LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
OCT 0 1 2014 2011		

MSU Is An Affirmative Action/Equal Opportunity Institution
c:\circ\datedue.pm3-p.1

AN EXP
OPERA

AN EXPERIMENTAL INVESTIGATION OF INFORMATION USE IN A JOB SHOP OPERATING UNDER DUAL RESOURCE CONSTRAINTS: A SIMULATION STUDY

Volume I

By

Lawrence Dean Fredendall

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Department of Management

1991

AN EXPE
OPERAT

Si

operatio

isolated

tions ma

have bee

that sche

Th

of inform

decrease

the job s

literature

explored

research e

system on

areas are

The

ship betwe

examine w

ABSTRACT

AN EXPERIMENTAL INVESTIGATION OF INFORMATION USE IN A JOB SHOP OPERATING UNDER DUAL RESOURCE CONSTRAINTS: A SIMULATION STUDY

By

Lawrence Dean Fredendall

Since 1967, scheduling has been one of the primary research areas of operations management. Most of this research has studied scheduling as an isolated activity, concentrating on the dispatching rule. Other areas of operations management such as the planning system and order review/release (ORR) have been ignored. However, many operations managers are now recognizing that scheduling is influenced by the other parts of the manufacturing system.

The manufacturing system is linked together and coordinated by the flow of information. The available information can be used to simplify scheduling and decrease operating costs. Research about the role of information in scheduling the job shop has begun in the ORR and the dual resource constrained (DRC) literature. In addition to investigating information usage, the ORR research explored methods of simplifying the shop floor dispatching decision. The research explored load leveling in the shop and the impact of the planning system on the shop floor control (SFC) activities. The primary DRC research areas are information use in the labor assignment and job dispatching decisions.

The two primary purposes of this research are: 1) to examine the relationship between the planning system and scheduling in the DRC shop; and, 2) to examine what information should be used by the SFC system to schedule

resource

new "wi

these ru

planning

T

of the av

nant of p

decision

discovere

informati

to be the

patching

cant imp

resources. To do this, the research proposes and tests two new ORR rules, a new "when" labor assignment rule, and a new job dispatching rule. Each of these rules investigates the use of information and is tested with two different planning systems.

The data was analyzed using ANOVA and pair-wise multiple comparisons of the average total cost per week. The analysis found that the major determinant of performance was the choice of the planning system, and that the ORR decision was improved by using additional information. However, it was discovered that how the information was used was more important than what information was used. A heuristic to conduct finite loading of the shop proved to be the superior rule for most of the treatment combinations. The job dispatching rules and the labor assignment rules were found to have little significant impact on shop performance when the ORR rules were used.

Copyright by

LAWRENCE DEAN FREDENDALL

1991

DEDICATED TO MY WIFE

ELAINE LENORE MALINOWSKI FREDENDALL

THANK YOU

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to the members of my dissertation committee, Dr. Steven A. Melnyk, Dr. Gary L. Ragatz, Dr. Soumen Ghosh, and Dr. Paul Rubin for their assistance, guidance and support in the preparation of this research. This dissertation could not have been completed without their contributions.

The Chairman of the Dissertation Committee, Dr. Steven A. Melnyk, deserves special thanks for his patient mentoring over the years. His counsel, support and guidance were all extremely valuable. Without his continuing interest, meticulous editing, his suggestions and his belief in my ability, this work would not have been possible.

Dr. Gary Ragatz generously took time to discuss this research with me whenever necessary. He offered important suggestions throughout the experiment and contributed greatly to my ability to complete the work.

Dr. Paul Rubin was of tremendous assistance to me in developing the optimizing algorithms and statistical tests. His depth of knowledge in the methodology and analysis areas was an asset to the work.

D

extreme

provided

I a

support

understa

at work.

play wit

N

to com

staff ar

Depart

contri

Steve

Mark

Mana

Dr. Soumen Ghosh provided valuable comments and suggestions. He was extremely willing to endure countless interruptions of his own work to help. He provided insight and encouragement during this work.

I am extremely grateful to my wife, Elaine Malinowski, for her continued support over the many years of my study. Hopefully my daughter Susan will understand someday why I had to spend so much time at the computer and away at work. And finally I am grateful to my son Joseph for waiting so patiently to play with me.

Many other individuals have contributed in significant ways to my ability to complete this dissertation. This includes other members of my family, the staff and faculty in the College of Business and my fellow Ph.D. students in the Department of Management at Michigan State University. Individuals who contributed include Vijay Kannan, Byung-kyu Sohn, David Mendez, Paul Blossom, Steve Lyman, Joe Carter, Mike Moch, Greg Magnan, Bret Wagner, Laura Ghosh, Mark Kielhorn, Faye Backie, Steve Szylvagi, Art Saltzman and Carolyn Mac-Manaman.

LIST OF
LIST OF
KEY TO
CHAPTE
1
1
CHAPT

TABLE OF CONTENTS

LIST OF TABLES xiii

LIST OF FIGURES xvi

KEY TO ABBREVIATIONS xxi

CHAPTER 1 1
1.1 INTRODUCTION 1
1.1.1 REEXAMINING THE SCHEDULING PROCESS 3
1.2 SCHEDULING THE DRC: BACKGROUND TO THE STUDY 3
1.2.1 THE PLANNING SYSTEM AND THE DRC SHOP 4
1.2.2 ORR AND THE DRC SHOP 6
1.2.3 INFORMATION AND THE DRC SHOP 10
1.3 RESEARCH QUESTIONS 12
1.4 RESEARCH METHODOLOGY 13
1.5 CONTRIBUTIONS OF RESEARCH 15
1.6 SUMMARY 16
1.7 ORGANIZATION OF DISSERTATION 19
1.8 CONCLUDING COMMENTS 19

CHAPTER 2 21
2.1 INTRODUCTION TO LITERATURE REVIEW 21
2.2 ORDER REVIEW/RELEASE 21
2.2.1 DEFINITION OF TERMS 22
2.2.2 STRUCTURE OF ORR LITERATURE REVIEW 27
2.2.3 EXPERIMENTAL RESEARCH 28
2.2.3 APPLIED RESEARCH 44
2.2.4 DESCRIPTIVE RESEARCH ABOUT I/O CONTROL 46
2.2.3 SUMMARY OF ORR RESEARCH 53
2.3 DUAL RESOURCES CONSTRAINED LITERATURE REVIEW 54
2.3.1 EXPERIMENTAL RESEARCH 57
2.3.5 SUMMARY OF DRC RESEARCH 73
2.4 SUMMARY OF LITERATURE REVIEW 74

CHAPTE
3
3
3
3.
3
3
3
3
CHAPTE
4
4
4
4.

CHAPTER 3 79
3.1 INTRODUCTION TO THE RESEARCH METHODOLOGY 79
3.2 JUSTIFICATION OF RESEARCH METHODOLOGY 80
3.3 EXPERIMENTAL VALIDITY 81
3.3.1 STATISTICAL CONCLUSION VALIDITY 81
3.3.2 INTERNAL VALIDITY 86
3.3.3 CONSTRUCT VALIDITY 87
3.3.4 EXTERNAL VALIDITY 87
3.4 DESCRIPTION OF THE SHOP ENVIRONMENT 89
3.4.1 KEY DEFINITIONS 89
3.4.2 RESEARCH ASSUMPTIONS 91
3.4.3 SHOP OPERATING RULES 92
3.4.3.1 ORDER CREATION IN THE PLANNING SYSTEM 93
3.5 PROCESSING OF THE SIMULATION CODE 96
3.5.1 X() ARRAY 98
3.5.2 COMMON LOGIC IN ALL EVENTS 99
3.5.3 ARRIVE EVENT 99
3.5.4 DEPART EVENT 101
3.5.5 SHIFT EVENT 101
3.5.6 EXIT EVENT 102
3.6 SIMULATION DESIGN ISSUES 102
3.6.1 VERIFICATION 103
3.6.2 INITIALIZATION BIAS 112
3.6.3 BATCH SIZE 113
3.6.6 SUMMARY 116
3.7 PERFORMANCE MEASURES 117
3.7.1 SECONDARY PERFORMANCE MEASURES 119
3.8 DATA COLLECTION 120
3.9 SUMMARY 121

CHAPTER 4 122
4.1 INTRODUCTION TO EXPERIMENTAL DESIGN 122
4.2 EXPERIMENTAL VARIABLES 122
4.2.1 THE TYPE OF PLANNING SYSTEM 122
4.2.2 ORDER REVIEW/RELEASE RULES 124
4.2.3 "WHEN" LABOR ASSIGNMENT MECHANISMS 131
4.2.4 JOB DISPATCHING RULES 133
4.3 A PRIORI HYPOTHESES 135
4.4 EXPERIMENTAL DESIGN 141

4.5 PILOT RUNS 142
4.5.1 COMPLEX ALGORITHM 142
4.5.2 PARAMETER SELECTION 147
4.6 SUMMARY 148

CHAPTER 5 154
5.1 INTRODUCTION TO EXPERIMENTAL RESULTS 154
5.1.1 OVERVIEW OF ANALYSIS 169
5.1.2 OVERVIEW OF STATISTICAL CALCULATIONS 170
5.2 ANALYSIS OF FIT OF ANOVA MODEL TO COST DATA 172
5.2.1 CHECK RESIDUALS FOR NORMALITY 173
5.2.3 CHECK RESIDUALS FOR HOMOGENEITY 174
5.2.4 SELECT APPROPRIATE TRANSFORMATION 174
5.3 ANALYSIS OF RECIPROCAL COST DATA 186
5.3.1 FIT OF RECIPROCAL DATA TO ANOVA MODEL 187
5.3.2 PAIR-WISE MULTIPLE COMPARISONS 188
5.3.3 EXAMINATION OF THE FOUR WAY INTERACTION . . 193
5.4 INVESTIGATION OF A PRIORI HYPOTHESES 206
5.4.1 HYPOTHESIS 1 215
5.4.2 HYPOTHESIS 2 215
5.4.3 HYPOTHESIS 3 217
5.4.4 HYPOTHESIS 4 218
5.4.5 HYPOTHESIS 5 219
5.4.6 HYPOTHESIS 6 219
5.4.7 HYPOTHESIS 7 230
5.4.8 HYPOTHESIS 8 231
5.4.9 HYPOTHESIS 9 232
5.4.10 Hypothesis 10 233
5.4.11 SUMMARY OF HYPOTHESES EVALUATION 234
XXXX
5.5 POST HOC ANALYSIS 253
5.5.1 CMS AND MLC ORR RULES 255
5.5.2 JOB SELECTION MECHANISM 267
5.5.3 PLANNING SYSTEM 268
5.5.4 ORR RULES AND PERFORMANCE 269
5.5.5 JOB DISPATCHING AND ORR RULES 270
5.5.6 "WHEN" ASSIGNMENT AND ORR RULES 271
5.5.7 SIGNIFICANCE OF JOB DISPATCHING 272
5.5.8 SUMMARY OF POST HOC ANALYSIS 279

5.6 SENSITIVITY ANALYSIS 280
5.6.1 ESTIMATION OF ORR PARAMETERS 281
5.6.2 COST STRUCTURE 294
5.7 SUMMARY 295

CHAPTER 6 298
6.1 ANALYSIS AND CONCLUSIONS: INTRODUCTION 298
6.2 INFORMATION USE IN SFC 298
6.2.1 INFORMATION AND THE JOB DISPATCH RULES 299
6.2.2 "WHEN" ASSIGNMENT AND INFORMATION 300
6.2.3 INFORMATION AND THE ORR RULE 302
6.3 THE PLANNING SYSTEM 311
6.4 IMPLICATIONS OF SENSITIVITY ANALYSIS 314
6.4.1 COST STRUCTURE AND PLANNING 314
6.4.2 COST STRUCTURE AND ORR RULE 315
6.4.3 COST STRUCTURE AND "WHEN" ASSIGNMENT 316
6.5 IMPLICATIONS FOR SCHEDULING THEORY 317
6.6 FUTURE RESEARCH DIRECTIONS 320
6.7 CONCLUSION 323

APPENDIX A 326
ATTRIBUTE STRUCTURE AND VARIABLE DICTIONARY 326
PART 1: ATTRIBUTE STRUCTURE 326
PART 2: VARIABLE DICTIONARY 327

APPENDIX B 331
SIMULATION CODE 331
PART 1: SIMAN EXPERIMENT PROCESSOR CODE 331
PART 2: FORTRAN CODE FOR USER WRITTEN SUBROUTINES . . . 333

APPENDIX C 394
FORTRAN CODE FOR SETTING OF SAMPLING PROCEDURES 394
APPENDIX D 418
ILLUSTRATION OF MLC JOB SELECTION MECHANISM 418

