00	1	.00	.00	.950
n	88.19	2	44.10	926.46	.000 *
θ	4939.28	2	2469.64	51888.31	.000 *
STEP	9.41	2	4.70	98.81	.000 *
RD × INIT	.04	1	.04	.78	.379
RD × *n*	.45	2	.23	4.77	.009 *
RD × θ	5.01	2	2.50	52.63	.000 *
RD × STEP	.03	2	.01	.29	.746
INIT × *n*	.02	2	.01	.20	.821
INIT × θ	.02	2	.01	.17	.845
INIT × STEP	.01	2	.00	.10	.906
n × θ	102.17	4	25.54	536.68	.000 *
n × STEP	1.18	4	.30	6.21	.000 *
θ × STEP	9.33	4	2.33	48.99	.000 *
RD × INIT × *n*	.00	2	.00	.03	.975
RD × INIT × θ	.02	2	.01	.25	.777
RD × INIT × STEP	.05	2	.03	.57	.568
RD × *n* × θ	.36	4	.09	1.90	.108
RD × *n* × STEP	.05	4	.01	.25	.908
RD × θ × STEP	.15	4	.04	.76	.550
INIT × *n* × θ	.02	4	.01	.12	.976
INIT × *n* × STEP	.01	4	.00	.06	.993
INIT × θ × STEP	.02	4	.00	.10	.981
n × θ × STEP	.97	8	.12	2.55	.010 *
RD ×INIT ×*n* ×θ	.02	4	.01	.13	.973
RD ×INIT ×*n* ×STEP	.04	4	.01	.20	.937
RD ×INIT ×θ ×STEP	.08	4	.02	.44	.782
RD× *n* × θ × STEP	.03	8	.00	.08	1.000
INIT ×*n* × θ ×STEP	.04	8	.01	.11	.999
RD×INIT×*n*×θ×STEP	.05	8	.01	.13	.998
(Model)	5161.43	107	48.24	1013.50	.000 *
(Total)	5207.70	1079	4.83		
R-Squared =	0.991				
Adj. R-Squared =	0.990				

Univariate Homogeneity of Variance Tests
Cochrans C (9, 108) = .02032, p = 1.000 (approx.)
Bartlett-Box F(107, 78010) = 1.85217, p = .000 *

* significance at $\alpha = 0.05$

Table 23: Effect Size Measures and Observed Power at $\alpha = 0.05$

Source of Variation	Partial ETA^2	Noncentrality	Power
RD	.087	92.040	1.000 *
INIT	.000	0.004	.031
n	.656	1852.920	1.000 *
θ	.991	103777.000	1.000 *
STEP	.169	197.617	1.000 *
RD × INIT	.001	.775	.175
RD × *n*	.010	9.544	.794 *
RD × θ	.098	105.261	1.000 *
RD × STEP	.001	.586	.101
INIT × *n*	.000	.395	.084
INIT × θ	.000	.337	.079
INIT × STEP	.000	.198	.067
n × θ	.688	2146.700	1.000 *
n × STEP	.025	24.826	.988 *
θ × STEP	.168	195.956	1.000 *
RD × INIT × *n*	.000	.051	.055
RD × INIT × θ	.001	.506	.094
RD × INIT × STEP	.001	1.132	.147
RD × *n* × θ	.008	7.611	.578
RD × *n* × STEP	.001	1.010	.107
RD × θ × STEP	.003	3.050	.247
INIT × *n* × θ	.000	.476	.076
INIT × *n* × STEP	.000	.250	.063
INIT × θ × STEP	.000	.415	.072
n × θ × STEP	.021	20.365	.919 *
RD ×INIT × *n* × θ	.001	.508	.077
RD ×INIT × *n* × STEP	.001	.809	.095
RD ×INIT × θ × STEP	.002	1.745	.154
RD × *n* × θ × STEP	.001	.626	.072
INIT × *n* × θ × STEP	.001	.909	.083
RD×INIT×*n*×θ×STEP	.001	1.001	.087

* indicates power of significance factors

5.11 Hypothesis Testing

The effectiveness of *INIT* is based on the analysis of variance (Table 22) for *lnavgfin* since this factor has no interaction effect with other factors. Conclusions on *STEP* and *n* are based on Duncan's multiple range tests due to significance interaction effects. The effect of *n* was investigated based on the setup problems only. For the other two experimental factors (*RD* and θ), *pctabopt* is used as the performance measure. The following standardized notations are used for hypothesis testing:

$Z_{INIT,\ \theta,\ n,\ STEP,\ RD}$ = *lnavgfin*, transformed average objective function value for problems using
INIT initial solution approach (*INIT* = *RND*, *PWS*)
θ tradeoff parameter (θ = 0, 0.5, 1)
n number of jobs in the sequence (*n* = 10, 30, 50)
STEP updates per temperature range (*STEP* = 1, 50, 100)
RD relative range of due date (*RD* = 0.2, 0.5)

$X_{INIT,\ \theta,\ STEP,\ RD}$ = *pctabopt*, percent above optimum solution for 10-job problems using
INIT initial solution approach (*INIT* = *RND*, *PWS*)
θ tradeoff parameter (θ = 0, 0.5, 1)
STEP updates per temperature range (*STEP* = 1, 50, 100)
RD relative range of due date (*RD* = 0.2, 0.5)

5.11.1. The Effect of The Initial Solution - INIT

H_0: $Z_{INIT = RND, \forall n, STEP, RD} = Z_{INIT = PWS, \forall n, STEP, RD}$

H_1: $Z_{INIT = RND, \forall n, STEP, RD} \neq Z_{INIT = PWS, \forall n, STEP, RD}$

This hypothesis investigates the contradictory views on the effect of the initial solution on the quality of the final solution. The *a priori* expectation of this factor was that the quality of the final solution is independent of the initial solution. Table 22 shows the ANOVA for *lnavgfin*. It clearly indicates that the analysis of variance failed to reject the null hypothesis. Table 27 shows the ANOVA for *pctabopt* for the 10-job problems, and it failed to reject the null hypothesis too. There is no interaction between *INIT* and other factors in both tables 22 and 27. There is no evidence to conclude that better than randomly generated initial solution leads to a better final solution. Therefore, this research agreed with the results of Suresh and Sahu (1993), but refuted the contradictory findings of Johnson *et al.* (1989), Potts and Wassenhove (1991), and Shang (1993). This finding is important in that it confirmed the theoretical basis of simulated annealing in which the final solution should be independent of the initial solution. A careful examination of various graphs (e.g. Figures 7 to 10 and APPENDIX 3) suggests at least three possible reasons why some researchers reported that the initial solution has a significant effect on the quality of the final solution. These reasons include: (1) using a starting temperature that was too small, (2) using a temperature decay rate that was too large, and (3) stopped the annealing process too soon. Any of the above reasons would dramatically reduce the chances for the annealing process to escape local optimums.

Depending on the magnitude of the objective function, the starting temperature should yield an initial probability of close to 0.5 on the Boltzmann distribution. The simple reason is that the annealing process should have an equal chance of rejecting or accepting a worse solution at very high temperature. The annealing temperature should be decreased gradually until it reaches a state that the Boltzmann distribution approaches zero. *Section 3.4.4.* provides a sound analytical approach for checking these possible problems.

5.11.2. Number of Updates Between Changes in Temperature (STEP)

H_0: $Z_{INIT,\theta,n,STEP=1,RD} = Z_{INIT,\theta,n,STEP=50,RD} = Z_{INIT,\theta,n,STEP=100,RD}$

H_1: $Z_{INIT,\theta,n,STEP=1,RD} \neq Z_{INIT,\theta,n,STEP=50,RD} \neq Z_{INIT,\theta,n,STEP=100,RD}$

The *a priori* expectation of this experimental factor was that a higher number of updates between changes in temperature range is expected to yield lower average objective values. The probabilities of evaluating every peak and trough between changes in temperature should be greatly enhanced by executing this loop indefinitely (Otten and Ginneken, 1989, pp 14). Therefore, it is expected that a higher number of updates should yield better results.

However, the Duncan's multiple range tests for *lnavgfin* in Table 24 report otherwise. *STEP* is not a significant factor at $\theta = 0.5$ and $\theta = 1.0$, regardless of the job size (n). At $\theta = 0.0$ and $n = 50$, *STEP* = 100 is statistically more effective than STEP = 50, which in turn is statistically more effective than *STEP* = 1. For smaller job size (i.e. θ

= 0.0 and n = 10 or n = 30), both *STEP* = 100 and *STEP* = 50 are statistically more effective than *STEP* = 1, although there is no statistical difference between *STEP* = 50 and *STEP* = 100. All the comparisons for this part of the analyses passed the Levene's homogeneity of variance tests.

Table 24: Duncan's Multiple Range Tests for *STEP* - *lnavgfin* (40 cases per cell)

Means			Duncan's			K-S	Levene's
STEP=1	*STEP*=50	*STEP*=100	Results	θ	*n*	Test	Test
0.9861	0.7352	0.7088	μ1 ≠ μ2 = μ3	0.0	10	.3294	.136
0.8638	0.3758	0.3282	μ1 ≠ μ2 = μ3	0.0	30	.5164	.976
0.8974	0.2818	0.2156	μ1 ≠ μ2 ≠ μ3	0.0	50	.0009	.365
4.1213	4.0848	4.0839	μ1 = μ2 = μ3	0.5	10	.0643	1.000
4.9514	4.8892	4.8856	μ1 = μ2 = μ3	0.5	30	.0200	.836
5.3541	5.2586	5.2518	μ1 = μ2 = μ3	0.5	50	.0068	.562
4.7631	4.7422	4.7427	μ1 = μ2 = μ3	1.0	10	.0332	.998
5.6104	5.5619	5.5655	μ1 = μ2 = μ3	1.0	30	.0118	.996
6.0265	5.9460	5.9371	μ1 = μ2 = μ3	1.0	50	.0194	.796

The most likely explanation for the mixed results for this part of the analyses relates to θ, which in turn determines the magnitude of the objective function values. When the problem changed from a pure setup objective to a combined setup and tardiness, and finally to a pure tardiness problem, the objective function increased at least four fold. The results can be due to the relative difference between the objective function values and the size of *STEP*. When *STEP* was increased from 1 to 50 and then 100, it has a

significant effect on the pure setup problem (where the magnitude of the objective function value is less than 1.0). When θ was increased to 0.5 and 1.0, the magnitude of the objective function values increased to an average of 4 and above (i.e. quadruple). *STEP* size of 1, 50 and 100 were no longer large enough to affect the solution. Therefore, increasing *STEP* size four times (the relative increased in the objective function) to 200 (50 x 4) and 400 (100 x 4) would probably affect the quality of the solution. The arbitrarily chosen factor levels of 1, 50 and 100 are probably too small to make any significant effect on the solution. Future research will be conducted to investigate this rationale, where the factor levels will be set as a percentage of the objective function.

Table 25: Duncan's Multiple Range Tests for *STEP* - *pctabopt* (20 cases per cell)

Mean of *pctabopt*			Duncan's Results	θ	*RD*
STEP=1	*STEP*=50	*STEP*=100			
74.4537	21.7454	24.7161	$\mu 1 \neq \mu 2 = \mu 3$	0.0	0.2
45.6824	23.0876	12.9553	$\mu 1 \neq \mu 2 = \mu 3$	0.0	0.5
3.2579	0.3834	0.4097	$\mu 1 \neq \mu 2 = \mu 3$	0.5	0.2
5.1819	0.3461	0.1445	$\mu 1 \neq \mu 2 = \mu 3$	0.5	0.5
2.6402	0.3732	0.4328	$\mu 1 \neq \mu 2 = \mu 3$	1.0	0.2
2.3501	0.2983	0.3402	$\mu 1 \neq \mu 2 = \mu 3$	1.0	0.5

Table 25 shows the Duncan's multiple range test for *STEP* using *pctabopt* as the performance measure at different level of θ and *RD*. This analysis compares the effect of *STEP* on the quality of the final solution for the 10-job problems. The test shows that the 10-job problems were more difficult to solve when *STEP* = 1, although there is no statistical difference between the means for *STEP* = 50 and 100. At least for the 10-job problems, it can be shown that *STEP* size of 50 or 100 improved the quality of the final solution.

5.11.3. Number of Jobs in the Sequence (n)

H_0: $Z_{INIT,\theta=0.0,n=10,STEP,RD} = Z_{INIT,\theta=0.0,n=30,STEP,RD} = Z_{INIT,\theta=0.0,n=50,STEP,RD}$

H_1: $Z_{INIT,\theta=0.0,n=10,STEP,RD} \neq Z_{INIT,\theta=0.0,n=30,STEP,RD} \neq Z_{INIT,\theta=0.0,n=50,STEP,RD}$

This test is meaningful only when θ = 0.0 (100% setup). When θ = 0.5 or 1.0, the performance measure, *avgfin*, involve tardiness which is not a linear function of *n*. Therefore, dividing the total objective function values by *n* does not yield a good 'average' measure for comparing tardiness with different job size. The problem is expected to be more difficult to solve with larger number of jobs in the sequence, thus resulting in higher average setup time per job. Table 26 shows that when θ = 0.0, and *STEP* = 1 or 50, the cell mean of *n* = 10 is statistically larger than the cell mean of *n* = 30 or *n* = 50. There is no statistical difference between the cell means of *n* = 30 and *n* = 50. At *STEP* = 100, the cell mean of *n* = 10 is statistically larger than the cell mean of *n* = 30, which is in turn statistically larger than the cell mean of *n* = 50. The results suggest that in general, it is

more difficult to minimize setup for smaller problem using simulated annealing. The simple explanation to this peculiar finding can be that the effect of including a 'bad' job (one with very high setup) in the sequence is minimized when the number of jobs in the sequence is large (i.e. larger denominator).

Table 26: Duncan's Multiple Range tests for *n* (40 cases per cell)

Means			Duncan's			K-S	Levene's
$n = 10$	$n = 30$	$n = 50$	Results	θ	*STEP*	Test	Test
0.9861	0.8638	0.8974	$\mu1 \neq \mu2 = \mu3$	0.0	1	.6154	.002
0.7352	0.3758	0.2818	$\mu1 \neq \mu2 = \mu3$	0.0	50	.2156	.000
0.7088	0.3282	0.2156	$\mu1 \neq \mu2 \neq \mu3$	0.0	100	.0484	.000

5.11.4. Tightness of the Relative Range of Due Dates (RD)

H_0: $X_{INIT,\theta,STEP,RD=0.2} = X_{INIT,\theta,STEP,RD=0.5}$

H_1: $X_{INIT,\theta,STEP,RD=0.2} \neq X_{INIT,\theta,STEP,RD=0.5}$

This test is meaningful only when θ = 0.5 or 1.0 because *RD* affects only tardiness objective. The *a priori* expectation was that the narrower the relative range of due dates, the more difficult is the problem. This expectation is consistent with Rinnooy Kan, *et al.* (1975), Ragatz (1993) and Rubin and Ragatz (1995). Table 27 shows the ANOVA using *pctabopt* as the performance measure. The interaction of *RD* with *STEP* and θ is

significant. Comparisons between the means of *RD* are conducted by fixing *STEP* and θ at a specific level, and applying the independent t-test to the means of *RD* at that level. Separate-variance t-test (marked as *Unequal* in SPSS outputs) was used where the Levene's test for equality of variance rejected the null hypotheses (H_O: $\sigma^1 = \sigma^2$), suggesting that the population variances from the two samples differed. Pooled-variance t-test (marked as *Equal*) was used where the Levene's test statistic was greater than 0.05, suggesting that there was no reason to suspect the population variances in the two groups were unequal. However, for large samples, the difference between these two tests was very small. Indeed, test results for *RD* showed exactly the same *p*-values, *t*-values, and 95% confidence interval for many cases. The only difference was that separate-variance t-test has a smaller degree of freedom (df) for the test.

Table 27: ANOVA - *pctabopt*

Source of Variation	SS	DF	MS	F	Sig of F
WITHIN+RESIDUAL	95185.49	324	293.78		
INIT	58.17	1	58.17	.20	.657
STEP	18469.19	2	9234.59	31.43	.000 *
RD	1606.63	1	1606.63	5.47	.020 *
θ	84138.27	2	42069.13	143.20	.000 *
INIT × *STEP*	114.94	2	57.47	.20	.822
INIT × *RD*	143.01	1	143.01	.49	.486
INIT × θ	5.14	2	2.57	.01	.991
STEP × *RD*	1342.75	2	671.37	2.29	.103
STEP × θ	23798.71	4	5949.68	20.25	.000 *
RD × θ	3522.31	2	1761.15	5.99	.003 *
INIT × *STEP* × *RD*	1305.35	2	652.68	2.22	.110
INIT × *STEP* × θ	515.85	4	128.96	.44	.780
INIT × *RD* × θ	11.74	2	5.87	.02	.980
STEP × *RD* × θ	3246.09	4	811.52	2.76	.028 *
INIT × *STEP* × *RD* × θ	1846.76	4	461.69	1.57	.182
(Model)	140124.90	35	4003.57	13.63	.000 *
(Total)	235310.39	359	655.46		
R-Squared =	.595				
Adj. R-Squared =	.552				

* significance at $\alpha = 0.05$

Table 28: t-test for Equality of Means - *RD* (20 cases per cell)

Reject H_0	Mean of *pctabopt*		θ	*STEP*	2-Tail Sig	95% CI for Diff	Levene's Test
	RD = 0.2	*RD* = 0.5					
NO	3.2579	5.1819	0.5	1	0.255	-5.30, +1.46	0.021*
NO	0.3834	0.3461	0.5	50	0.217	-0.40, +0.48	0.518
NO	0.4097	0.1445	0.5	100	0.177	-0.13, +0.66	0.129
NO	2.6402	2.3501	1.0	1	0.769	-1.70, +2.28	0.315
NO	0.3732	0.2983	1.0	50	0.290	-0.51, +0.66	0.870
NO	0.4328	0.3402	1.0	100	0.234	-0.38, +0.57	0.860

* indicates separate variance t-test was used since H_0: $\sigma_1^2 = \sigma_2^2$ was rejected.

Table 28 shows the results of the t-test for equality of means for *RD* at different levels of *STEP* and θ. Test results cannot reject the null hypotheses regardless of the levels of *STEP*. Therefore, there is no evidence to believe that it is more difficult for simulated annealing to solve the 10-job problems when *RD* = 0.2. Rinnooy Kan, *et al.* (1975), Ragatz (1993) and Rubin and Ragatz (1995) reported that it was more difficult for their solution approaches to solve the problems with tight relative range of due dates. This suggests that simulated annealing may be the preferred approach for solving tardiness problems with tight relative due date range.

5.11.5. Tradeoff Parameters (θ)

H_0: $X_{INIT,\theta=0.0,\ STEP,\ RD} = X_{INIT,\theta=0.5,\ STEP,\ RD} = X_{INIT,\theta=1.0,\ STEP,\ RD}$

H_1: $X_{INIT,\theta=0.0,\ STEP,\ RD} \neq X_{INIT,\theta=0.5,\ STEP,\ RD} \neq X_{INIT,\theta=1.0,\ STEP,\ RD}$

The *a priori* expectation was that the single objective sequencing problem is easier to solve than the multi-objective problem (i.e. θ = 0.5), because when θ = 0.5, the heuristic needs to consider a weighted average of the two objectives. Therefore, the problem was expected to be more complex and difficult to solve, thus producing higher 'percent above optimum', *pctabopt*, for the 10-job problems.

Table 29 shows the results of Duncan's multiple range tests on *pctabopt* with different levels of *STEP* and *RD*. The results rejected the *a priori* expectation. There is no evidence to support the *a priori* expectation that multi-objective problem was more difficult to solve. However, test results show that the pure setup problem was more

difficult to solve. The mean for the pure setup problem is substantially higher than those involving tardiness ($\theta = 0.5$ and $\theta = 1.0$). The explanation could be that the perturbation scheme was structured to perform better in exploring the 'neighborhood' of the optimum solution for tardiness problems than the setup problems.

Table 29: Duncan's Multiple Range Tests for θ - *pctabopt* (20 cases per cell)

Mean of *pctabopt*			Duncan's Results	*STEP*	*RD*
$\theta = 0.0$	$\theta = 0.5$	$\theta = 1.0$			
74.4537	3.2579	2.6402	$\mu1 \neq \mu2 = \mu3$	1	0.2
45.6824	5.1819	2.3501	$\mu1 \neq \mu2 = \mu3$	1	0.5
21.7454	0.3834	0.3732	$\mu1 \neq \mu2 = \mu3$	50	0.2
23.0876	0.3461	0.2983	$\mu1 \neq \mu2 = \mu3$	50	0.5
24.7161	0.4097	0.4328	$\mu1 \neq \mu2 = \mu3$	100	0.2
12.9553	0.1445	0.3402	$\mu1 \neq \mu2 = \mu3$	100	0.5

CHAPTER 6: CONCLUSIONS

6.1 Conclusion

Although a significant amount of research exists on operations scheduling problems, most of it either totally ignores setup times or assumes that setup times on each machine are independent of the job sequence. In addition, most past research tends to focus on a single objective. This research investigates the multi-objective scheduling problem with sequence dependent setup times. By varying a tradeoff parameter, θ, between 0 and 1, schedulers can now sequence to minimize a weighted sum of more than one objective. By setting $\theta = 1$ for a particular objective and 0's for the other objectives, the algorithm becomes a single objective minimization problem. This research provides an attractive and efficient approach to enable the industrial schedulers to strive for a balance among various scheduling objectives.