APPEND
R

LIST OF

APPENDIX E 421
RATIO OF VARIANCES BY CELLS 421

LIST OF REFERENCES 434

LIST OF TABLES

2-1 SUMMARY OF ORR SINGLE MACHINE RESEARCH 29

2-2 SUMMARY OF ORR EXPERIMENTAL LITERATURE 32

2-3 SUMMARY OF ORR EXPERIMENT WITH MULTIPLE MACHINES 37

2-4 SUMMARY OF ORR EXPERIMENTS IN DRC SHOPS 42

2-5 SUMMARY OF ORR APPLICATIONS LITERATURE 43

2-6 SUMMARY: DESCRIPTIVE I/O CONTROL RESEARCH 47

2-7 SUMMARY OF DRC ANALYTICAL RESEARCH 58

2-8 SUMMARY 1: OPERATING ISSUES IN THE DRC SHOP 61

2-9 SUMMARY 2: OPERATING ISSUES IN DRC SHOP 62

2-10 SUMMARY OF DRC ASSEMBLY SHOP 65

2-11 SUMMARY OF DRC SHOP DESIGN 69

2-12 SUMMARY OF DRC IMPLEMENTATION LITERATURE 70

2-13 SUMMARY OF DRC DESCRIPTIVE RESEARCH LITERATURE 71

2-14 INFORMATION CONTENT OF SHOP FLOOR CONTROL 78

3-1 RUNS TO DETERMINE SAMPLING CONDITIONS 83

3-2 VERIFICATION RUNS OF SIMULATION 105

3-3 COMPARISON OF SIMULATION TO THEORETICAL RESULTS 108

3-4 COMPARISON OF THIS SIMULATION TO RAGATZ (1985) 109

3-5 COMPARISON OF PARK (1987) TO THIS MODEL 111

4-1 CELLS IN MAIN EXPERIMENT: LABELED BY ROW, COLUMN 137

4-2 PARAMETERS IN EACH TREATMENT CELL 153

5-1 MEAN TOTAL COST/WEEK AND STANDARD DEVIATION 155

5-2 MEAN HOURS TARDY PER WEEK 156

5-3 MEAN HOURS TOTAL INVENTORY PER WEEK 157

5-4 MEAN NUMBER OF LABOR TRANSFERS PER WEEK 158

5-5 MEAN HOURS TARDY PER TARDY JOB 159

5-6 MEAN 90th PERCENTILE OF TARDINESS 160

5-7 MEAN FLOW TIME PER JOB* 161

5-8 MEAN 90th PERCENTILE OF FLOW TIME 162

5-9 MEAN TIME IN QUEUE ON SHOP FLOOR PER JOB 163

5-10 MEAN TIME IN ORDER POOL PER JOB 164

5-11 MEAN TIME IN SYSTEM PER JOB 165

5-12 MEAN PERCENT JOBS TARDY PER WEEK 166

5-13 MEAN HOURS FINISHED GOODS INVENTORY/WEEK 167

5-14 MEAN WORK IN PROCESS INVENTORY/WEEK 168

5-15 ANOVA TABLE: DEPENDENT VARIABLE IS COST 175

5-16 SHAPIRO-WILKS W AND THE (KURTOSIS) FOR COST 176

5-17 MEANS AND STANDARD DEVIATION OF RECIPROCAL OF COST . . 179

5-18 SHAPIRO-WILKS W AND KURTOSIS FOR RECIPROCAL OF COST . . 180

5-19 ANOVA TABLE: FRIEDMAN TEST 191

5-20 ANOVA TABLE: RECIPROCAL COST DATA 192

5-21 ANOVA TABLE: RECIPROCAL HOURS TARDY 208

5-22 ANOVA TABLE: RECIPROCAL TOTAL INVENTORY 209

5-23 ANOVA TABLE: RECIPROCAL LABOR TRANSFERS 210

5-24 SUMMARY OF ANOVA FOR TOTAL COST AND COMPONENTS 214

5-25 SUMMARY OF A PRIORI HYPOTHESES 252

5-26 PERFORMANCE MEASURE SUMMARY 273

5-27 COST RATIOS FOR SENSITIVITY ANALYSIS 281

LIST OF FIGURES

1-1 MANUFACTURING PLANNING AND CONTROL SYSTEM 5

1-2 SHOP FLOOR CONTROL: AN INTEGRATING FRAMEWORK 8

1-3 ILLUSTRATION OF DRC JOB SHOP 9

1-4 SOURCES OF INFORMATION FOR SFC SYSTEM 11

2-1 DEFINITIONS OF TERMS 24

2-2 ORDER REVIEW/RELEASE FUNCTION 26

2-3 CLOSED PLANNING & SFC SYSTEM 41

2-4 TAXONOMY OF ORR LITERATURE 55

2-5 ILLUSTRATION OF A DUAL RESOURCE CONSTRAINED SHOP 56

2-6 SUMMARY OF ORR RESEARCH IN A DRC SHOP 75

3-1 FLOWCHART OF SIMULATION CODE 97

4-1 COMPLEX ALGORITHM 143

4-2 CELL 2111: COST STRUCTURE 20/1/1 144

4-3 CELL 2112: COST STRUCTURE 20/1/1 145

4-4 CELL 3211: COST STRUCTURE 20/1/1 149

4-5 CELL 3212: COST STRUCTURE 20/1/1 150

4-6 CELL 4311: COST STRUCTURE 20/1/1 151

4-7 CELL 4312: COST STRUCTURE 20/1/1 152

5-1 HOMOGENEITY OF VARIANCE: HARTLEY TEST 182

5-2 HOMOGENEITY OF VARIANCE: HARTLEY TEST: $\alpha = 0.01$ 183

5-3 BONFERRONI COMPARISONS OF RECIPROCAL COST 189

5-4 KRUSKAL-WALLIS TEST OF RECIPROCAL COST 190

5-5 FOUR WAY INTERACTION 195

5-6 FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM 198

5-7 FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM . 199

5-8 FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM 200

5-9 FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM . 201

5-10 FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM 202

5-11 FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM . 203

5-12 COMPARISONS: RECIPROCAL HOURS TARDY 211

5-13 BONFERRONI COMPARISONS: TOTAL INVENTORY RECIPROCAL . . 212

5-14 BONFERRONI COMPARISONS: RECIPROCAL LABOR TRANSFERS . . 213

5-15 HYPOTHESIS 2: TOTAL COST 221

5-16 HYPOTHESIS 2: TOTAL INVENTORY 222

5-17 HYPOTHESIS 2: LABOR TRANSFERS 223

5-18 HYPOTHESIS 3: TOTAL COST 224

5-19 HYPOTHESIS 3: TOTAL INVENTORY 225

5-20 HYPOTHESIS 3: LABOR TRANSFERS 226

5-21 HYPOTHESIS 4: TOTAL COST 227

5-22 HYPOTHESIS 5: TOTAL COST 228

5-23 HYPOTHESIS 6: TOTAL COST 229

5-24 HYPOTHESIS 7: TOTAL COST 236

5-25 HYPOTHESIS 7: HOURS TARDY/WEEK 237

5-26 HYPOTHESIS 7: TOTAL INVENTORY 238

5-27 HYPOTHESIS 7: LABOR TRANSFERS 239

5-28 HYPOTHESIS 8: TOTAL COST 240

5-29 HYPOTHESIS 8: HOURS TARDY/WEEK 241

5-30 HYPOTHESIS 8: TOTAL INVENTORY 242

5-31 HYPOTHESIS 8: LABOR TRANSFERS 243

5-32 HYPOTHESIS 9: TOTAL COST 244

5-33 HYPOTHESIS 9: HOURS TARDY PER WEEK 245

5-34 HYPOTHESIS 9: TOTAL INVENTORY 246

5-35 HYPOTHESIS 9: LABOR TRANSFERS 247

5-36 HYPOTHESIS 10: TOTAL COST 248

5-37 HYPOTHESIS 10: HOURS TARDY PER WEEK 249

5-38 HYPOTHESIS 10: TOTAL INVENTORY 250

CHAPTER 1

1.1 INTRODUCTION

Scheduling has traditionally been one of the primary research areas of operations management. In their survey of this field, Melnyk, Carter, Dilts and Lyth (1985, pp. 222-282) noted over 1200 articles about scheduling. All of these have been published since 1967 when Conway, Maxwell and Miller (1967) completed the first major study about scheduling.

In this past research, scheduling was treated as an isolated activity. The other areas of operations management such as the planning system, order review/release (ORR), lot sizing, and subcontracting were ignored or treated as exogenous. Consequently, the academic researcher was left with the dispatching rule as the only control parameter in the shop as noted by Eilon, Chowdhury and Serghiou (1975, p.45):

> If the arrival of jobs, their processing requirements, and the operating facilities are given, the only control parameter at the disposal of the dispatcher is ... the order in which jobs should be processed.

However, researchers have now recognized that scheduling is not an isolated activity. Operations managers have developed a systems perspective and view scheduling as part of the manufacturing system (Krajewski and Ritzman, 1990, p.25). As a result, it is as important to understand the relationships that exist in the manufacturing system as it is to identify the "best" dispatching rule (Melnyk et al., 1985, pp.158-166).

The key to understanding the relationships in the manufacturing system is a careful analysis of the information about the planning decision, the ORR decisions, the lot sizing and subcontracting decisions. Information is the major component that links these decisions together into a manufacturing system. It is the flow of information that coordinates the planning system and the shop floor control[1] (SFC) system. The SFC system uses this information to monitor and control the progress of the jobs through the shop (Melnyk et al., 1985, p.36). For example, this information can be used by managers to reduce inventory, simplify complex dispatching rules and coordinate multiple resources (McKay, Safayeni and Buzacott, 1988).

As mentioned above, the scheduling decision seldom involves only one resource (e.g. machine capacity). Machines, material labor and tooling may all be limited resources which need to be considered in any allocation decision (Browne, Boon and Davies, 1981).

As a result of these and other developments, it is time to reexamine our approach to scheduling. There is an opportunity to bring a new richness to the research. New areas of scheduling can be exposed for further study and a different perspective on scheduling can be developed.

---

[1]. Shop floor control is defined as "that group of activities directly responsible for managing the transformation of planned orders into a set of outputs" (Melnyk et al., 1985, p.35).

1.1.1 REEXAMINING THE SCHEDULING PROCESS

This study was placed in a DRC job shop where the constraining resources were labor and machines. The two primary purposes of this study were:

- To examine the relationship between the planning system and the scheduling system in the DRC job shop. There were two goals. First, to examine how controlling the flow of work at the planning stage impacted shop performance. Second, to explore the interaction of the SFC decisions with the planning system.
- To explore the role of information in scheduling the shop floor resources. This includes exploring what information to use and how to use this information in scheduling.

The remainder of this chapter will briefly summarize the prior research about the DRC shop, the planning system, the use of ORR rules and finally, the use of information.

1.2 SCHEDULING THE DRC: BACKGROUND TO THE STUDY

The two major sources of prior research are the DRC and ORR literature. A brief overview this literature is provided as background for further discussion. In the DRC literature the constrained resources are usually labor and machine capacity. Previous studies in the DRC shop environment (e.g. Nelson, 1967; Fryer, 1973; Weeks, 1977; and, Maggard, Hogg and Phillips, 1980) primarily explored the effectiveness of different job dispatching rules when used with

different labor assignment rules. This traditional DRC research was conducted in a variety of shops. These shops had different levels of labor constraints (labor efficiency at various machines, labor capacity, and delays in transferring operators), alternate job routings, different shop structures and how the use of information impacted shop performance (Treleven, 1989).

Three different issues that were not fully emphasized in the prior DRC research are developed and fully integrated in this research. They are:

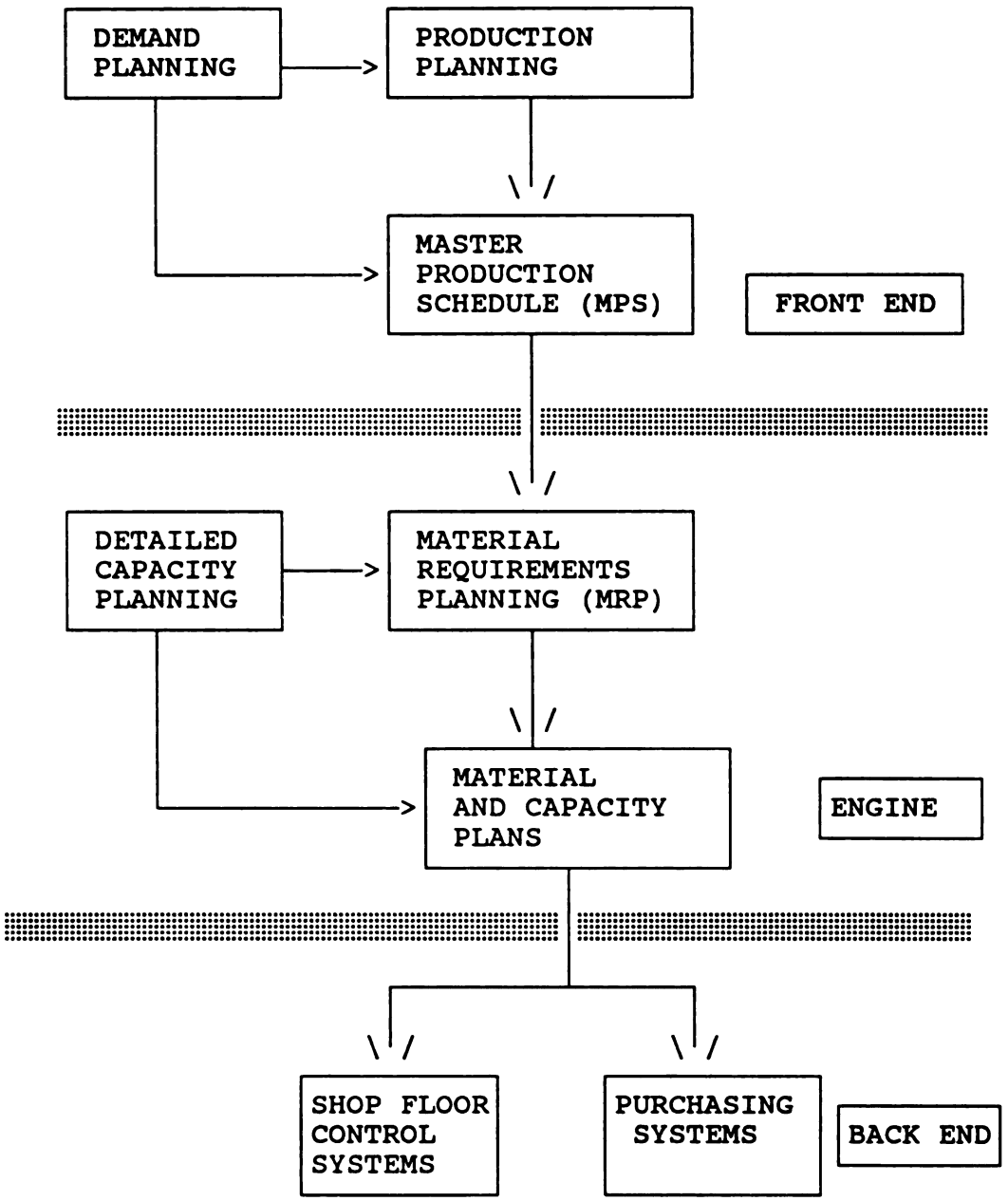
- the role of the planning system in a DRC shop.
- the use of ORR mechanisms as part of the SFC system in a DRC shop.
- the role of information in controlling a DRC shop.