Schedulers are not limited to the single objective minimization tools that tend to minimize a sequencing objective yet experience an unacceptable tradeoff of other objectives. For instance, a production sequence that minimizes setup may be unacceptably high on tardiness and work in process or vice versa. Although this research deals with a maximum of two objectives, the same principle can be applied if a scheduler needs to consider more than two objectives. The number and the importance (or weights) of various sequencing objectives can easily be increased or decreased by simply changing the value of the tradeoff parameters, (e.g. letting a tradeoff parameter = 0 would effectively eliminate the associated objective).

Simulated annealing is used to arrive at a good solution to the sequencing problem. The algorithm identified many optimum solutions in the smaller problems, although it is not known if the algorithm successfully identified any optimum solutions for the larger problems. In the 10-job problems, simulated annealing positively identified the optimum solutions for 40% or 144 (Table 16: 36.1% x 180 + 43.9% x 180) problems out of a total of 360 problems. Simulated annealing offers a powerful search heuristic for obtaining excellent solutions in this research. The task of multi-objective sequencing was easily implemented as a simulated annealing problem, and yet the same formulation could be used to solve single objective sequencing problem.

The algorithm retained a feasible solution and its corresponding objective value at all times. In addition, scheduler implementing a simulated annealing approach to scheduling can record any schedule that is better than the best incumbent. The implications for 'on-line' production scheduling is that the best incumbent can be used should there be a need to terminate the annealing process prematurely or unexpectedly. The implications of last-minute changes to a production schedule is that the simulated annealing heuristic can quickly reach a 'near optimum' solution. The algorithm is easily implemented on a personal computer.

It is difficult to determine if the annealing process has positively located the optimum solution for any problems. However, it is rather easy to ensure that the annealing process has reached the steady state (i.e. the objective function value has remained constant for an extended period of time) before stopping the process simply by plotting the objective function values over times (Figures 7 to 10 and Appendices 1 and 2). The

annealing process is deemed to have reached the steady state when the probability of accepting a worse solution approaches zero. The correct annealing parameters must be selected to ensure an efficient annealing scheme that will locate a good production sequence with minimum computation times. Although simulated annealing has been used successfully in solving various complex optimization problems, there is a lack of proper guidelines regarding how these parameters can be selected. Past research has typically treated these parameters as experimental factors. However, it is clear that these parameters can be derived based on the theoretical basis of simulated annealing as presented in section 3.4.4.

The results of this research agrees with the theoretical basis of simulated annealing and the findings of Suresh and Sahu (1993) that a properly tuned simulated annealing scheme should be independent of the initial solution. Using better than randomly generated initial solution did not yield better final solution. The usage of incorrect annealing parameters probably accounted for the contradictory findings of Johnson, *et al.* (1989), Potts and Wassenhove (1991), and Shang (1993).

Contrary to the solution approaches used in Rinnooy Kan, *et al.* (1975), Ragatz (1993) and Rubin and Ragatz (1995), this research shows that the simulated annealing heuristic was not affected by the relative range of due dates. It indicates that simulated annealing may be the preferred solution technique to solve sequence dependent scheduling problems with tight relative range of due dates.

As for the effect of job size (n) on the setup problem, research results indicate that smaller problems tend to yield higher average setup. A probable explanation is that the

effect of selecting a 'bad' job with high setup will affect the average objective value of the smaller problem (smaller denominator) more than the larger problem (larger denominator).

The research results also indicate that although higher updates between changes in temperature produced lower average objective value for the pure setup problem, it has no statistical impact on the other problems. This finding is rather unexpected because theoretically higher updates should lead to better solutions. An explanation is that the magnitude of the pure setup problems are much smaller than the other problems. There may be a relationship between the size of the updates and the magnitude of the objective function value. The levels (*STEP* = 1, 50, 100) selected for this experimental factor are probably too small for those problems involving tardiness. When the performance measure (*pctabopt*) was used to compare the results of the 10-job problems, test statistics show that *STEP* = 50 and 100 yield final solutions closer to the optimum solution than when *STEP* = 1.

6.2 Assumptions/Limitations of Present Research

The conclusions and findings from this research although suggestive are not conclusive beyond the scope of this research. Specifically, the assumptions made in this research were:

(i) Processing times of the jobs are normally distributed, $p \sim N(100, 25^2)$,

(ii) Setup times of the jobs are uniformly distributed, $s \sim U(0, 19)$,

(iii) Due dates of the jobs are uniformly distributed, $d \sim U(25n, 75n)$ or $d \sim (25n, 75n)$,

(iv) Sequencing a single processor or machine,

(v) No machine breakdowns, and

(vi) All information of the jobs (p, s, d) are known when the jobs arrived at the shop,

(vii) No cancellation of jobs or changes in due dates,

(viii) Simulated annealing is the only solution technique used.

These assumptions and problem environment suggested by the literature were adopted so that results can be compared to past scheduling research. The usefulness and efficacy of simulated annealing in solving more realistic problems (those involving multiple machines with random arrival of jobs and multiple routings) is still a matter for investigation. The results of this research suggest that simulated annealing might hold considerable promise in solving these complex scheduling problems.

6.3 Suggestions For Future Research

Future research could explore the effectiveness of simulated annealing in solving more complex scheduling problems. The following ideas seem worth pursuing:

(i) In many instances, simulated annealing was unable to return to a better solution found earlier in the annealing process. Future research will investigate the effect of adding a "memory loop" in the simulated annealing scheme in an attempt to improve the annealing process. Solution approaches include: (a) replacing the final solution with a previous solution at the end of the annealing process if the latter is a better solution, (b) returning to the previous solution if the annealing process cannot find a better solution after a certain number of iterations, or (c) heating up the annealing process by increasing the temperature and decreasing the decay rate if the final solution is worse than a previous found solution.

(ii) Comparing the performance of simulated annealing to (a) Next Best Rule (NB), (b) Next Best Rule With Variable Origin (NB'), (c) Next Best Rule After Column Deduction (NB") (Gavett 1965), and (d) Lin-Kernighan's TSP algorithm (Lin and Kernighan, 1973) for minimizing sequence dependent setup times.

(iii) Comparing the performance of simulated annealing to (a) Branch and Bound (Ragatz, 1993), and (b) Genetic Search (Rubin and Ragatz, 1995) for minimizing tardiness with sequence dependent setup times.

(iv) Investigating whether it is possible to extend Gavett's NB, NB', and NB" to handle minimizing tardiness problem.

(v) Extending the research to cover more than one machine scheduling problem. For example, in the case of two machines, the objective function becomes:

$$H = \theta \text{ (tardiness)} + (1 - \theta) * (S_1 + S_2)$$

where,
S_1 is the setup times incurred on machine 1
S_2 is the setup times incurred on machine 2

(vi) Investigating the relationship of the ratio of the various annealing parameters (T_{max}, T_0, and r) to the objective function value and the quality of the final solution.

(vii) Sensitivity analysis of the simulated annealing heuristic to the distribution of job parameters (i.e. processing time, setup time and due date distribution).

6.4 Contributions to Operations Management and Scheduling Literature

This research addressed many issues in scheduling problems that have been typically ignored by past research. By addressing these issues, the problem setting considered in this research is more realistic. The issues addressed include:

- This research looked at sequence dependent setup times, which often occurs in practice but usually ignored in research,

- This research addressed minimizing tardiness objective, which is very important to the industrial schedulers, but has been ignored in sequence dependent setup environment
- This research also addressed the multiple objective scheduling problems in a sequence dependent setup environment. It allows the scheduler to consider more than one objective, or to strive a balance between conflicting objectives.

This research shows that simulated annealing offers a promising alternative to solve scheduling problems where setup times are sequence dependent. Practicality of simulated annealing approach for scheduling include:

- Simulated annealing can often derive a fairly good solution quickly (although not necessary the optimum solution), and it can retain the best incumbent at all time. Therefore, it is possible to generate new schedule fairly quickly or to reschedule frequently if necessary.
- Simulated annealing is robust. It can consider a variety of environments and objectives. It can also be used to solve facility layout problems and to schedule workforce.
- It can be easily implemented on a personal computer.
- Any feasible initial solution is a good starting point. Therefore, it is unnecessary to waste resources trying to derive a good initial solution.
- This research also provides a good analytical approach for establishing the proper annealing parameters.

APPENDICES

APPENDIX 1: CHARTS OF CASES WITH WORSE FINAL SOLUTIONS

Chart 1: Case 1 to 3

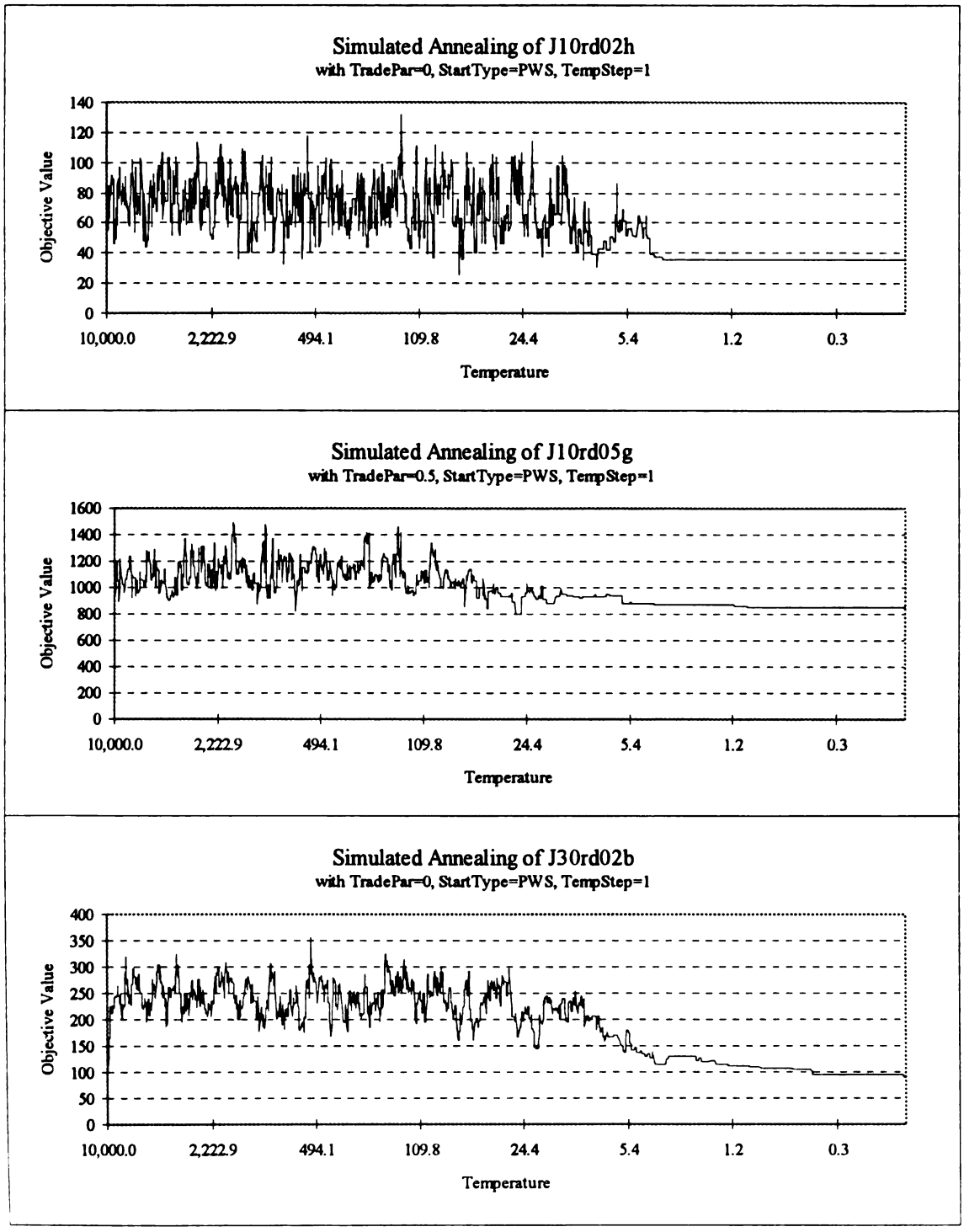


Chart 2: Case 4 to 6

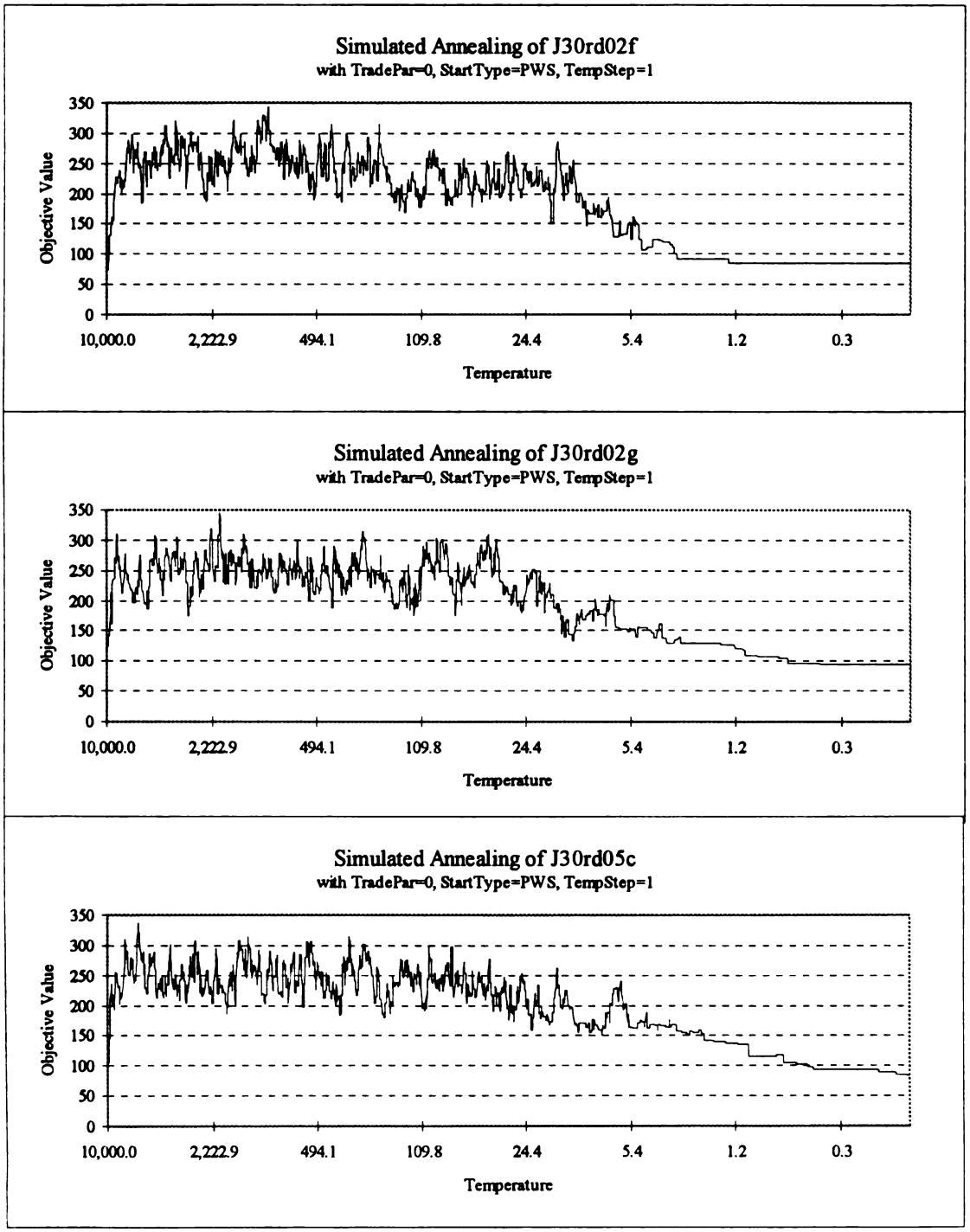


Chart 3: Case 7 to 9

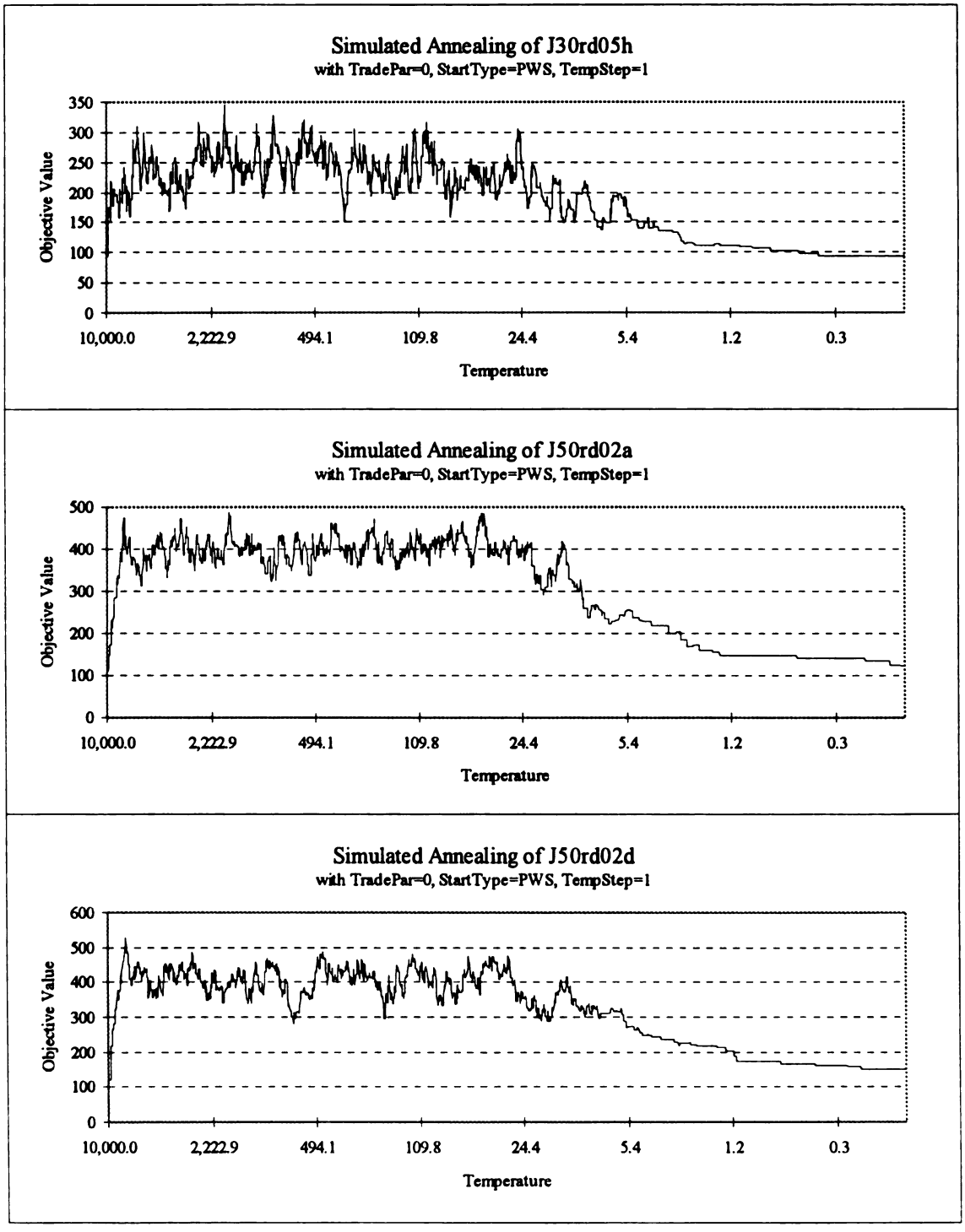