Each of these issues is now discussed in more detail below.

1.2.1 THE PLANNING SYSTEM AND THE DRC SHOP

The planning system consists of all activities that occur prior to the release of an order to the shop floor. These activities are summarized in the framework developed by Vollmann, Berry and Whybark (1988, p.16) as illustrated in Figure 1-1. Within this framework, the planning segment of the manufacturing planning and control system consists of such activities as production planning, master production scheduling, rough cut capacity planning, material and requirements planning and detailed capacity requirements planning. After these

FIGURE 1-1
MANUFACTURING PLANNING AND CONTROL SYSTEM
(From: Vollmann, Berry, Whybark, 1988, p. 16)



planning activities are completed, the set of orders is given to the shop floor or the "Back End" which is responsible for the actual production of the product. From the shop floor's perspective, one of the major effects of the planning system is the variability of the work flow. The planning system could provide the shop a smooth workload over time by pulling work forward or by delaying work. If the planning system did not smooth the workload then there would be a large variance from week to week in the amount of work received by the shop (Kingsman, Tatsiopoulos and Hendry, 1989).

1.2.2 ORR AND THE DRC SHOP

The second issue addressed in this dissertation is the role of ORR rules in scheduling within the DRC shop. ORR is one of the SFC activities. The relationship of ORR rules with the other SFC activities is developed by Melnyk, Carter, Dilts and Lyth (1985, p.40) in their framework shown in Figure 1-2.

This framework shows how information is fed forward from the planning system to the SFC system and how feedback is sent from the shop to the planning system. They found that this "closed loop" model represented the information flows in job shops they studied.

The five distinct but interrelated activities of SFC as defined by Melnyk, et al. (1985) are:

1) Order Review/Release (ORR). The ORR activity documents orders, filters out infeasible orders and controls the release of orders to the shop to keep the load level.

2) Detailed Assignment. The detailed assignment activity allocates available resources (material, labor, tooling, machine capacity) to specific orders.

3) Data Collection and Monitoring. The data collection activity gathers all pertinent information about the orders.

4) Control/Feedback. The control/feedback activity provides information to the detailed assignment activity to allow timely changes.

5) Order Disposition. Finally, the order disposition activity collects the shop performance measures and provides this information to other departments.

This research explores the first two SFC activities in depth. These two activities, the ORR rule and the detailed assignment of resources, are treated as experimental variables in this research.

The ORR function serves as a gateway between the planning system and the shop floor as illustrated in Figure 1-3. All jobs released from the planning system enter the shop through the ORR control point. While the ORR activity involves all the tasks listed above, prior research and this research examines only how the ORR activity controls the level of work in the shop. Once the job is released by the ORR rule to the shop floor, its progress through the shop is

FIGURE 1-2
SHOP FLOOR CONTROL: AN INTEGRATING FRAMEWORK
(Melnyk, Carter, Dilts, Lyth, 1985, p.40)

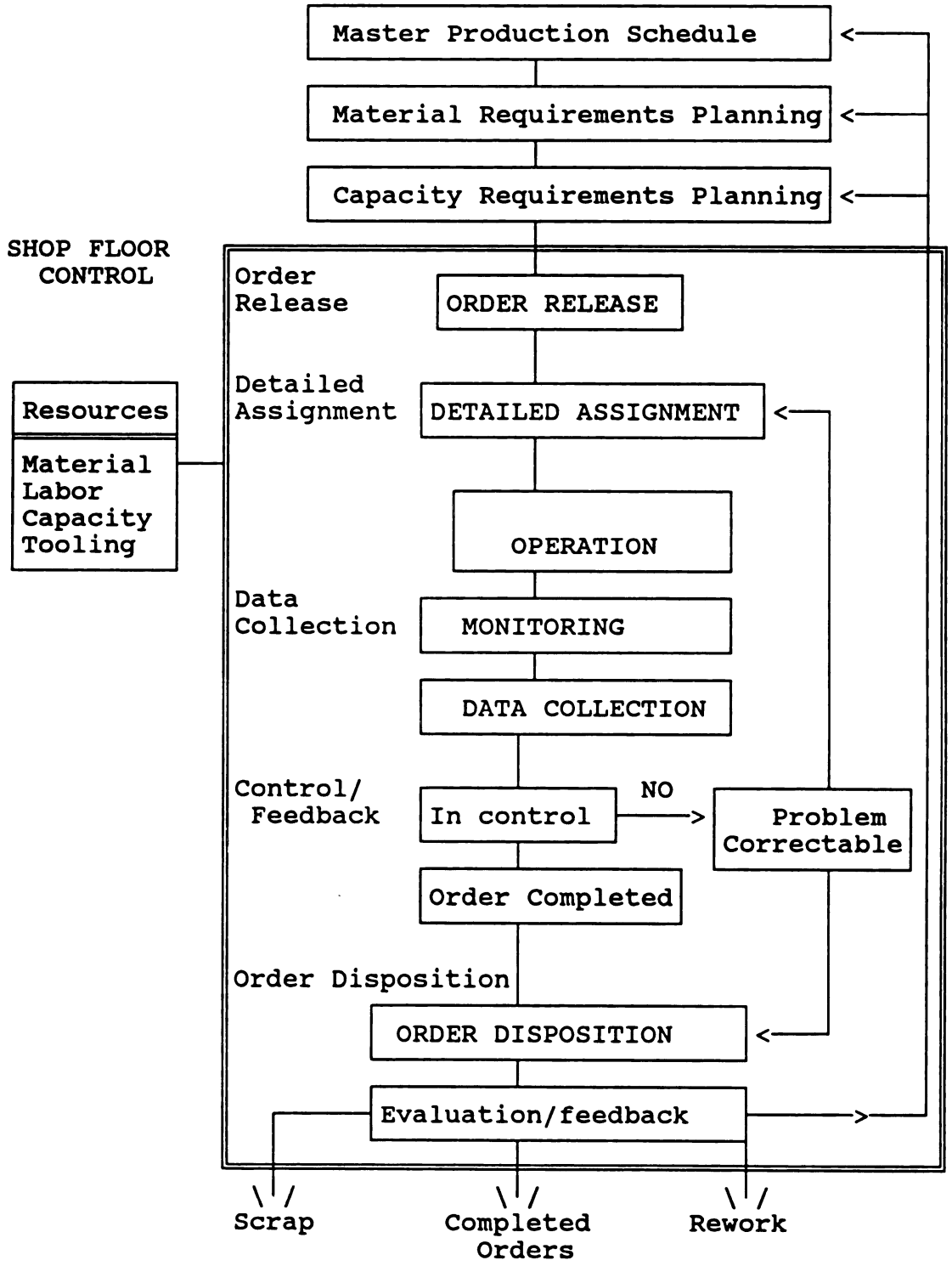
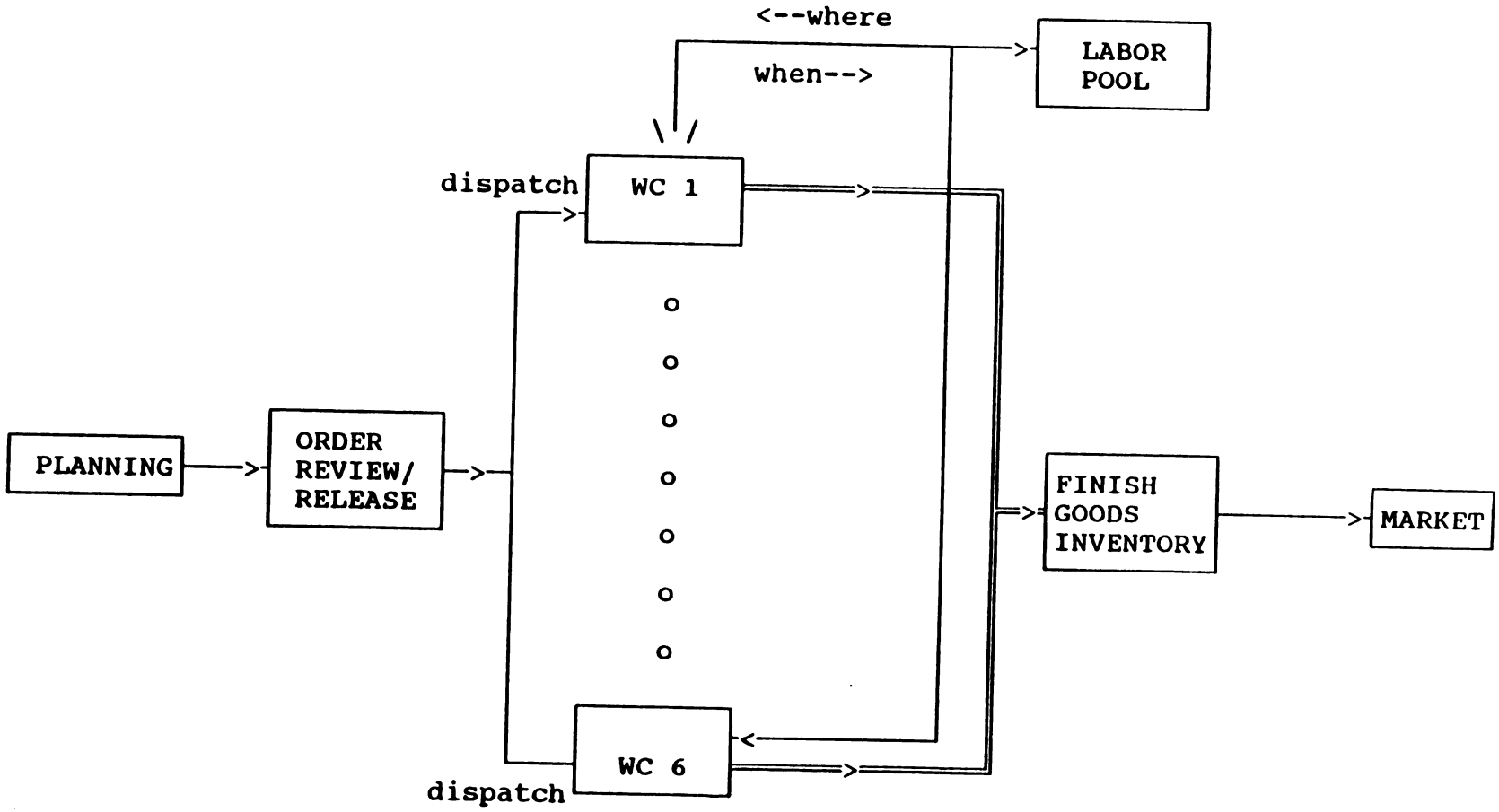


FIGURE 1-3
ILLUSTRATION OF DRC JOB SHOP



controlled by the second SFC activity, detailed assignment, using job dispatching and labor assignment rules.

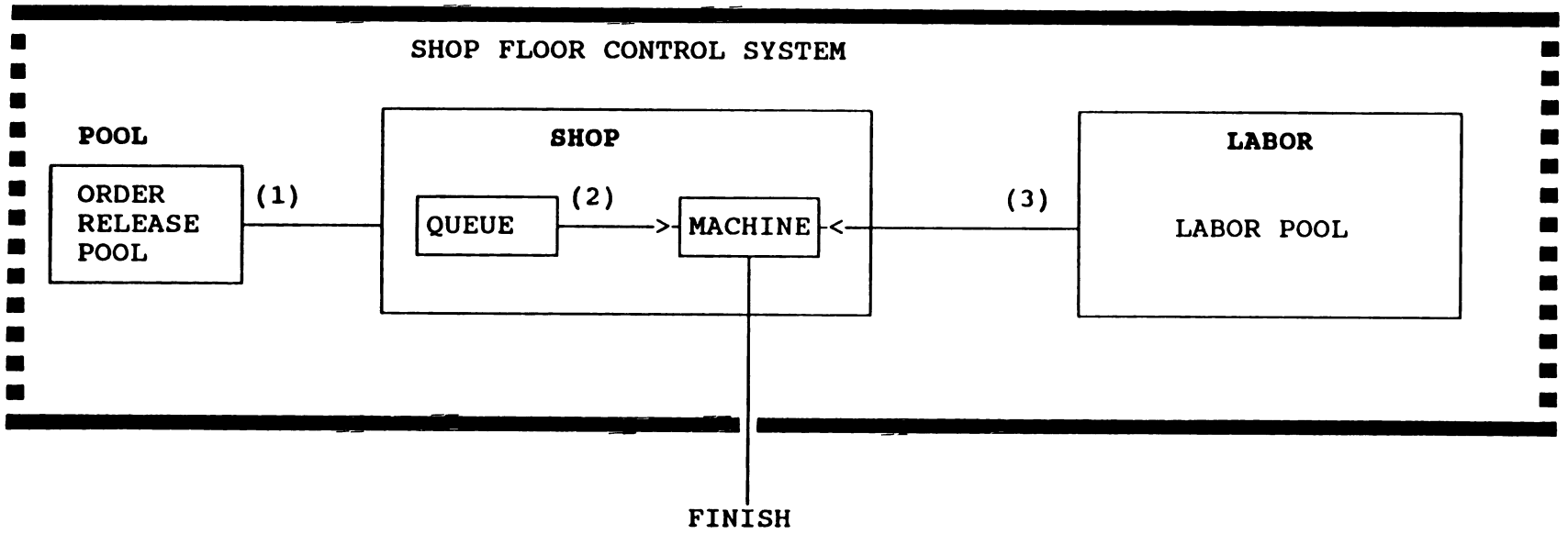
1.2.3 INFORMATION AND THE DRC SHOP

The third issue examined is the role of information in SFC decisions. Researchers have recognized the need to model the flow of information in the shop (McKay, Safayeni and Buzacott, 1988), but to date there has been limited research about the role of information.

As illustrated in Figure 1-4, this research classifies the information in a DRC shop as being either POOL, SHOP or LABOR information. POOL information is any information about jobs in the order pool not yet released to the shop floor. SHOP information is any information about jobs in process on machines or waiting in queues on the shop floor. LABOR information is information about machine operators either in the labor pool or working on machines.

Park and Bobrowski (1989) and Bobrowski and Park (1989) investigated the use of POOL and SHOP information with both infinite and finite loading ORR rules in a DRC shop. Treleven (1987) and Gunther (1979 and 1980) investigated information usage in the labor assignment rules of a DRC shop. Ragatz and Mabert (1988) investigated the impact of using POOL and SHOP information in the ORR rule of a shop with only machine constraints. To date, no studies have examined the impact of using all three sources of information available in a DRC shop.

FIGURE 1-4
SOURCES OF INFORMATION FOR SFC SYSTEM



(1) ORDER REVIEW RELEASE
(2) JOB DISPATCHING
(3) LABOR ASSIGNMENT

This study will be the first to explore how the use of all three types of information (POOL, SHOP and LABOR) effects DRC shop performance. By examining the planning system, the ORR rule and the use of these information sources simultaneously, there is the potential of gaining insight into the operation of the DRC shop. Further, this insight could help to shorten and reduce inventories and other operating costs.

1.3 RESEARCH QUESTIONS

The two primary purposes of this study can be explored further as individual research questions. First, examining the role of information in scheduling the shop floor resources requires answering the following questions.

1. How does the addition of POOL information to the job dispatching rule affect shop performance?
2. How does the addition of POOL information to the "when" labor assignment rule affect shop performance?
3. How does the addition of LABOR information to the ORR rules affect shop performance?
4. How does the use of information in the ORR rules to create a finite schedule when loading the shop affect shop performance?

To address these research questions, the existing heuristics for the job dispatching, "when" labor assignment and ORR decisions were not adequate. None of the existing heuristics used all three information sources. Consequently,

two new ORR rules were developed to incorporate LABOR information. A new job dispatching rule was developed to utilize POOL information. And finally, a new "when" labor assignment rule was developed to use POOL information. These new heuristics were evaluated against the best existing heuristics identified by prior researchers.

The research questions developed to examine the relationship of the planning and scheduling systems and the choice of a planning system were:

1. For a given SFC system, how did the variance in the workload produced by the planning system impact shop performance?
2. For a given planning system how did the use of information to load the shop to a finite capacity impact shop performance?
3. Did the influence of the planning system on the shop depend on the sources of information used in the SFC decisions?

The method used to operationalize this research is now explained.

1.4 RESEARCH METHODOLOGY

This research was conducted using a computer simulation of a general job shop. The simulation was written in SIMAN version 3.5 (Pegden, 1987) with user written fortran subroutines. The data was gathered using simulation because analytical research models have not been able to model the complexity of the DRC shop (Nelson, 1966; Takacs, 1968; Treleven, 1989).

As illustrated in Figure 1-3, the shop was modeled so that the flow of orders went from the planning system to the ORR control point and then to the shop floor. Once orders were released to the shop floor, the sequence in which they were processed at each work center (WC) was determined by the job dispatching rule. However, for the job to be processed it needed both a machine and an operator. There were fewer operators than machines, so labor assignment rules were used to allocate operators from the labor pool to the work centers.

The labor assignment rule actually consists of two component rules. The first is the "when" labor assignment rule, the second is the "where" labor assignment rule. The "when" labor assignment rule determines when a laborer should be removed from one work center. The "where" labor assignment rule determines to which work center a laborer should be assigned (Treleven, 1989).

The experimental variables in the research to be explained in detail later are:

- The planning system.
- The ORR rule.
- The "when" labor assignment rule.
- The job dispatching rule.

The experiment consisted of four phases. Phase 1 included the development, validation and verification of the simulation model. Phase 2 identified the correct parameter settings for the ORR rules used in the research. It also

determined the batch size needed for independence, the batch size needed for normality and the number of samples needed. Phase 3 used a full factorial design for the main experiment. The performance measure in the main experiment was total cost. Phase 4 consisted of sensitivity analysis of the results to the cost structures used in calculating the total cost.

1.5 CONTRIBUTIONS OF RESEARCH

This research extends prior job shop models by continuing the integration of the DRC and ORR literature and by incorporating the planning system into the scheduling research. The research also explicitly examines the impact of using different sources of available information in SFC rules.

This study responds to needs identified by prior researchers. These needs were 1) the examination of additional job dispatching rules in a DRC shop (Treleven, 1989); 2) the impact of using "anticipated workloads at work centers" in SFC rules (Treleven, 1989); 3) the impact of using ORR rules which have "extended visibility" about future workloads (Melnyk and Ragatz, 1988); 4) the impact of the planning system on the SFC system (Melnyk and Ragatz, 1988; Tatsiopoulos and Kingsman, 1983); 5) the need for further investigation of the information used in the ORR rules (Ragatz and Mabert, 1988); and finally, 6) the need to investigate information usage in scheduling (McKay et al., 1988; Browne et al., 1981).

This study extended the work of Ragatz (1985) and Ragatz and Mabert (1988) about the impact of information usage in a machine constrained shop to a DRC shop. Specifically, this research investigates the use of LABOR information in the ORR decision, POOL information in the job dispatching decision and POOL information in the "when" labor assignment rule. In prior research these activities did not have access to those sources of information. Finally, this study examined how variation in the work load received from the planning system influences the performance of the SFC system.

This research increases the knowledge base about the role of information in the SFC decisions and the interaction of the SFC rules with the planning system. Industry practitioners may be able to use this information to improve their flexibility and delivery times while simultaneously reducing costs. It provides practitioners a basis to determine the benefits they can receive from increased information collection and usage. The simultaneous decrease in the costs of computers and the increased sophistication of manufacturing planning and control systems allows practitioners to create "smooth" workloads.

1.6 SUMMARY OF FINDINGS

This is a brief summary of the analysis of the data and the findings. The data was first fitted to the ANOVA model. However the data did not meet the assumptions of the ANOVA model, so the reciprocal of the data was fitted to the

ANOVA model. The reciprocal data satisfied the ANOVA assumptions, and all subsequent analysis was conducted on the transformed data.

The study found:

- That the planning system was a major determinant of DRC shop performance. The SMOOTH planning system always performed significantly better than the UNCONTROLLED planning system regardless of the SFC rules used.
- That the ORR rule had a significant impact on shop performance for all levels of the planning system examined. The best ORR rule was the modified load conversion heuristic developed for this research. This ORR rule loaded the shop to a finite limit.
- That the sub-components of the ORR rule (e.g. the trigger mechanism and the job selection mechanism) each impact shop performance differently. The trigger mechanism influences work in process inventory, time in system measures and the number of labor transfers. The job selection mechanism influences finished goods inventory and the tardiness performance measures.
- That the "when" labor assignment rule was usually not significant. Also, there was no advantage to using POOL information in the "when" labor assignment rule. Under most conditions, where there was a difference between performance of the "when" labor assignment rules, the rule which held an operator at a machine until all

the work in the queue was completed was the best rule. The "when" rule influenced only the number of labor transfers in the shop.

- That there were no significant differences between the job dispatching rules used if ORR rules were also used. In those treatments where the job dispatching rules were significant, the MODD rules were usually the best performing rules, but there was no significant pattern. There was no advantage to using POOL information in the job dispatching rule.

Based on this research it is proposed for further research that the best trigger mechanism would do the following:

- restrict inventory at the gateway work center to zero;
- and allow inventory to build in front of the critical work center.

It was also proposed for future research that the best job selection mechanism would do the following:

- ensure that there was always work to be done at the critical work center by keeping inventory in front of the critical work center and preventing idle time at the critical work center;
- hold jobs not going to the critical work center in the order pool longer, reducing the level of finished goods inventory.

1.7 ORGANIZATION OF DISSERTATION

The research questions are explored in detail in the following chapters.

Chapter 2 reviewed and discussed prior relevant research from the ORR literature and DRC literature. A taxonomy of prior research is presented.

Chapter 3 describes the assumptions underlying the structure of the shop and the operation of the control system. This chapter includes a discussion of the research methodology used, the processing of the simulation code, the validation and verification of the simulation model and the performance measurements.

Chapter 4 states and examines the a priori hypotheses and explains the experimental variables. The chapter then discusses the data collection techniques. The data from the pilot runs is explored and used to establish parameters for the settings used for the experimental variables.

Chapter 5 presents the experimental results, the analysis of the data from the main experiments, and the sensitivity analysis of the data.

Chapter 6 summarizes the research and develops conclusions and recommendations for future research.

1.8 CONCLUDING COMMENTS

The focus of operations management has changed so it now emphasizes a broader systems wide perspective. It is now recognized that activities that take place on the shop floor are influenced by actions taken elsewhere within the

manufacturing system at the planning or design stages. With this recognition comes the awareness that multiple opportunities exist to control the work flow and the nature of the work flow. Two of these control mechanisms explored here were the ORR system and the planning system. Operations managers now recognize the value of using information from the shop floor activities to coordinate work flows.

Three SFC activities (ORR, "when" labor assignment and job dispatching) are explored within the context of the DRC shop. This study examines how these three areas can be used to improve operations and to enhance system performance. These areas are examined to make both researchers and managers aware of their potential combined impact on shop performance. Dispatching is not an isolated activity. By recognizing the presence of these other SFC decisions, the scheduling research can become consistent with the realities faced by managers of multiple resource constrained systems. The remainder of this dissertation builds on the issues raised in this first chapter.

CHAPTER 2

2.1 INTRODUCTION TO LITERATURE REVIEW

This research studies how information has been used to control the flow of work in two separate bodies of research: the dual resource constrained research and the order review/release research. To understand this study, it is necessary to know what work has been done previously in these two areas. This review will illustrate how prior research has approached the problem of controlling the flow of work and suggest extensions of the past research. Finally, this review will provide a rationale for this study by positioning it in the existing body of knowledge.

All the acronyms used in this literature review and throughout this study are defined at the front of this study in the "Key to Abbreviations".

This chapter is divided into three major sections. They are:

1. a review and taxonomy of the ORR literature,
2. a review and taxonomy of the DRC literature,
3. a summary and integration of the ORR and DRC literature.

2.2 ORDER REVIEW/RELEASE

The terms input/output (I/O) control, input control and output control are used by some authors instead of the term order review/release. Some examples of how these terms are used in the literature are given below.

- Wight (1970) and Belt (1976) used the term I/O control to refer to techniques which ensure that input to the shop is less than or equal to output from the shop.
- Kingsman, Tatsiopoulos and Hendry (1989) used the term input control to mean the process of leveling the work load whether the leveling was performed by the planning system or by the ORR rule.
- Kingsman et al. (1989) defined the term output control as the process of adjusting the work center capacity to match the load in the shop.

These authors do not provide clear definitions of how the terms they use differ from or correspond to other terms such as order review/release. These terms as used in this research are defined below.

2.2.1 DEFINITION OF TERMS

ORR was defined by Melnyk et al. (1985) as consisting of four major activities. These activities are: 1) order documentation, 2) material checking, 3) capacity evaluation and 4) load leveling. Each is discussed separately below:

1. Order documentation obtains and organizes the information the shop floor needs to process the order.
2. Material checking ensures that raw materials are available before the order is released.

other

the pl

is don

backw

synony

concer

shop.

as mac

overlap

capacit

T

tively re

This defi

3. Capacity evaluation estimates whether sufficient capacity is available to process the order.
4. Load leveling controls the release of orders to the shop floor. Load leveling does not involve the planning system. It controls the flow of work after the work leaves the planning system. Load leveling is the major focus of ORR research in the literature.

The differences and similarities between the definitions of ORR and the other terms listed above is illustrated in Figure 2-1. Load leveling performed by the planning system is referred to in this research as "SMOOTHING", whether it is done by changing the order acceptance practices or moving work forwards or backwards by changing the due dates.

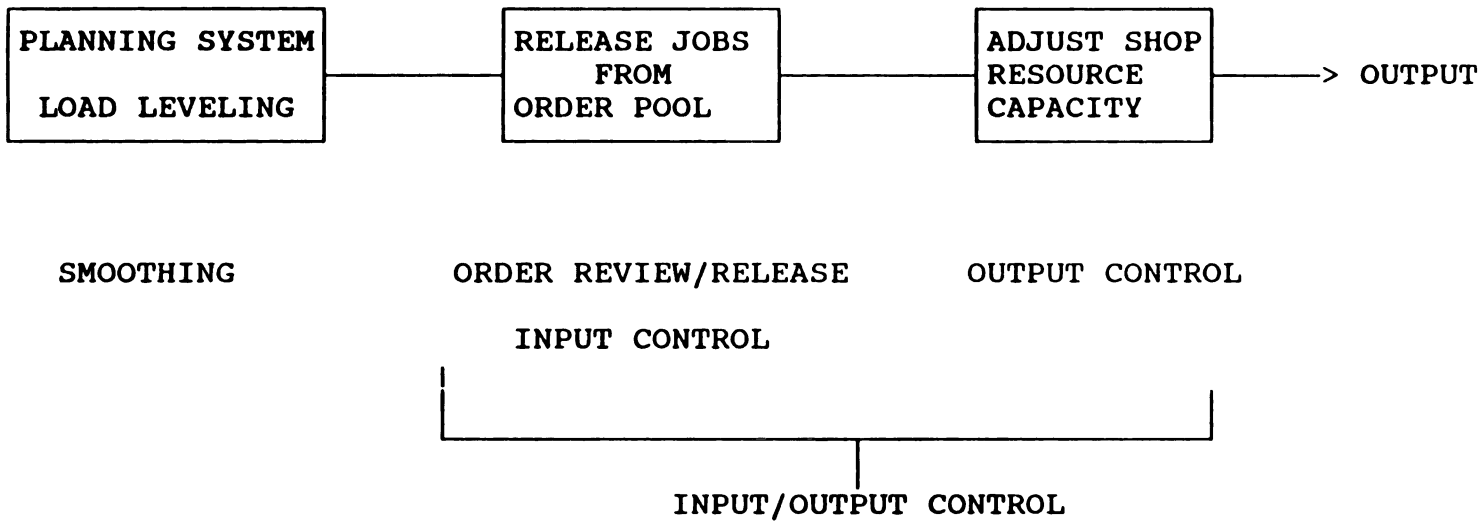
Two of the activities of ORR (capacity evaluation and load leveling) are synonymous with the term "Input Control". Both ORR and Input Control are concerned with selecting jobs from the order pool and releasing them to the shop.