Chart 4: Case 10 to 12

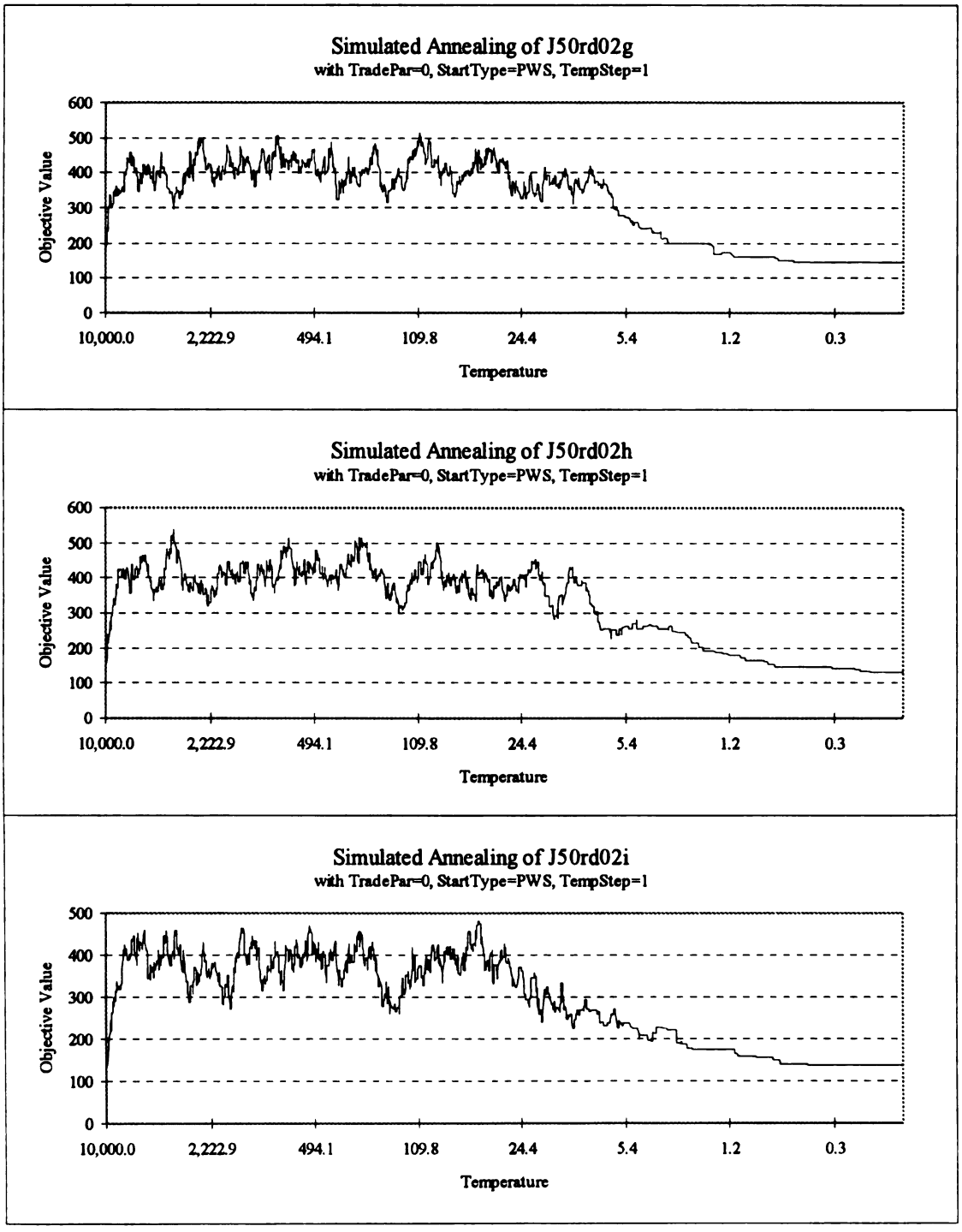


Chart 5: Case 13 to 15

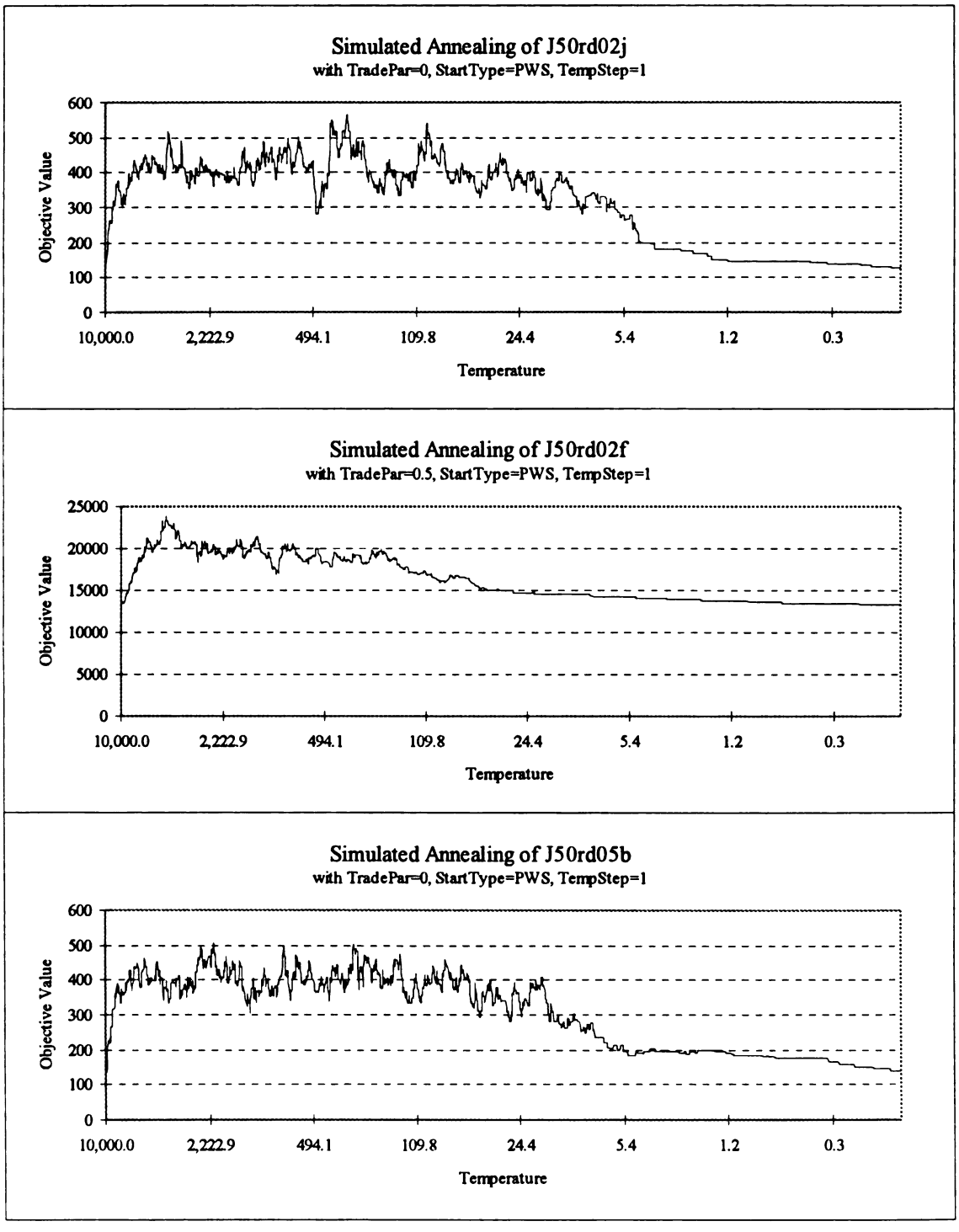


Chart 6: Case 16 to 18

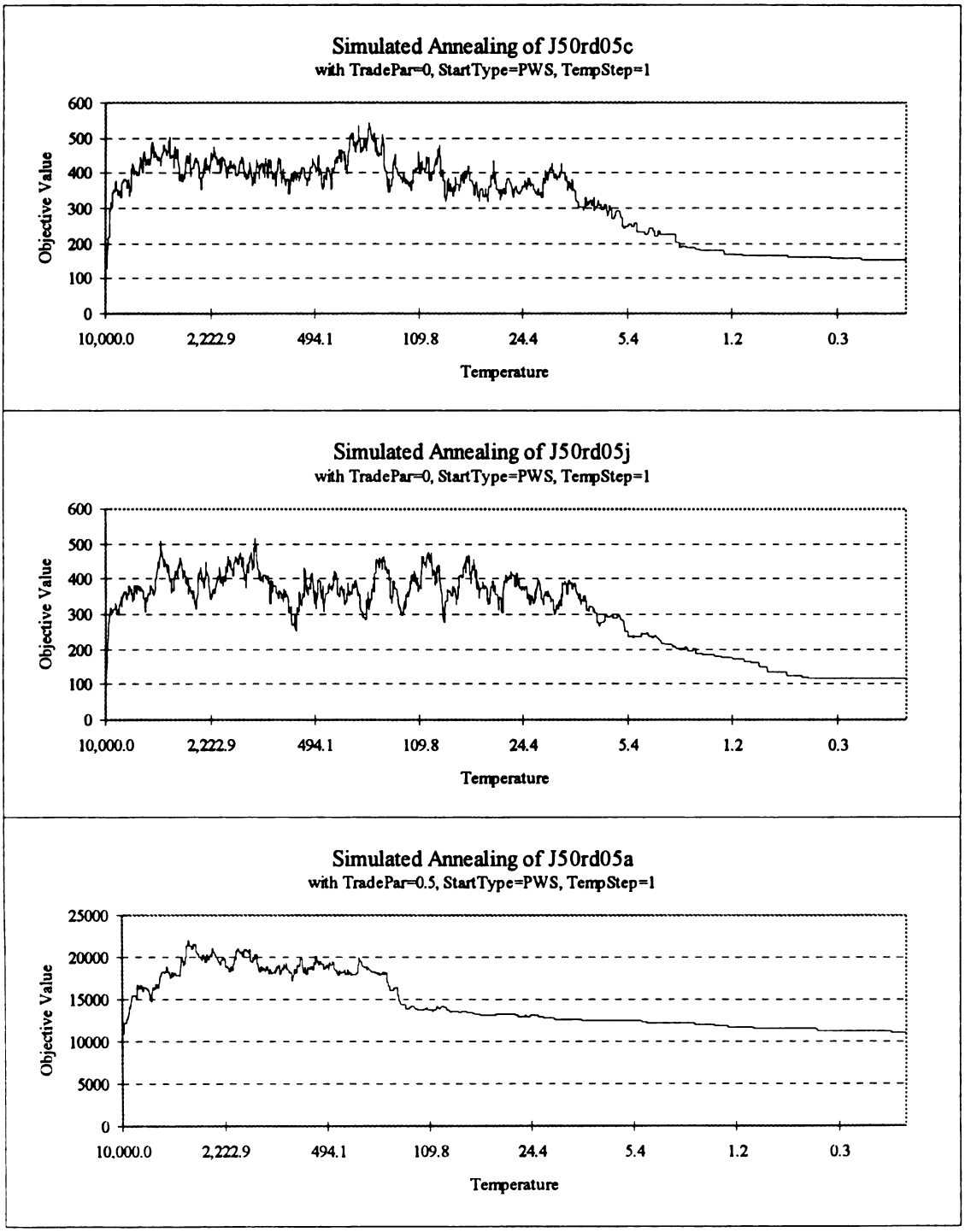
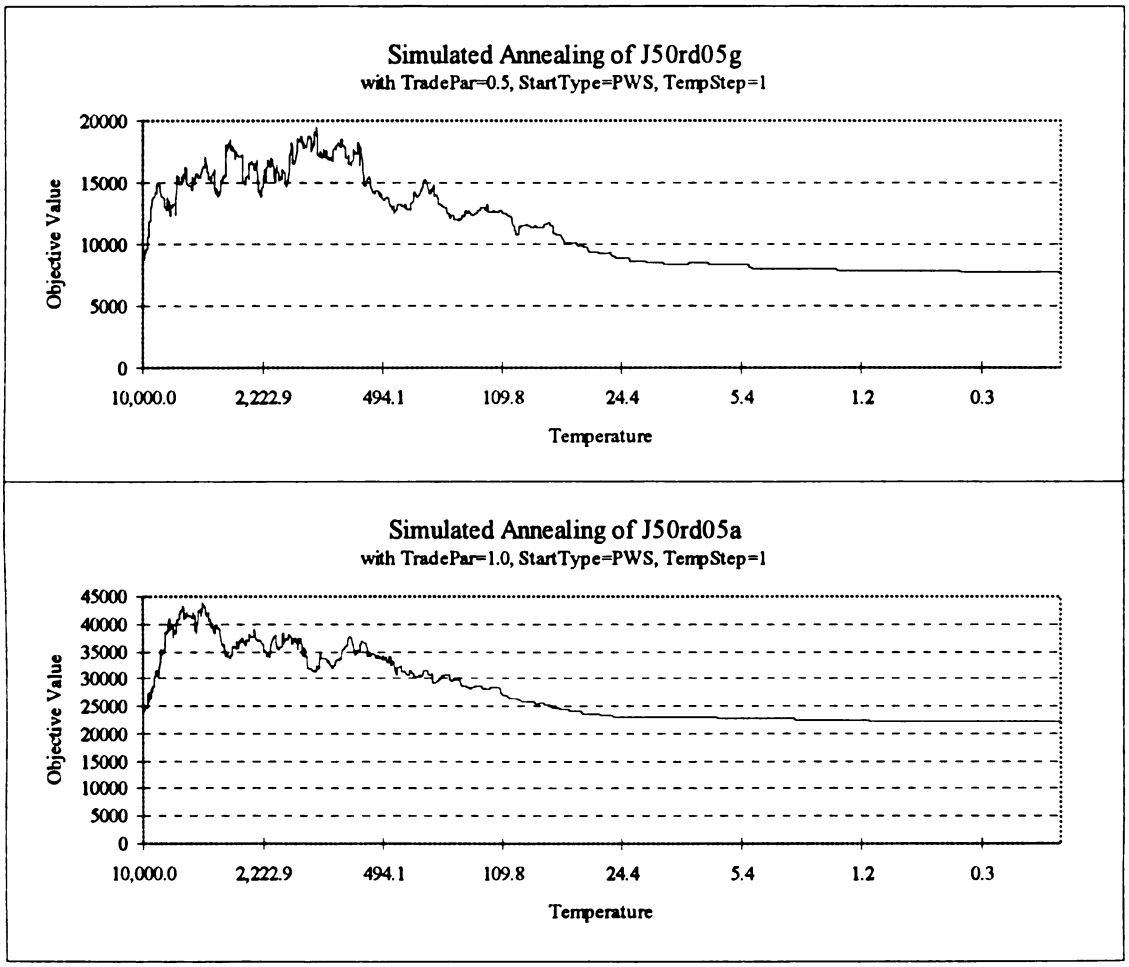