The term "Output Control" refers to the adjustment of shop resources such as machines or labor by working overtime or undertime. These activities do not overlap with ORR because ORR evaluates available capacity, it does not adjust capacity.

The term "Input/Output Control" refers to techniques which both selectively release orders from the order pool to the floor and do adjust capacity. This definition incorporates ORR as a special case as illustrated in Figure 2-1.

FIGURE 2-1
DEFINITIONS OF TERMS

FIGURE 2-1
DEFINITIONS OF TERMS



This

the d

of vie

Ragat

B

is time

that eit

The literature concerned with each of these areas is reviewed separately. This review begins with the ORR literature.

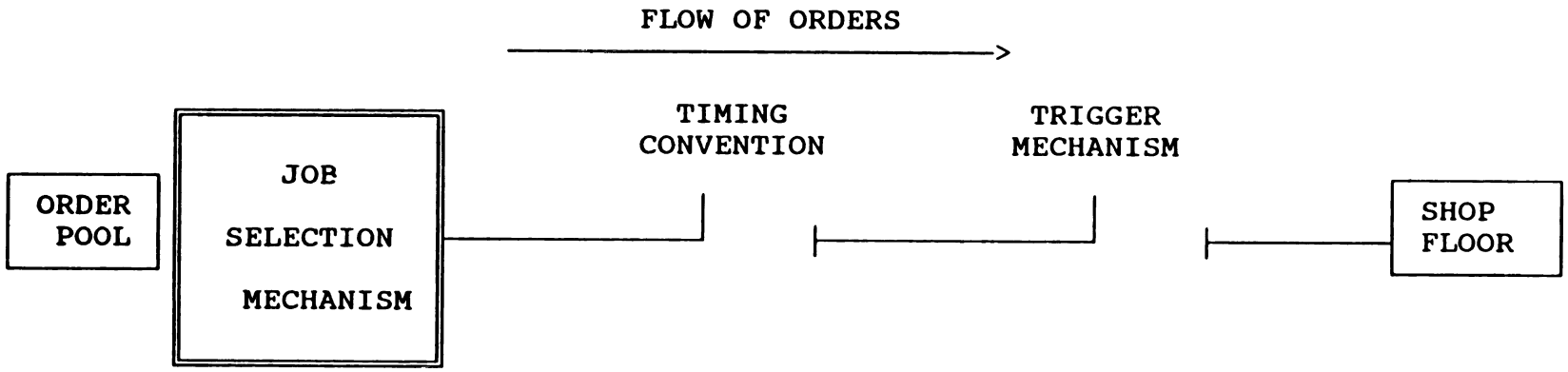
Additional terminology is needed to explain the load leveling activity of the different ORR rules or input control methods used in the literature. One way of viewing the literature is to use the framework developed by Melnyk and Ragatz (1989). They identified three distinct mechanisms in each ORR rule.

- the timing convention which determines whether releases from the pool to the shop floor are made continuously or periodically. It is shown in Figure 2-2 as an open switch which must be closed for orders to leave the order pool.
- the trigger mechanism which checks to make sure that a specific set of conditions is met before releasing work to the shop. It is illustrated in Figure 2-2 as a switch which must be closed for orders to be released.
- the job selection mechanism which selects the job from the order pool to release to the shop when the timing and the trigger level allow a release.

Both the timing convention and the trigger mechanism must agree that it is time to release a job before a job is selected from the order pool. This means that either mechanism can prevent the release of a job.

FIGURE 2-2
ORDER REVIEW/RELEASE FUNCTION

FIGURE 2-2
ORDER REVIEW/RELEASE FUNCTION



secti

sectio

2.2.2 STRUCTURE OF ORR LITERATURE REVIEW

This review of the ORR literature is divided into three sections. The sections are:

1. a review of the formal experiments in ORR;
2. a review of the applications and implementations of ORR systems which have been reported; and,
3. a review of the descriptive research, field studies and the research about input/output control.

The complete taxonomy is illustrated in Figure 2-4 at the end of this section. All the literature is reviewed using the structure shown below:

- What information sources are used by the ORR rule to level the load in the shop.
- What information sources are used in the job dispatching rule to select orders from the queue for processing.
- If the research about ORR is placed in a DRC shop, what information sources are used in the labor assignment rules.
- Does the planning system used in the research level the work load before it is sent to the order pool.

2.2.3 EXPERIMENTAL RESEARCH

There are four different types of experimental research in the literature. They are:

- single machine research;
- multiple machine research with optimization;
- multiple machine research using heuristics;
- labor and machine constrained research.

Each of these types is discussed separately.

2.2.3.1 SINGLE MACHINE RESEARCH

The experimental factors used in the single machine experiments are summarized in Table 2-1. Baker (1984) used simulation to examine the single machine system and Kanet (1988) used queuing analysis. The purpose of these two studies was to initiate a systematic examination of the consequences of input control.

Baker (1984) used a continuous (C) timing convention. The trigger mechanism was the mean shop load (MSL). MSL was defined as the hours of processing time remaining for all jobs in the queues or jobs currently being processed on the machine. It operated independently (IND) of the job selection mechanism and used SHOP information. MSL allowed the shop to be loaded with work from the order pool up to a predetermined limit. The job selection mechanism used to select jobs from the pool to place into the shop was First Come First Served (FCFS) which used only SHOP information. This ORR rule

CHA

ORF

J

DIS
S

SH
SH
PLA
JOB
DU
RO
SHOP
EA
PERF

RESE
STAT

TABLE 2-1
SUMMARY OF ORR SINGLE MACHINE RESEARCH

	SINGLE MACHINE	
CHARACTERISTICS	BAKER 1984	KANET 1988
ORR RULE		
TIMING CONVENTION	C	C
TRIGGER MECHANISM	IND.	IND.
POOL		
SHOP	MSL	MSL.
SHOP, POOL		
SHOP, POOL, LABOR		
JOB SELECTION MECH.	IMM	IMM
POOL	FCFS	FCFS
SHOP		
SHOP, POOL		
SHOP, POOL, LABOR		
DISPATCH RULE		
SHOP	FCFS, SOT CR, MODD	FCFS
SHOP, POOL		
SHOP, POOL, LABOR		
PLANNING SYSTEM	POISSON	POISSON
JOB CHARACTERISTICS		
DUE DATE	TWK, SLK	
ROUTING		
SHOP CHARACTERISTICS		
EARLY COMPLETION	Y	Y
PERFORMANCE CRITERIA	T	F
RESEARCH METHODOLOGY	SIMUL.	ANALYTIC.
STATISTICAL ANALYSIS	-	

was tested with a variety of job dispatching rules and compared to the immediate release (IMM) of jobs to the shop. Baker (1984) concluded that the SFC system using the modified operation due date (MODD)[1] job dispatching rule and immediately releasing jobs to the floor (IMM) performed better than the SFC system using ORR rules.

Kanet (1988) used the same timing convention, trigger mechanism and job selection mechanism as Baker (1984), but used only the FCFS job dispatching rule. Kanet (1988) analyzed the shop as an M/M/1 queuing model. He used the results of the queuing analysis and the results of prior simulation experiments to develop an explanation for the behavior of input control in job shops. He concluded that in a M/M/1 shop neither the system flow time nor the number of jobs in the system are changed. What changes is where the jobs wait. With input control, jobs wait in the order pool instead of waiting on the shop floor. Kanet reasoned that if the shop had multiple machines, the best trigger mechanism would release work to the gateway work center only when the queue of the gateway work center was empty. By doing this, the average flow time of jobs in the system would remain the same while the jobs would wait in the order pool and not on the shop floor. He hypothesized that by placing load limits on downstream work centers, idle time might be introduced into the schedule and

---

[1]. MODD = $\min\{\max(TNOW + OPT_j; ODD_j)\}$ where TNOW is the time now, OPT is the operation processing time of operation j and ODD is the operation due date of operations j.

cause deterioration of the system performance. This would result in an increase in mean tardiness.

2.2.3.2 MULTIPLE MACHINE RESEARCH WITH OPTIMIZING ALGORITHMS

The experimental factors used in these experiments are summarized in Table 2-2. There are three general approaches in the literature to optimize the selection of jobs in the order pool. They are:

- Formulate the problem as a mixed integer program (MIP). This approach was taken by Irastorza and Deane (1974, 1976), and Onur and Fabrycky (1987).
- Develop a schedule using FFL and then find improvements in the schedule. This approach was taken by Bobrowski (1989) and Nicholson and Pullen (1971).
- Using a modified linear decision rule (LDR) to prioritize jobs in the order pool. These jobs are loaded up to the MSL for the shop. This approach was used by O'Grady and Azoza (1987).

Each of these approaches is discussed in turn.

2.2.3.2.1 Mixed Integer Programming

Irastorza and Deane (1974, 1976) developed a mixed integer program (MIP) to select jobs from the order pool. The MIP used POOL and SHOP information to balance the workload in the shop between work centers while ensuring that jobs met their due dates. The timing convention was periodic.

TABLE 2-2
SUMMARY OF ORR EXPERIMENTAL LITERATURE

TABLE 2-2
SUMMARY OF ORR EXPERIMENTAL LITERATURE

	MULTIPLE MACHINES WITH OPTIMIZATION					
CHARACTERISTICS	BOBROWSKI 1989	IRASTORZA & DEANE 1974	IRASTORZA & DEANE 1976	O'GRADY & AZOZA 1987	ONUR &FABRYCKY 1987	NICHOLSON & PULLEN 1971
ORR RULE						
TIMING CONVENTION	P	P	P	P	P	P
TRIGGER MECHANISM	IND. & RD	IN MIP	IN MIP	IND	IN MIP	
POOL						
SHOP	MSL	AWL/M	AWL/M	MSL	MSL	
SHOP, POOL						
JOB SELECTION MECH.		IMM	IMM			
POOL	FCFS					
SHOP						
SHOP, POOL	MIL, FFL+ SEARCH	MIP	MIP	LDR: NES, WLS	FFL, MIP	FFL + SEARCH
SHOP, POOL, LABOR						
DISPATCH RULE						
SHOP	CR	SOT, SRO	SOT, SRO	FCFS,SOT	FCFS	FCFS
SHOP, POOL						
SHOP, POOL, LABOR						
PLANNING SYSTEM	POISSON	POISSON	POISSON	NORMAL	POISSON	-
JOB CHARACTERISTICS						
DUE DATE		SLK	SLK	TWK	TWK	-
ROUTING	ALTERN.					
SHOP CHARACTERISTICS						
EARLY COMPLETION	Y	Y	Y	N	Y	Y
PERFORMANCE CRITERIA	COST	WIP, T MWB,SWB, V(L)	WIP, T MQB,V(L)	2*ORDER COST	SDD	COST
RESEARCH METHODOLOGY	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SEARCH
STATISTICAL ANALYSIS	COMPAR.	N	N	N	N	N

The trigger mechanism was based on the work load present at each work center (AWL/M). The trigger mechanism was included in the objective function of the MIP. The job dispatching rules were slack for remaining operations (SRO) and shortest operation processing time (SOT) both of which used only SHOP information. They found that shops without ORR performed better than shops with ORR mechanisms.

Onur and Fabrycky (1987) developed an optimizing algorithm to do I/O control in a job shop. They simulated the performance of their system against a system which did only input control. The algorithm was formulated as a mixed integer program (MIP) which considered the workload level in the shop, the WIP inventory level and the due date status of the jobs. The simulation was run first to obtain estimates of waiting time for given loads to use as input into the MIP. However to avoid computational problems, only a small number of possible waiting times could be estimated using the simulation. They compared their MIP to a finite loading heuristic which was not fully described. The shop load was used as the trigger mechanism for the finite loading algorithm. They concluded that when the shop was heavily loaded there was significant improvement in overall performance when their MIP was used for I/O control as opposed to using finite loading for input control.

2.2.3.2.2 Finite Forward Loading With Search Algorithm

Bobrowski (1989) developed an ORR rule which used a Forward Finite Loading (FFL) job selection mechanism for a flow dominant machine constrained

shop with alternate routes. The timing convention was continuous, and the trigger mechanism was included in the job selection mechanism. FFL worked by first using the critical ratio (CR) rule to organize the order pool. Then jobs were loaded onto the machines sequentially in the first time period the machine had available capacity. If the final operation was scheduled to be completed on or after the job's due date the job was released. A loading exchange heuristic then examined the schedule to identify improvements. This FFL job selection mechanism used both POOL and SHOP information. It was tested against the modified infinite loading (MIL) ORR rule which will be explained later and against the MSL ORR rule. Under most conditions the FFL rule with the exchange heuristic improved performance of the shop.

Nicholson and Pullen (1971) proposed a Direct Search (DIR) algorithm for loading the shop which held jobs off the floor. The algorithm worked by first loading all the jobs into the shop. Using FFL, the delay costs of this schedule were calculated. The algorithm then checked to see if the delay cost could be reduced by removing one of the jobs from the shop and putting it into the order pool. This was repeated until no improvement was found. A small example was worked out. The algorithm did not use a trigger mechanism. Once on the floor jobs were controlled with the FCFS job dispatching rule. At different points in the routing, the job could choose alternative routes which were selected by sending the job to the machine with the smallest load.

2.2.3.2.3 Modified Linear Decision Rule

O'Grady and Azoza (1987) developed an algorithm to load jobs into the shop they called 'Net Excess Stock' (NES). The algorithm developed a priority list for loading jobs into a shop by considering the WIP and FGS inventories in a make-to-stock firm. They developed a second algorithm as an extension to NES called work load smoothing (WLS). WLS loaded the work and balanced the work loads between machines to avoid bottlenecks. The trigger mechanism for each algorithm was the shop load. They tested the algorithms using a simulation of a shop with 15 different products where the demand for each product was drawn from a normal distribution with a mean of 100, and a standard deviation between 50 and 100. Any negative values for demand were converted to zero. Costs were calculated using idle time, WIP, FGS and back order costs. They concluded that inclusion of WLS in the algorithm improved performance.

2.2.3.3 MULTIPLE MACHINE RESEARCH USING HEURISTICS

Several researchers examined the effect of using ORR rules on shop performance. Their goal was to test different heuristics for loading the shop against each other and against the immediate release of all jobs in the pool. Machines were the only constraints in these experiments. The experimental factors are summarized in Table 2-3.

Bertrand (1983) investigated the use of a simple ORR rule which used only SHOP information. This ORR rule used a continuous timing convention, and the trigger mechanism was the mean shop load (MSL) measured in hours of

processing time remaining for all jobs in the queue or in service on the machines. FCFS was used as the job selection mechanism to select a job from the pool. He found that the ORR function was not an important factor in reducing the standard deviation of lateness compared to the due date rule and the job dispatching rule.

Ragatz and Mabert (1988) examined the performance of four different ORR rules with four job dispatching rules under three different conditions of due date tightness in a machine constrained job shop. The timing convention was always periodic (P) but they used two different trigger mechanisms. When the MNJ job selection mechanism was used, the trigger mechanism was the number of jobs on the shop floor (NUM). For the other job selection mechanisms, the trigger mechanism was included as part of the release date calculation in the job selection mechanism. A major goal of their study was to examine how information used in the job selection mechanism influenced the performance of the job selection mechanism. All four ORR rules were compared to each other and to the immediate release of jobs (IMM). The dominant job selection mechanism was MIL which used both POOL and SHOP information to calculate job release dates by subtracting a multiple of the job's processing time and the current shop congestion from the job's due date. Under some conditions backward finite loading (BFL) performed as well as MIL, but never better.

BFL worked by developing a time phased load for each work center using information from the POOL and the SHOP. Backward infinite loading (BIL) used

TABLE 2-3
SUMMARY OF ORR EXPERIMENT WITH MULTIPLE MACHINES

	MACHINE CONSTRAINED		
CHARACTERISTIC	BERTRAND	MELNYK & RAGATZ	RAGATZ & MABERT
	1983	1989	1988
ORR RULE			
TIMING CONVENTION	C	C	P
TRIGGER MECHANISM	IND.	IND.	IND. & RD
POOL			
SHOP	MSL	MSL, QWL	NUM
SHOP, POOL			
SHOP, POOL, LABOR			
JOB SELECTION MECH.	IMM	IMM	IMM
POOL	FCFS		BIL
SHOP			MNJ
SHOP, POOL		AGGWINQ, WCEDD	MIL, BFL
DISPATCH RULE			
SHOP	EDD,CEDD	FCFS, SOT EDD,SRO	FCFS, SOT EDD, CR
SHOP, POOL			
SHOP, POOL, LABOR			
PLANNING SYSTEM	POISSON	POISSON, CONSTANT	POISSON
DUE DATE RULE	SLK	TWK	TWK
JOB ROUTINGS	RANDOM	RANDOM	RANDOM
EARLY COMPLETION OK	Y	Y	N
PERFORMANCE CRITERIA	M(SDD)	M(T),P(T) M(J),M(Q) M(S) M(SDD(Q))	COST
RESEARCH METHODOLOGY	SIMUL.	SIMUL.	SIMUL.
STATISTICAL ANALYSIS	ANOVA	ANOVA	ANOVA

only information from the POOL to calculate the job release dates. Minimum number of jobs (MNJ) selected a job from the pool to release to the shop using the current dispatch rule whenever the number of jobs in the shop dropped below a minimum level. MNJ used both SHOP and POOL information. Both BIL and MNJ were dominated by MIL. They found that shops which used ORR always performed as well as or better than shops which released jobs immediately (IMM).

Melnyk and Ragatz (1989) tested two job selection mechanisms in a machine constrained system. Both mechanisms used a continuous timing convention and incorporated the trigger mechanism into the job selection mechanism. The first mechanism (AGGWINQ) used the aggregate shop load limit as the trigger mechanism, and FCFS to select the job in the pool whose initial work center had the fewest jobs in the queue. The second mechanism (WCEDD) used the level of work in individual queues (QWL) as the trigger mechanism. A job was released to the work center whose queue contained less than the desired QWL. Both of these mechanisms used SHOP and POOL information. They found that using ORR mechanisms introduced tradeoffs into the system. They traded off poor delivery performance for reduced WIP inventory.

Melnyk and Ragatz (1989) explained their results by developing a model of ORR in a multiple machine shop. This model supported the explanation proposed by Kanet (1988) based on his results for a single machine shop. First the use of ORR for input control did not change the average time in the system

for jobs, but shifted much of the queue time from the shop floor to the order pool. Second the use of ORR provided some advantages by increasing the system flexibility and reducing congestion on the shop floor, but it traded off reduced WIP inventory for decreased delivery performance. Melnyk and Ragatz (1989) proposed four possible explanations for the degradation of delivery performance. Their first explanation was that the job selection mechanism was not "powerful" enough, i.e. it did not adequately process the information available to it. This agreed with Kanet's (1988) explanation that limiting loads to downstream work centers introduced idle time into the schedule. Their second explanation was that the trigger mechanisms were not appropriate for the job selection mechanisms. Third, they proposed that the use of the total work content (TWK) due date method to set job due dates ignored available POOL and SHOP information and could work counter to the ORR rule. And fourth, the ORR rule and the planning system were not adequately integrated with each other. To explore this last explanation, they conducted a second set of experiments to examine the integration of planning and ORR. They did this by keeping the number of jobs released each period by the planning system to the shop constant. This second set of experiments was not statistically analyzed, but the smoothing by the planning system did seem to improve the overall performance. However, the relative performance of the ORR rules compared to systems which released jobs immediately was not improved. Their explanation for this was that ORR should be operated as part of a closed loop system where information was fed back from

the shop to planning. This would adjust capacity (Output Control) and smooth the flow of orders from planning to the shop as well as using ORR to level the load. This is illustrated in Figure 2-3 where orders flow in one direction and information flows in the other direction. This integrated system in Figure 2-3 used not only I/O control but planning system also SMOOTHED the load.

2.2.3.4 ORR RESEARCH IN DUAL RESOURCE CONSTRAINED SHOPS

There have been only two studies of ORR in DRC shops. The experimental factors used in these studies are shown in Table 2-4. This table also summarizes the labor assignment procedures used in the experiment.

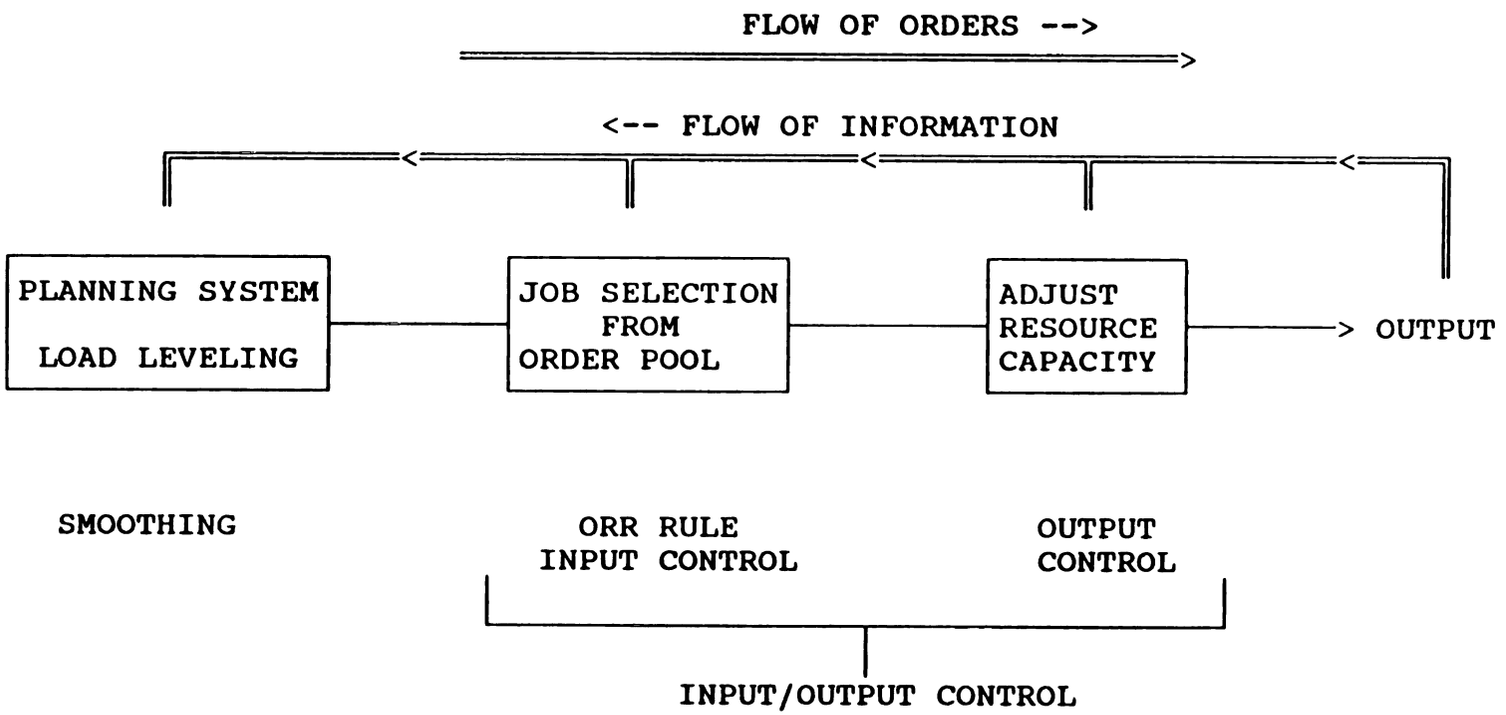
Bobrowski and Park (1989) investigated the ORR function in a DRC shop where each operator was 100% efficient on every machine while Park and Bobrowski (1990) investigated the ORR function in a DRC shop where the efficiency of the operator on each machine varied. In both studies a periodic timing convention was used and both studies examined the performance of two job selection mechanisms: modified infinite loading (MIL)[1], and forward finite loading (FFL). The trigger mechanism for these two rules was included in the job selection mechanism as part of the release date (RD) calculation. The Bobrowski and Park (1989) study also examined the maximum shop load (MSL) trigger mechanism worked independently (IND) of the critical ratio (CR) job selection mechanism. There was no labor transfer delay in either study, but

---

[1]. The authors referred to their rule as backward infinite loading, but because of its great similarity to the MIL rule used by Ragatz and Mabert (1988) it is referred to here as MIL.

FIGURE 2-3
CLOSED PLANNING & SFC SYSTEM

FIGURE 2-3
CLOSED PLANNING & SFC SYSTEM



CH

OR

DI

PI
"W

"W

DU
JO
EA
PE
RE
ST

TABLE 2-4
SUMMARY OF ORR EXPERIMENTS IN DRC SHOPS

CHARACTERISTIC	PARK BOBROWSKI 1990	BOBROWSKI & PARK 1989
ORR RULE		
TIMING CONVENTION	P	P
TRIGGER MECHANISM	RD	IND. & RD
POOL		
SHOP		MSL
SHOP, POOL		
JOB SELECTION MECH.		IMM
POOL		CR
SHOP, POOL	MIL, FFL	MIL, FFL
SHOP, POOL, LABOR		
DISPATCH RULE		
SHOP	CR	CR, MODD
PLANNING SYSTEM	POISSON	POISSON
"WHERE" LABOR ASSIGN.		
SHOP		
SHOP, POOL		
SHOP, LABOR	LONQ	LONQ
SHOP, POOL, LABOR		
"WHEN" LABOR ASSIGN.		
SHOP		
SHOP, POOL		
SHOP, LABOR	QUE, JOB	QUE, JOB
SHOP, POOL, LABOR		
DUE DATE RULE	TWK	TWK
JOB ROUTINGS	RANDOM	RANDOM
EARLY COMPLETION OK	N	N
PERFORMANCE CRITERIA	COST	COST
RESEARCH METHODOLOGY	SIMUL.	SIMUL.
STATISTICAL ANALYSIS	ANOVA	ANOVA

TABLE 2-5
SUMMARY OF ORR APPLICATIONS LITERATURE

CHARACTERISTIC	BECHTE 1982	BECHTE 1988a	BECHTE 1988b	SHIMOYASHIRO ISODA & AWANE 1984
ORR RULE				
TIMING CONVENTION	P	P	P	P
TRIGGER MECHANISM	IND.	IND.	IND.	
POOL				
SHOP	WCL	WCL	WCL	QWL
SHOP, POOL				
SHOP,POOL,LABOR				
JOB SELECTION MECH				
POOL				
SHOP				
SHOP, POOL	LC	LC	LC	VECTSIM
SHOP,POOL,LABOR				
DISPATCH RULE				
SHOP	FCFS	FCFS, EDD	FCFS, EDD	SOT, FCFS
SHOP, POOL				
SHOP,POOL,LABOR				
PLANNING SYSTEM	ACTUAL	ACTUAL	ACTUAL	ACTUAL
JOB CHARACTERISTICS				
DUE DATE	ACTUAL	ACTUAL	ACTUAL	ACTUAL
SHOP CHARACTERISTIC				
FLOW	ACTUAL	ACTUAL	ACTUAL	ACTUAL
EARLY COMPLETION				
BATCHING				
PERFORMANCE CRITERIA	WIP	WIP, LT	WIP,LT	M(F),TOTAL(L)
RESEARCH METHODOLOGY	SIMUL	IMPLEM.	IMPLEM.	SIMULATION
STATISTICAL ANALYSIS	-	-	-	-

there was a labor transfer cost. These studies were the first to combine the ORR literature and the DRC literature. However, the studies used only SHOP and LABOR information for the labor assignment rules. These studies did not use the LABOR information for the ORR decisions. Bobrowski and Park (1989) tested two job dispatching rules, CR and MODD. They found that the MODD dispatching rule performed better than the CR rule using a total cost performance measure. They did not use additional information in the job dispatching rules.