Chart 7: Case 19 & 20



APPENDIX 2: THREE-WAY ANOVA - *LNAVGFIN*

Source of Variation	SS	DF	MS	F	Sig of F
WITHIN+RESIDUAL	46.26	972	.05		
NRDθ	5139.85	17	302.34	6352.39	.000 *
INIT	.00	1	.00	.00	.950
STEP	9.41	2	4.70	98.81	.000 *
NRDθ × *INIT*	.15	17	.01	.18	1.000
NRDθ × *STEP*	11.73	34	.34	7.25	.000 *
INIT × *STEP*	.01	2	.00	.10	.906
NRDθ × *INIT* × *STEP*	.30	34	.01	.18	1.000
(Model)	5161.43	107	48.24	1013.50	.000 *
(Total)	5207.70	1079	4.83		
R-Squared =	0.991				
Adj. R-Squared =	0.990				

Univariate Homogeneity of Variance Tests

Cochrans C (9, 108) = .02032, p = 1.000 (approx.)

Bartlett-Box F(107, 78010) = 1.85217, p = .000 *

* significance at $\alpha = 0.05$

Three-Way ANOVA for *LNAVGFIN*:

n, *RD*, and θ were combined into one factor, *NRDθ*, with 18 levels (3 × 2 × 3). The results are similar to those presented in Table 22: Full Factorial ANOAV - *LNAVGFIN*.

APPENDIX 3: J30RD05E WITH LOWER TEMPERATURE DECAY RATE

Final Solutions With Lower T_{max} and r

T_{max} =	10,000	25	25
T_0 =	0.10	0.05	0.05
r =	0.995	0.9973	0.999
# of iterations =	2297	2297	6622
STEP = 1	46	56	63
STEP = 50	41	42	41
STEP = 100	51	47	37
AVERAGE	46.00	48.33	47.00

Note:

This section relates to Section 5.5, page 68. It shows the additional two runs for a particular 30-job problem (J30rd05e) to investigate if using a smaller T_{max}, T_0, and r would produce better result. The first column shows the original results and annealing parameters used for this research. The second column shows the results for a different set of annealing parameters used, but maintaining the same number of iterations to bring T_{max} to T_0 (i.e. 2297). The third column allows a larger number of iterations with a much smaller r. From these two additional runs, there is no clear indication that the result is better with smaller parameters. Charts 8 and 9 show how the objective function values of these two additional runs decreased as the temperature was decreased.

The number of iterations to bring T_{max} to T_0 was calculated as followed:

$$\#\text{ of iterations} = \frac{\ln(T_0 / T_{max})}{\ln(r)}$$

Chart 8: Objective Values As Temperature Decreased

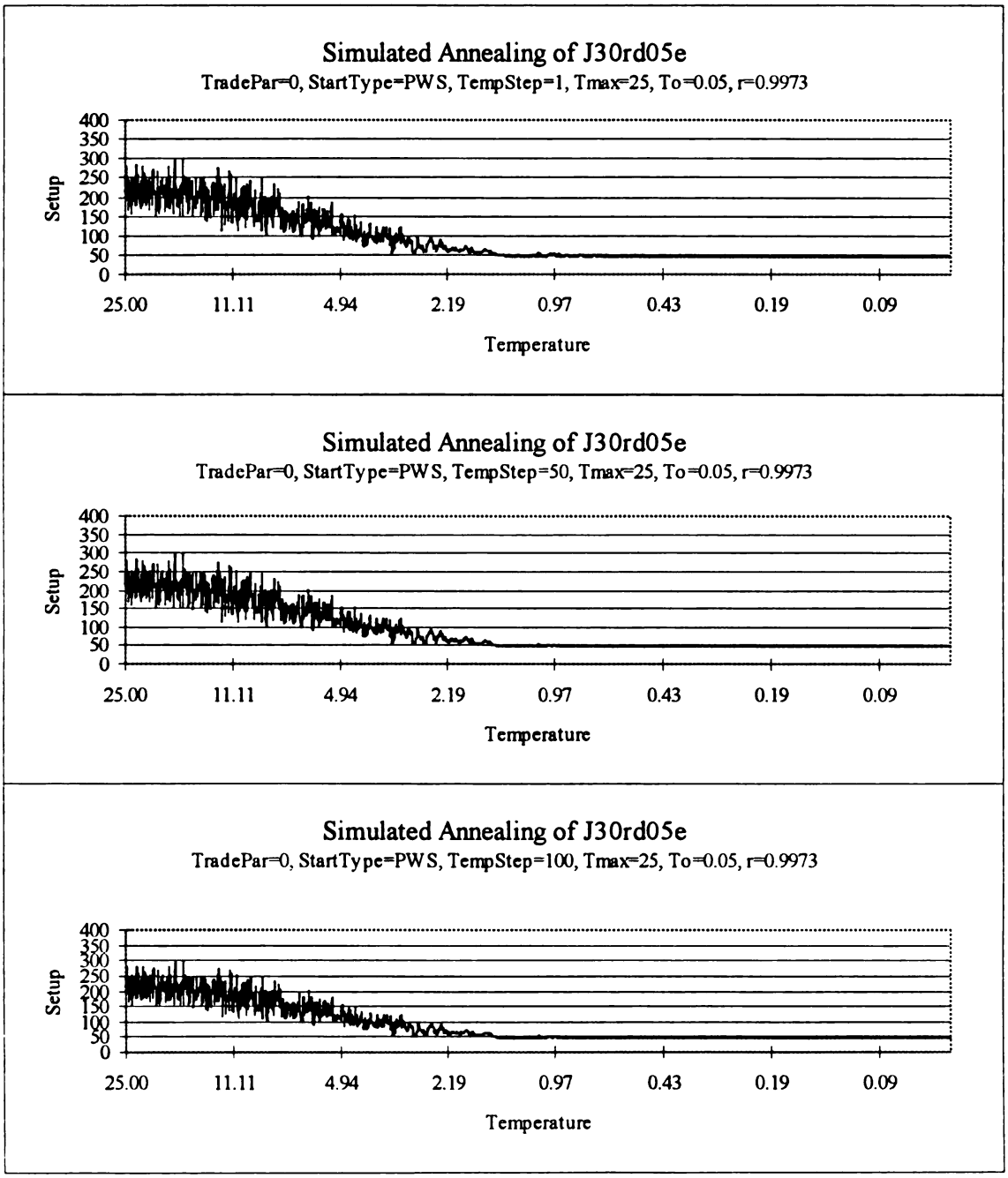
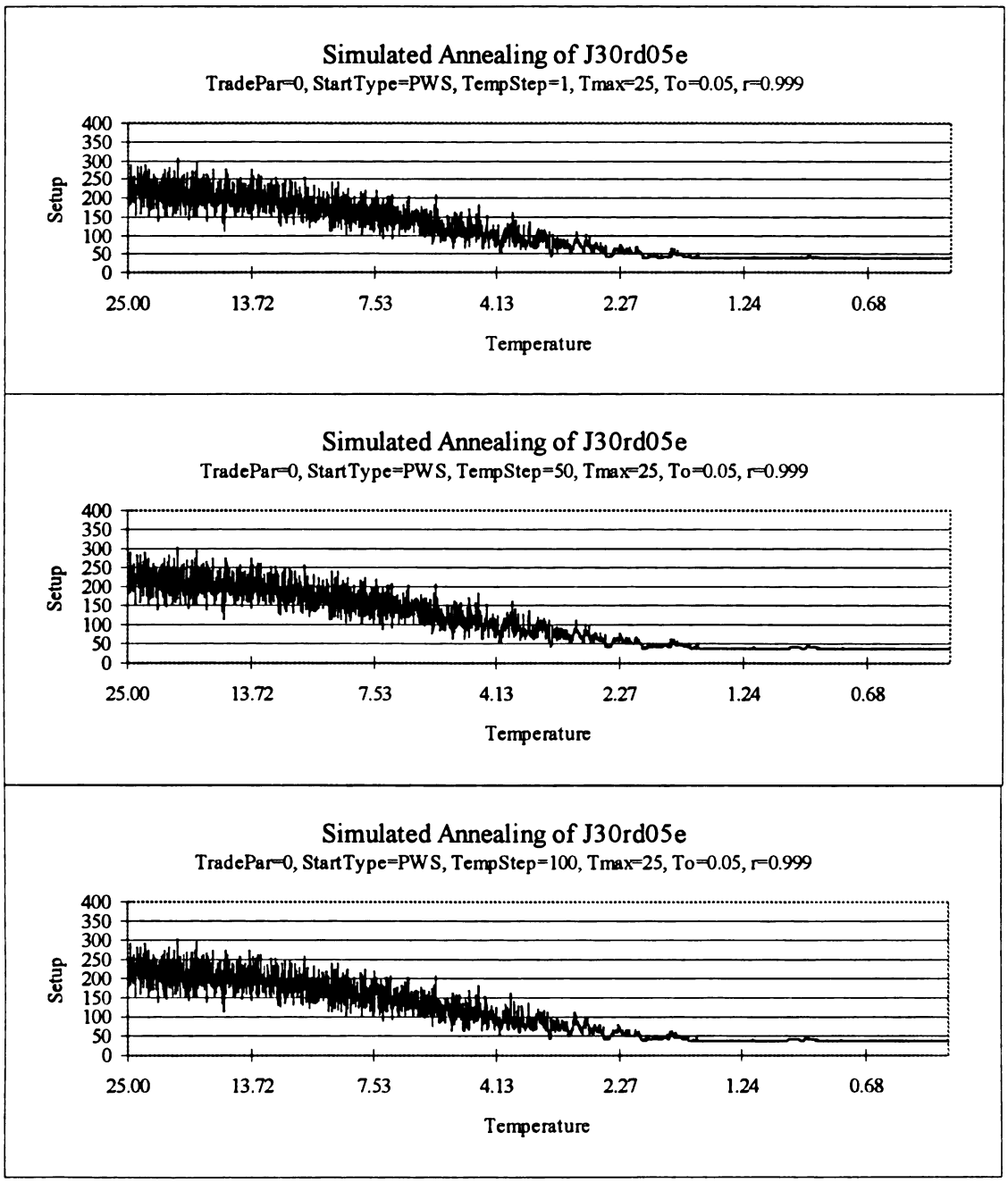