Park and Bobrowski (1990) investigated two "when" labor assignment rules called JOB and QUE. The JOB rule allows an operator to be eligible for transfer whenever a job is completed. The QUE rule allows an operator to be eligible for transfer when all the work in the queue is completed. Bobrowski and Park (1989) used only the JOB labor assignment rule. The "where" labor assignment rule used in both studies stipulated that eligible workers be transferred to the work center with the longest queue length measured in hours of work (LONQ). Both studies found that the JOB rule performed better than the QUE rule; that the cross training of workers to improve their efficiency on different machines was a significant factor in shop performance; and that MIL was the dominant job selection mechanism.

2.2.3 APPLIED RESEARCH

The studies by Bechte (1982, 1988a, 1988b) shown in Table 2-5 used the same ORR rule. Its job selection mechanism used SHOP and POOL information

to estimate the future load at each work center. The rule attempted to anticipate which work center could become a bottleneck and balance the load in the shop to prevent the bottleneck. The future load on a work center was estimated by calculating the net present value of future work at each work center. The net present value was calculated by multiplying the work to be processed at each work center by a discount factor. The discount factor was a predetermined discount rate raised to the (I - 1) power, where I was the number of operations preceding the operation being discounted. This technique was called Load Conversion (LC). The ORR rule used a periodic timing convention (P) and a trigger mechanism which was independent of the job selection mechanism and used the amount of work in the individual work center to determine when to trigger a release. In tests in actual factories, he found that lead times, lead time variance and inventory were reduced without impacting capacity utilization.

Shimoyashiro, Isoda and Awane (1984) tested a job selection mechanism using data from an actual machine constrained job shop. The mechanism used POOL and SHOP information to balance the load between work centers by estimating the load at each work center in future time periods. They used a periodic timing convention and included the trigger mechanism in the job selection mechanism. They reported that ORR improved shop performance compared to the system the shop had used previously.

2.2.4 DESCRIPTIVE RESEARCH ABOUT I/O CONTROL

This portion of the review is subdivided into input control, input/output control and output control, as shown in Table 2-6. Some of this research did not include any part of the I/O control logic as an experimental variable. However, all the research occurred in a shop which used I/O control. Those studies which did not use I/O as an experimental variable are included here because they help to demonstrate the many different situations in which I/O control is used.

2.2.4.1 INPUT CONTROL

This section reviews three different types of research about input control. They are:

- Research with input control in actual shops, Browne and Davies (1984);
- Descriptions about how input control is actually used or should be used. This includes Rowe (1960), Melnyk et al. (1985), Melnyk and Carter (1987), and Sandman and Hayes (1982).
- Research in shops using an ORR rule as part of the control system but not as an experimental variable (Bobrowski and Mabert, 1988).

Each of these is discussed in turn.

2.2.4.1.1 Input Control In Practice

Browne and Davies (1984) developed a simulation for an actual job shop whose orders required assembly. The shop already used ORR for loading and

TABLE 2-6
SUMMARY: DESCRIPTIVE I/O CONTROL RESEARCH

INPUT CONTROL	INPUT/OUTPUT CONTROL	OUTPUT CONTROL
BOBROWSKI, MABERT (1988)	BELT (1976)	ACKERMAN (1963)
BROWNE, DAVIS (1984)	FRY, SMITH (1987)	HOLLOWAY, NELSON (1973)
MELNYK, CARTER, DILTS, LYTH (1985)	GRAVES (1986)	KARNI (1982)
MELNYK, CARTER (1987)	GUE (1975)	
MELNYK, RAGATZ (1988)	HARTY (1973)	
ROWE (1960)	HEILWEIL (1989)	
SANDMAN, HAYES (1982)	KINGSMAN, TATSIOPOULOS, HENDRY (1989)	
WIGHT (1970)	PLOSSL, WIGHT (1973)	
	TATSIOPOULOS (1988)	
	TATSIOPOULOS, KINGSMAN (1983)	

their goal was to find a better ORR rule. The ORR rules which were tested used SHOP and POOL information to calculate the release date of the job. The only variable in the simulation was a flow allowance subtracted from the due date to determine when the job was to be released.

2.2.4.1.2 Descriptions of Input Control

Rowe (1960) developed release rules to delay the release of jobs in the shop. He referred to these rules as flow allowance rules. The flow allowance of a job determined when it would be sent to the shop. The flow allowance was based on the amount of work in process on the job's route. They found that the ORR rule did not change the total waiting time of all the jobs, but it did change which jobs waited the longest. Similar results were later found by Kanet (1988) and Melnyk and Ragatz (1989).

Melnyk, Carter, Dilts and Lyth (1985) and Melnyk and Carter (1987) surveyed job shops and the job shop literature. They developed the model illustrated in Figure 1-2, which shows ORR as the first step in SFC. This model was described earlier and is not described here again.

Sandman and Hayes (1982, pp.65-77) described a simple method of input control which operated by counting the number of orders on the floor. They did not test the method which is very similar to the methods tested by Baker (1984), Kanet(1988), Bertrand (1983) and Ragatz (1985).

2.2.4.1.3 Input Control In The Shop Floor Control System

Bobrowski and Mabert (1988) examined the impact of using different alternate routing strategies in a job shop. The job shop control system included the use of the finite forward loading (FFL) ORR rule. They did not form any conclusions about the effect of using an ORR rule.

2.2.4.2 INPUT/OUTPUT CONTROL

As defined earlier, I/O control meant that both the work entering the shop and the capacity in the shop could be adjusted. This discussion of I/O control is organized in three sections. They are:

- reviews of literature which are primarily explanations of how I/O control works.
- reviews of literature which explain how to implement I/O control methods in the shop.
- reviews of mathematical formulations of I/O control.

2.2.4.2.1 Explanations of I/O control

Plossl and Wight (1973) and Wight (1970) described how I/O control worked. Advantages which he saw for the use of I/O control were that it allowed the expediting of orders, and it controlled manufacturing lead time. He recommended that the gateway work center never have a large queue. He then described the typical implementation problems of I/O control.

Harty (1973) described the seven principles for controlling capacity, which included input control, output control and the limiting of the work input to both the gateway and downstream work centers.

Belt (1976) described and developed methods to perform capacity requirements planning as described by Wight (1970). This method involves the use of tables to limit input and increase output by adjusting capacity as necessary to ensure that the lead times of products do not change.

2.2.4.2.2 Implementations of I/O Control

Gue (1975) described how input/output control worked in a foundry which used a MRP system. They separated the concept of I/O control from load balancing. Load balancing was described as the need to shift labor between work stations. I/O control was defined as planning for a given output, staffing for that output and limiting input to the corresponding amount.

Fry and Smith (1987) described the implementation of I/O control procedures in a job shop. The six step procedure that they used required a shift in performance measures, identification of bottlenecks, establishing of maximum inventory levels at work stations, the reduction of production lot sizes, change of the dispatch rules and the limiting of input to equal output.

Kingsman, Tatsiopoulos and Hendry (1989) described in detail how to implement input/output control. They described how to use I/O control to implement an "order accepting" procedure which ensures that no orders are accepted or due dates committed which cannot be met. This was referred to as

SMOOTHING in Figure 2-1. In addition their method allowed for management interaction to change priorities of jobs as needed.

Tatsiopoulos (1988) discussed issues involved in using the standard I/O control software contained in some MRP software packages. Important influences on I/O that they described were the similarity of the routing of the jobs, the variance in the operation processing times of the jobs, the mean and the variance of the number of operations in the jobs, the nervousness of the master production schedule and the frequency of internal disruptions caused by machine breakdowns and operator absence. The software packages allow easy control of work at the gateway centers, but it is difficult to control work at downstream work centers. He suggested that I/O control could be improved by scheduling bottlenecks to reduce the variance of operation processing times. They suggested that I/O control be extended to the MPS level to control the customer order acceptance rate and the timing of due dates. This would regulate the input rate of orders into the order pool (i.e. SMOOTHING in Figure 2-1).

Tatsiopoulos and Kingsman (1983) reviewed the issues of lead time management. They concluded that I/O control work had made progress and they now needed to address the issue of the input from the planning system into the pool (i.e. SMOOTHING in Figure 2-1). This would allow the firm to control the total manufacturing lead time from order commitment by marketing to delivery of the finished goods.

2.2.4.2.3 Mathematical Formulations of I/O Control

Graves (1986) used I/O control at the planning level. This model did not limit input. Instead the model predicted the needed capacity level given the input into the shop. The goal was to keep the lead time for the product constant. This was done using a linear control model. The purpose of the model was to explore different shop configurations to analyze the trade off between shorter flow times and the efficient allocation of shop resources.

2.2.4.3 OUTPUT CONTROL

As defined earlier, output control meant that capacity was adjusted to equal the load entering the shop. The load being placed on the shop was not changed.

Ackerman (1963) was one of the first writers to address the issue of output control in the job shop. He developed an algorithm called "Even-Flow". The algorithm worked by scheduling the job backward from the due date, assigning one operation to each time period. Then the load of each work center in each time period was checked and unbalanced loads (i.e. the capacity and load are unequal) were adjusted by using alternate routes or different labor assignments.

Holloway and Nelson (1973) developed a formulation to adjust capacity of the work centers given a set of jobs with fixed due dates. The formulation was solved using heuristics.

indivi

done

into t

func

ty o

trea

Mab

yet

plan

atte

sea

taxo

use

LAB

exam

smoo

Karni (1982) developed optimizing algorithms to adjust capacity at individual work centers in response to their load at each time period. This was done by planning delays for each job. The algorithms assumed that the input into the work center was given and that the due dates must be met.

2.2.3 SUMMARY OF ORR RESEARCH

There is a large and diverse body of research on the order review/release function of shop floor control. The diverse findings illustrate both the complexity of the topic and the divergent views of what constitutes ORR. Some authors treat ORR strictly as input control into the shop (Baker, 1984; Ragatz and Mabert, 1988), some treat it as part of I/O control in the shop (Belts, 1976), and yet other authors treat ORR as the final "smoothing" of a workload which the planning system has already smoothed (Kingsman et al., 1989). This review has attempted to show the boundaries of the problems investigated by these researchers and suggest how they can be unified.

The taxonomy used in this review is illustrated in Figure 2-4. The taxonomy demonstrates that there has been no investigation of ORR rules which use all three sources of information available in the shop: POOL, SHOP and LABOR information.

A research area in ORR with potential benefits for the shop, is the examination of the load leveling done at the order pool when the load is being smoothed by the planning system. Melnyk and Ragatz (1989) conducted

preliminary investigation of this topic. They suggested that this relationship between ORR and the planning system be investigated further.

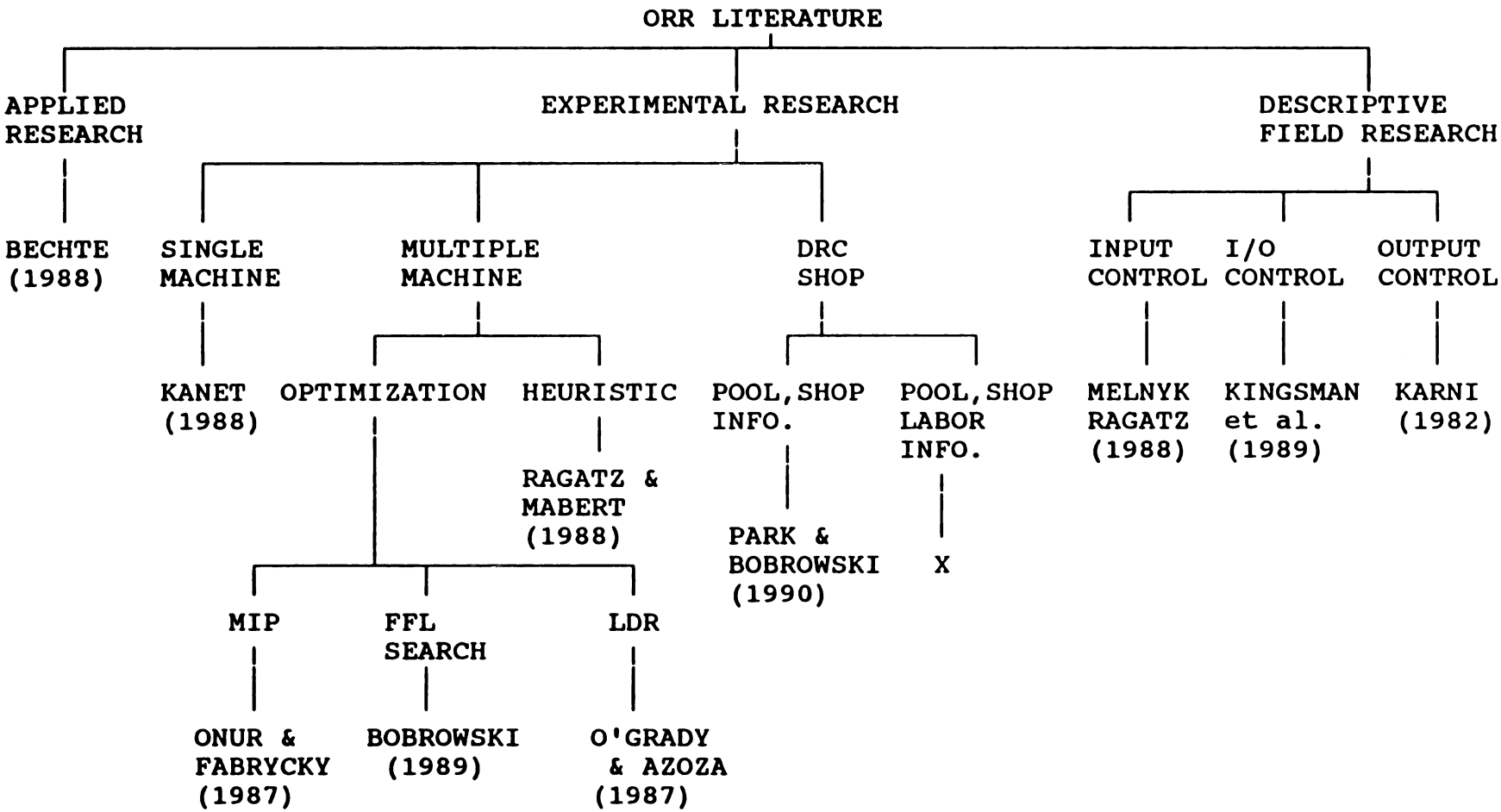
2.3 DUAL RESOURCES CONSTRAINED LITERATURE REVIEW

A dual resource constrained shop requires that two separate resources be available simultaneously to process a job. A DRC shop is illustrated in Figure 2-5. This illustration explicitly includes job dispatching and labor assignment in the model. Because there are multiple divisions in the shop, operators can be transferred both within and between divisions (Treleven, 1989).