Chart 9: Objective Values As Temerature Decreased



APPENDIX 4: VISUAL BASIC SIMULATED ANNEALING CODE

```
Option Explicit
' Declare all variables required in this form
Dim StepPerTemp, JobTemp(2), RndIntNo(2), I, J, K1, K2, K3, K4, n, A, B, C, D(9) As Integer
Dim DueDate(10), ProcTime(10), SetupTime(10, 10), Job(10), JobInit(10), JobSW(10) As Integer
Dim NoOfIter, TimeLength, Setup, Tardy, TardyTemp As Long
Dim Energy, EnergyInit, EnergySW, EnergyNew, EnergyChange As Single
Dim Tmax, TFinal, TempRate, TNow, TradeOffPar As Single
Dim BoltzProb, BoltzRnd As Double
Dim DataFile As String


Sub cmdCancelFrm2_Click ()
  End
End Sub


Sub cmdContinueFrm2_Click ()

 'Check all parameters entered are valid
 ' (a) Does txtTmax.Text have a value?
 If Tmax <= 0 Then
  MsgBox "You must enter a valid Starting Temperature, Tmax"
  txtTmax.SetFocus
  Exit Sub
 End If

 ' (b) Does txtTFinal.Text have a value?
 If TFinal <= 0 Or TFinal > Tmax Then
  MsgBox "TFinal must be positive, and TFinal < Tmax"
  txtTFinal.SetFocus
  Exit Sub
 End If

 ' (c) Does txtTempRate.Text have a value?
 If TempRate <= 0 Or TempRate > Tmax Then
  MsgBox "Temp Step positive, and TempRate < Tmax"
  txtTempRate.SetFocus
  Exit Sub
 End If

 ' Randomize random seed
 Randomize
```

```
' *** 10-Job Problems ***
n = 10

'(1) First loop, read data files
For K1 = 1 To 10
Select Case K1
 Case 1
  DataFile = "j10rd02a.dat"
 Case 2
  DataFile = "j10rd02b.dat"
 Case 3
  DataFile = "j10rd02c.dat"
 Case 4
  DataFile = "j10rd02d.dat"
 Case 5
  DataFile = "j10rd02e.dat"
 Case 6
  DataFile = "j10rd02f.dat"
 Case 7
  DataFile = "j10rd02g.dat"
 Case 8
  DataFile = "j10rd02h.dat"
 Case 9
  DataFile = "j10rd02i.dat"
 Case 10
  DataFile = "j10rd02j.dat"
End Select

' Read data file & all variables
Open "c:\vb\j10rd02\" & DataFile For Input Access Read As #1
Do While Not EOF(1)
 Input #1, I, B, C, D(1), D(2), D(3), D(4), D(5), D(6), D(7), D(8), D(9), A
 DueDate(I) = B: ProcTime(I) = C
 For J = 1 To n - 1
  If I > J Then
   SetupTime(I, J) = D(J)
  Else
   SetupTime(I, J + 1) = D(J)
  End If
 Next J
Loop
Close #1

' Write to "J10RD0??.chk" file to check if data reading is correct
Open "c:\vb\j10rd02\" & Left$(DataFile, 8) & ".chk" For Output As #2
  For I = 1 To n
     Print #2, I; ","; DueDate(I); ","; ProcTime(I); ",";
     For J = 1 To n
        Print #2, SetupTime(I, J); ",";
     Next J
     ' Force printing to new line
     Print #2, I
  Next I
```

```
Close #2

' Show time & the current file
Cls
Print "Start Time = " & Time
Print "Working on file " & DataFile
Print "Please do not re-boot"
Print "Home Phone:   355-2960"
Print "Office Phone:  353-6498"

'(2) Second loop, TradeOffPar
For K2 = 1 To 3
Select Case K2
  Case 1
    TradeOffPar = 0
  Case 2
    TradeOffPar = .5
  Case 3
    TradeOffPar = 1
End Select

'(3) Third loop, initial solution
For K3 = 1 To 2
  If K3 = 1 Then
    ' Random Initial Solution & set all JobInit = JobInit(1)
    JobInit(1) = 1 + Int(Rnd * n)
    For A = 2 To n
      JobInit(A) = JobInit(1)
    Next A

    ' Ensure each job occurs only once
    For A = 2 To n
      For B = 1 To (A - 1)
        If JobInit(A) = JobInit(B) Then
          JobInit(A) = 1 + Int(Rnd * n)
          B = 0
        End If
      Next B
    Next A

    ' calculate Initial Energy function
    TimeLength = ProcTime(JobInit(1))
    TardyTemp = TimeLength - DueDate(JobInit(1))
    If TardyTemp < 0 Then
      TardyTemp = 0
    End If
    Tardy = TardyTemp
    Setup = 0

    For A = 1 To n - 1
      TimeLength = TimeLength + SetupTime(JobInit(A), JobInit(A + 1)) + ProcTime(JobInit(A + 1))
      TardyTemp = TimeLength - DueDate(JobInit(A + 1))
      If TardyTemp < 0 Then
```

```
    TardyTemp = 0
   End If
   Tardy = Tardy + TardyTemp
   Setup = Setup + SetupTime(JobInit(A), JobInit(A + 1))
  Next A

  EnergyInit = TradeOffPar * Tardy + (1 - TradeOffPar) * Setup

 Else
  ' PairWise Swap
  For A = 1 To n
   JobSW(A) = JobInit(A)
  Next A

  EnergySW = EnergyInit

  For I = 1 To n
   For J = 1 To n
    If I < J Then
     JobTemp(1) = JobSW(I)
     JobTemp(2) = JobSW(J)
     JobSW(I) = JobTemp(2)
     JobSW(J) = JobTemp(1)

     ' calculate Swapped Energy function
     TimeLength = ProcTime(JobSW(1))
     TardyTemp = TimeLength - DueDate(JobSW(1))
     If TardyTemp < 0 Then
      TardyTemp = 0
     End If
     Tardy = TardyTemp
     Setup = 0

     For A = 1 To n - 1
      TimeLength = TimeLength + SetupTime(JobSW(A), JobSW(A + 1)) + ProcTime(JobSW(A + 1))
      TardyTemp = TimeLength - DueDate(JobSW(A + 1))
      If TardyTemp < 0 Then
       TardyTemp = 0
      End If
      Tardy = Tardy + TardyTemp
      Setup = Setup + SetupTime(JobSW(A), JobSW(A + 1))
     Next A

     ' accept swap if lower Energy, else return to original state
     EnergyNew = TradeOffPar * Tardy + (1 - TradeOffPar) * Setup
     If EnergyNew < EnergySW Then
      EnergySW = EnergyNew
     Else
      JobSW(I) = JobTemp(1)
      JobSW(J) = JobTemp(2)
     End If
    End If
   Next J
```

```
  Next I
 End If

'(4) Fourth loop, StepPerTemp
For K4 = 1 To 3
 Select Case K4
  Case 1
   StepPerTemp = 1
  Case 2
   StepPerTemp = 50
  Case 3
   StepPerTemp = 100
 End Select

' Set variables to keep track of statistics
NoOfIter = 0: TNow = Tmax

' Select the correct initial solution
If K3 = 1 Then
 For A = 1 To n
  Job(A) = JobInit(A)
 Next A
 Energy = EnergyInit
Else
 For A = 1 To n
  Job(A) = JobSW(A)
 Next A
 Energy = EnergySW
End If

' open file #3 as ouput, save k1 to k4, JobInit(), EnergyInit
Open "c:\vb\j10rd02\F" & K1 & K2 & K3 & K4 & ".sim" For Output As #3
 Print #3, "DataFile, K1 "; DataFile
 Print #3, "Trade Off Parameter, K2 "; TradeOffPar
 Print #3, "Type of Initial Solution, K3 "; K3
 Print #3, "Step Per Temperature Range, K4 "; StepPerTemp
 Print #3, "Starting Temperature, Tmax "; Tmax
 Print #3, "Frozen temparature, TFinal "; TFinal
 Print #3, "Temperature Step, TempRate "; TempRate
 For I = 1 To n
  Print #3, "The "; I; "th job is job id "; Job(I)
 Next I
 Print #3, "The Initial Energy is "; Energy

 ' 2. Proposed simulated annealing scheme
 Do While TNow > TFinal
  ' 2.1 perform the loop L times
  For I = 1 To StepPerTemp
   ' 2.1.1 generate RndIntNo()~U(1,n)
   For J = 1 To 2
    RndIntNo(J) = 1 + Int(Rnd * n)
   Next J
   Do While RndIntNo(1) = RndIntNo(2)
```

```
  RndIntNo(2) = 1 + Int(Rnd * n)
Loop

' swith the sequence/position of these two jobs
JobTemp(1) = Job(RndIntNo(1))
JobTemp(2) = Job(RndIntNo(2))
Job(RndIntNo(1)) = JobTemp(2)
Job(RndIntNo(2)) = JobTemp(1)

' 2.1.2 calculate the new Energy
TimeLength = ProcTime(Job(1))
TardyTemp = TimeLength - DueDate(Job(1))
If TardyTemp < 0 Then
  TardyTemp = 0
End If
Tardy = TardyTemp
Setup = 0

For A = 1 To n - 1
  TimeLength = TimeLength + SetupTime(Job(A), Job(A + 1)) + ProcTime(Job(A + 1))
  TardyTemp = TimeLength - DueDate(Job(A + 1))
  If TardyTemp < 0 Then
    TardyTemp = 0
  End If
  Tardy = Tardy + TardyTemp
  Setup = Setup + SetupTime(Job(A), Job(A + 1))
Next A

EnergyNew = TradeOffPar * Tardy + (1 - TradeOffPar) * Setup
EnergyChange = EnergyNew - Energy

' 2.1.3 if downhill move, accept new state
' 2.1.4 else calculate Boltzmann probability
If EnergyChange <= 0 Then
  Energy = EnergyNew
Else
  BoltzProb = (EnergyChange / TNow)
  ' add a check to ensure BoltzProb doesn't go beyond its range
  If BoltzProb > 200 Then
    BoltzProb = 0
  Else
    BoltzProb = 1 / (1 + 2.718281828 ^ BoltzProb)
  End If
  BoltzRnd = Rnd

  ' 2.1.4.1 if BoltzProb (small at low T) > BoltzRnd, accept change
  ' 2.1.4.2 else return to original sequence at step 2.1.1
  If BoltzProb > BoltzRnd Then
    Energy = EnergyNew
  Else
    Job(RndIntNo(1)) = JobTemp(1)
    Job(RndIntNo(2)) = JobTemp(2)
  End If
```

```
      End If
    Next I
    ' end of 2.1 perform the loop L times

    ' write Energy to file & reduce T
    Print #3, TNow; ","; Energy
    TNow = TNow * TempRate
    NoOfIter = NoOfIter + 1

  Loop
  ' end of do while loop that checks TNow > TFinal

 ' write final sequence & close output file that tracks energy
 Print #3, "Total Iterations = "; NoOfIter
 For I = 1 To n
  Print #3, Job(I); ",";
 Next I
 Print #3,
 Print #3, "Final Energy "; ","; Energy
 Close #3
 Print K1 & K2 & K3 & K4 & ".sim"

Next K4
Next K3
Next K2
Next K1

End Sub

Sub txtTempRate_Change ()
 TempRate = Val(txtTempRate.Text)
End Sub

Sub txtTFinal_Change ()
 TFinal = Val(txtTFinal.Text)
End Sub

Sub txtTmax_Change ()
 Tmax = Val(txtTmax.Text)
End Sub
```

‘ Note: This is the Microsoft Visual Basic Code for the 10-job problems.
‘ For the 30 & 50-job problems, the following modifications are needed:
‘ (i) change data file name
‘ (ii) change n = 30 or 50

LIST OF REFERENCES

REFERENCES

Amar, A.D. and Gupta, J.N.D., 1986, Simulated Versus Real Life Data in Testing the Efficiency of Scheduling Algorithms. *IIE Transactions*, **18** (1), 16-25.

Baker, K.R., and Schrage, L., 1978, Finding An Optimal Sequence by Dynamic Programming: An Extension to Precedence-Related Tasks. *Operations Research*, **26** (3), 111-120.

Barnes, J.W., and Vanston, L.K., 1981, Scheduling Jobs With Linear Delay Penalties And Sequence Dependent Setup Costs. *Operations Research*, **29** (1), 146-159.

Berry, D.A., and Lindgren, B.W., 1990, *Statistics: Theory and Methods* (Pacific Grove, CA: Brooks/Cole Publishing Company).

Blackstone, J.H., Phillips, D.T., and Hogg, G.L., 1982, A State-of-The-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations. *International Journal of Production Research*, **20** (1), 27-45.

Box, G.E.P., and Cox, D.R, 1964, An Analysis of Transformations. *Journal of the Royal Statistical Society*, B(26) 211-243.

Brown, D.E, Huntley, C.L., Markowicz, B.P., and Sappington, D.E., 1992, Rail Network Routing and Scheduling Using Simulated Annealing. *IEEE International Conference On Systems, Man and Cybernetics*, **1**, 588-592.

Conway, R.W., Maxwell, W.L., and Miller, L.W., 1967, *Theory of Scheduling* (Reading, MA: Addison-Wesley).

Corwin, B.D. and Esogbue, A.O., 1974, Two Machine Flow Shop Scheduling Problems With Sequence Dependent Setup Times: A Dynamic Programming Approach. *Naval Research Logistics Quarterly*, **21** (3), 515-523.

Day, J.E. and Hottenstein, M.P., 1970, A Review of Sequencing Research. *Naval Research Logistics Quarterly*, **7**, 11-39.

Driscoll, W.C. and Emmons, H., 1977, Scheduling Production on One Machine With Changeover Costs. *AIIE Transactions*, **9** (4), 388-395.

Emmons, H., 1969, One Machine Sequencing To Minimize Certain Functions Of Job Tardiness. *Operations Research*, **17**, 701-715.

Fisher, M.L., 1976, A Dual Algorithm for The One Machine Scheduling Problem. *Mathematical Programming*, **11**, 229-251.

Gavett, J.W., 1965, Three Heuristic Rules For Sequencing Jobs To A Single Production Facility. *Management Science*, **11** (8), B166-B176.

Graves, S.C., 1981, A Review of Production Scheduling. *Operations Research*, **29** (4), 646-675.

Gupta, S.K., 1982, n jobs And m Machines Job Shop Problem With Sequence Dependent Setup Times. *International Journal of Production Research*, **20** (5), 643-656.

Hax, A.C., and Candea, D., 1984, *Production And Inventory Management* (Englewood Cliffs, NJ: Prentice-Hall, Inc.).

Haynes, R.D., Komar, C.A., and Byrd, J., 1973, The Effectiveness of Three Heuristic Rules For Job Sequencing In A Single Production Facility. *Management Science*, **19** (5), 575-580.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C., 1989, Optimization By Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning. *Operations Research*. **37** (6), 865-892.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., 1983, Optimization By Simulated Annealing. *Science*, **220**, 671-680.

Laursen, P.S., 1993, Theory And Methodology: Simulated Annealing For The QAP - Optimal Tradeoff Between Simulation Time And Solution Quality. *European Journal Of Operational Research*, **69**, 238-243.

Lin, S., and Kernighan, B.W., 1973, An Effective Heuristic Algorithm For The Traveling Salesman Problem, *Operations Research*, **21**, 498-516.

Lockett, A.G. and Muhlemann, A.P., 1972, Technical Notes: A Scheduling Problem Involving Sequence Dependent Changeover Times. *Operations Research*, **20** (4), 895-902.

Montgomery, D.C, 1984, *Design And Analysis of Experiments: Second Edition* (New York, NY: John Wiley & Sons).

Neter, J., Wasserman, W., and Kutner, M.H., 1989, *Applied Linear Regression Models* (Homewood, IL: Richard D. Irwin).

Norusis, M.J., 1993a, *SPSS for Windows - Base System User's Guide Release 6.0* (Chicago, IL: SPSS Inc.).

Norusis, M.J., 1993b, *SPSS for Windows - Advanced Statistics Release 6.0* (Chicago, IL: SPSS Inc.).

Otten, R.H.J.M., and van Ginneken, L.P.P.P., 1989, *The Annealing Algorithm* (Norwell, MA: Kluwer Academic Publishers).

Panwalkar, S.S., Dudek, R.A., and Smith, M.L., 1973, Sequencing Research And The Industrial Scheduling Problem. *Symposium on The Theory of Scheduling And Its Applications*, Beckmann, M., Goos, P.G., and Zurich, H.P.K., ed., 29-38.

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

Picard, J.C., and Queyranne, M., 1976, The Time Dependent Traveling Salesman Problem And Applications to The Tardiness Problem in One Machine Scheduling. *Paper EP76-R-14*, Dept de Genie Industriel, Ecole Polytechnique de Montreal.

Potts, C.N., and Van Wassenhove, L.N., 1985, A Branch And Bound Algorithm For The Total Weighted Tardiness Problem. *Operations Research*, **33** (2), 363-377.

Potts, C.N., and Van Wassenhove, L.N., 1987, Dynamic Programming And Decomposition Approaches For The Single Machine Total Tardiness Problem. *European Journal of Operational Research*, **32**, 405-414.

Potts, C.N., and Van Wassenhove, L.N., 1991, Single Machine Tardiness Sequencing Heuristics. *IIE Transactions*, **23** (4), 346-354.

Prabhakar, T., 1974, A Production Scheduling Problem With Sequencing Considerations. *Management Science*, **21** (1), 34-42.

Ragatz, G.L., 1993, A Branch And Bound Method For Minimum Tardiness Sequencing On A Single Processor With Sequence Dependent Setup Times. *Proceedings: Twenty-fourth Annual Meeting of The Decision Sciences Institute*, 1375-1377.

Rinnooy Kan, A.H.G., Lageweg, B.J., and Lenstra, J.K., 1975, Minimizing Total Costs in One Machine Scheduling. *Operations Research*, **23** (5), 908-927.

Rubin, P.A. and Ragatz, G.L., 1995, Scheduling In A Sequence Dependent Setup Environment With Genetic Search. *Computers & Operations Research*, 22(1), 85-99.

Schrage, L. and Baker, K.R., 1978, Dynamic Programming Solution of Sequencing Problems With Precedence Constraints. *Operations Research*, **26** (3), 444-449.

Shang, J.S., 1993, Theory And Methodology: Multicriteria Facility Layout Problem: An Integrated Approach. *European Journal Of Operational Research*, **66**, 291-304.

Srikar, B.N. and Ghosh, S., 1986, A MILP Model for The n-job, M-stage Flowshop With Sequence Dependent Set-up Times. *International Journal of Production Research*, **24** (6), 1459-1474.

Suresh, G., and Sahu, S., 1993, Multi-objective Facility Layout Using Simulated Annealing. *International Journal of Production Economics*, **32** (2), 239-254.

Van Laarhoven, P.J.M., Aarts, E.H.L., and Lenstra, J.K., 1992, Job Shop Scheduling By Simulated Annealing. *Operations Research*, **40** (1), 113-125.

White, C.H. and Wilson, R.C., 1977, Sequence Dependent Set-up Times And Job Sequencing. *International Journal of Production Research*, **15** (2), 191-202.