Some of the research conducted on this type of shop illustrated in Figure 2-5 are: the impact of interdivisional transfers of labor; the impact of intra-divisional labor transfers; the effect of server transfer delays, or information access delays; the effect of cross-training of workers in the shop; and the effect of job dispatching rules in DRC shop.

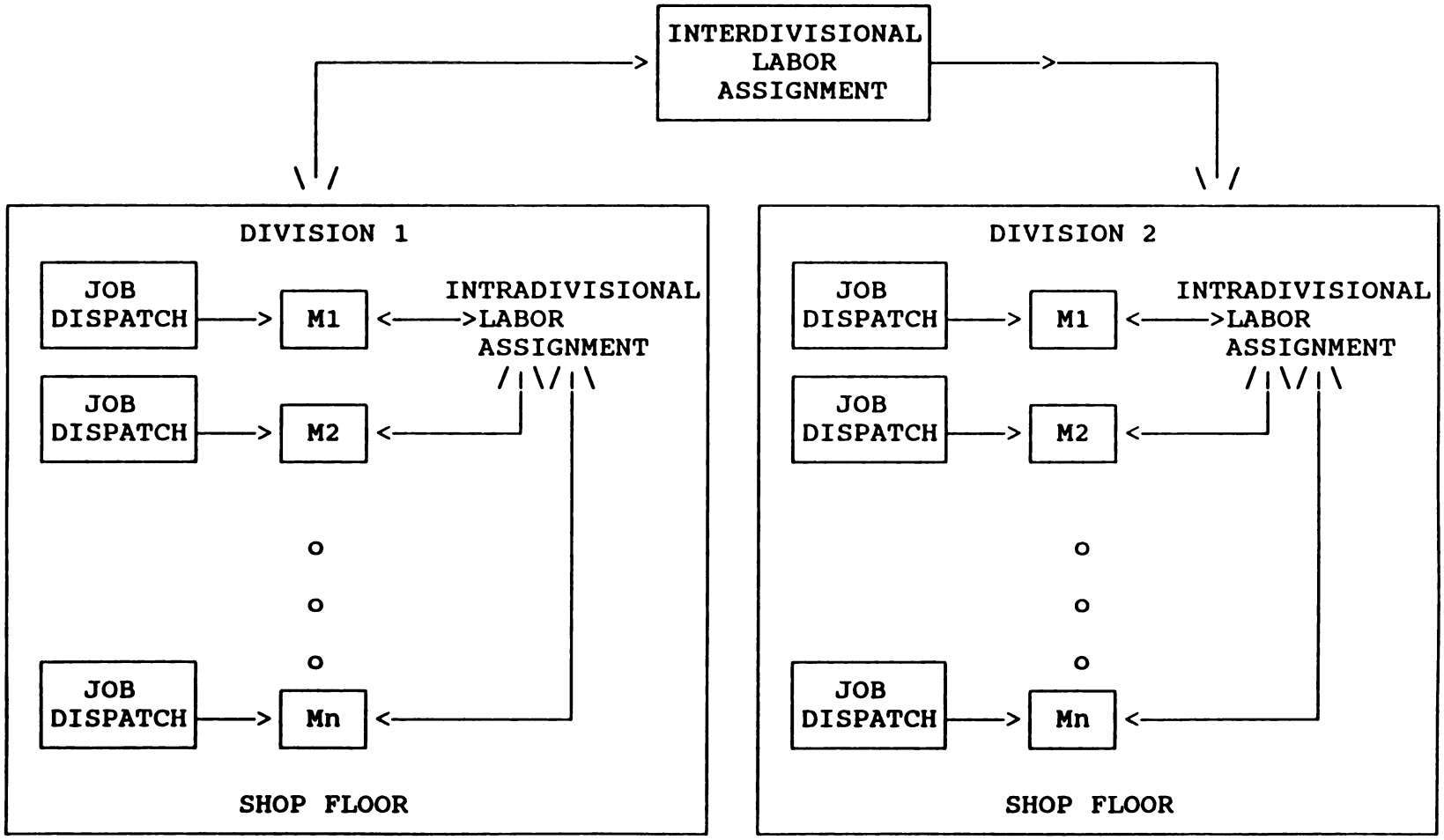
Treleven (1989) reviewed the experimental DRC research and classified the research issues as either research about shop design or operating policy. The research about shop design was concerned with the extent of cross training of the operators, the machine staffing levels, the labor utilization levels, the effect of shop structure (number of echelons), the effect of the shop size, the routing patterns of jobs, and the collection and use of information. Research about the operating policy included research about the due date

FIGURE 2-4
TAXONOMY OF ORR LITERATURE



This taxonomy shows the different types of research conducted in the past. Only the last research at a particular node is shown. "X" indicates that there is no research at a node.

FIGURE 2-5
ILLUSTRATION OF A DUAL RESOURCE CONSTRAINED SHOP



assignment rules, the job dispatching rules and finally the labor assignment decisions.

In addition to experimental research, this review also examined published articles about the implementation of DRC control systems and field research about DRC control systems.

The shop design, operating policies, performance criteria, research methodology and statistical analysis are summarized in the tables beginning with Table 2-7. These tables also show the types of information used in the job dispatch rule, and the "where" and "when" labor assignment rules. The tables also note the type of planning system used.

2.3.1 EXPERIMENTAL RESEARCH

The experimental research about the DRC shop was subdivided into several sections. They are:

- analytical research,
- research about the shop design,
- research about the operating policy and
- research about DRC assembly job shops.

Some of the earlier studies examined both shop design and shop operating policies in the same article. This research was categorized by its predominant focus. The research about assembly job shops was reviewed separately from the other research because different operating policies are examined in these shops.

TABLE 2-7
SUMMARY OF DRC ANALYTICAL RESEARCH

	AVI-ITZHA MAXWELL & MILLER 1965	NELSON 1966	NELSON 1968	TAKACS 1968
SHOP DESIGN				
# OPERATORS	12	1	1	1
# MACHINES	24	2	2	2
# WORK CENTERS	12	2	2	2
# DIVISIONS	3	1	1	1
OPERATORS				
TRANSFER DELAY	N	N	N	N
CROSS-TRAINING	Y	Y	Y	Y
JOB DUE DATE	3TWK	-	-	-
PLANNING SYSTEM		NEG. EXP.	NEG. EXP.	POISSON
OPERATING POLICY				
DISPATCH RULE		in diff equation	FCFS,SOT, FISFS	FCFS
INTERDIVISIONAL	N	N	N	N
"WHERE" ASSIGN.	FISFS	in diff. equation	FISFS,SOT FCFS,LONQ	LONQ
"WHEN" ASSIGN.		in diff. equation	JOB, QUE	QUE
PERFORMANCE MEASUR		WIP	M(N),V(F) M(F), FRACT(F)	M(Q),V(Q)
RESEARCH METHOD.	SIMUL.	DYNAMIC CONTROL	SIMUL.	ANALYTIC
STATISTICAL ANALYS	-	-	-	N

Two studies of DRC shops were reviewed earlier. These were Park and Bobrowski (1989) and Bobrowski and Park (1989) who incorporated an examination of ORR rules into the DRC shop. They are not reviewed again in this section, but their results are noted in the summary.

2.3.1.1 ANALYTICAL STUDIES OF DRC SHOPS

A unique characteristic of a dual constrained system is the prospect of blocking. Labor blocking occurs when there is work in the queue and the operators are idle but cannot be assigned due to a lack of machine availability. Blocking does not lend itself to easy analytical solution. Hence, in a complex system, simulation is the choice of the analyst (Maggard, 1970). However some analytical studies of DRC shops were conducted. These studies were summarized in Table 2-7.

Avi-Itzak et al. (1965) analyzed a shop that had one server for two types of customers. They examined the impact on expected waiting times of switching rules. A switching rule determined when the server should change from performing work with one type of customer to doing work with the other type of customer. An example of a switching rule is requiring the server to complete all jobs in the queue of the current customer before switching to a different type of customer. Their formulations assumed zero setup times. They were not able to extend the analysis to more complex systems.

Takacs (1968) was able to derive a formula for the flow time of a job when there was only one laborer in a two machine system. The formulation assumed no setup and no transfer delays.

Nelson (1966) modeled total work in process inventory costs in a two machine one server system for both a tandem flow shop and a job shop. Nelson (1968) used simulation of a two machine one server shop and compared the results to queuing theory for the machine limited system in an attempt to experimentally develop a DRC queuing theory. However, he concluded that 'no general equivalence relation' exists between the DRC system and queuing theory for the machine constrained system.

2.3.1.2 OPERATING ISSUES

All the authors cited in this section are summarized in Tables 2-8 and 2-9. As defined in chapter one, the "when" labor assignment rule determines when an operator becomes eligible to transfer to a different machine. The "where" labor assignment rule determines which machine an eligible operator is transferred to.

Nelson (1967, 1970) examined the importance of labor cross training, the "when" labor assignment rule and the job dispatching rules. They found that the choice of the "when" rule and the amount of labor cross training influenced shop performance. The job dispatch rules used only SHOP information, and both the "where" and the "when" labor assignment rules used SHOP and LABOR information.

TABLE 2-8
SUMMARY 1: OPERATING ISSUES IN THE DRC SHOP

	FRYER 1973	GUNTHER 1979 & 1981	HOGG, PHILLIPS, ET AL. 1975	HOGG, PHILLIPS, MAGGARD 1977	HOLSTEIN & BERRY 1972	MAGGARD, ET AL. 1973	NELSON 1967
SHOP DESIGN							
# OPERATORS	12	1	1-8	2	10	1-6	1-4
# MACHINES	24	2	2-9	3	40	2-9	4
# WORK CENTERS	12	1	2-9	2	10	2-9	2
# DIVISIONS	3	1	1	1	1	1	1
OPERATORS							
TRANSFER DELAY TIME	N	Y	N	N	N	N	N
CROSS-TRAINING	Y	-	Y	3 LEVELS	Y	Y	2 LEVELS
JOB DUE DATE	-	-			-	-	
PLANNING SYSTEM	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.	8 JOB UN 8 JOB EXP	NEG. EXP.	NEG. EXP.
OPERATING POLICY							
DISPATCH RULE							
SHOP	FCFS, SOT	FCFS, SOT	FCFS	FCFS	SRO,SOT, FISFS	FCFS,SOT	FCFS,SOT, FISFS
NO INFORMATION						RANDOM	
INTERDIVISIONAL	Y	N	N	N	N	N	N
"WHERE" ASSIGN.							
SHOP, LABOR	FISFS, LONQ	PSOT,PLNQ PFCFS	FCFS	DISPATCH	WORK FLOW LONQ	DISPATCH	LONQ, DISPATCH, RANDOM
NO INFORMATION							
"WHEN" ASSIGN.							
SHOP, LABOR	JOB, QUE	IN WHERE, QUE	JOB	JOB	JOB	-	JOB
PERFORMANCE MEASURE	TRAN,M(F) V(F)	M(F),V(F) TRAN	M(LU), M(F),M(W)	M(LU) M(F),M(W)	TRAN,M(F) V(F),M(Q)	M(F),V(F)	M(F),V(F) M(N)
RESEARCH METHODOLOGY	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.	
STATISTICAL ANALYSIS	N	ANOVA	N	-	N	N	

TABLE 2-9
SUMMARY 2: OPERATING ISSUES IN DRC SHOP

	NELSON 1970	ELVERS, TRELEVEN 1985	TRELEVEN, & ELVERS 1985	TRELEVEN 1987	WEEKS, & FRYER 1976	WEEKS, & FRYER 1977	WEEKS, 1979	WEEKS & WINGLER 1979
SHOP CHARACTERISTICS								
# OPERATORS	2	12,18,36	9,12	9	6	12	4-12	4
# MACHINES	4	16,24,48	18	18	12	24	8-24	8
# WORK CENTERS	2	8	9	9	6	12	4-12	4
# DIVISIONS	1	2	3	3	3	3	1-3	1
OPERATORS								
TRANSFER DELAY TIME	N	Y	Y	Y	N	N	N	N
CROSS-TRAINING	N	Y	Y	Y	Y	Y	Y	Y
JOB DUE DATE	-	TWK	TWK		3 TWK, K	3 TWK	3TWK,	3 TWK
PLANNING SYSTEM	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.
OPERATING POLICY								
DISPATCH RULE								
SHOP	FISFS,SOT	FCFS, SOT EDD, CR, SRO	FISFS,SOT EDD,CR, SRO	SOT, SRO	FISFS,SOT SRO	FISFS,SOT SRO	FCFS w/ EFF.	SOT, SRO
INTERDIVISIONAL	N	N	N	N	N	N	N	
"WHERE" ASSIGN.								
SHOP, LABOR	EFF	LONQ	FISFS,CR, SOT,SRO, EDD,LONQ, 5 OTHERS	LONQ,MW/L FCFS	FISFS,SRO LONQ	FISFS	FCFS	FISFS
"WHEN" ASSIGN.								
SHOP, LABOR	JOB,QUE, RANDOM	QUE	QUE	JOB, QUE, PULL	JOB, QUE	JOB, QUE	JOB	JOB
PERFORMANCE MEASURE	M(LU), M(F),V(F)	P(L),M(L) V(L),M(Q) V(Q),TRAN	M(Q),V(Q) M(L),V(L) P(L),TRAN	M(LU) M(F),M(W)	TRAN,M(F) V(F),M(L) V(L),P(L)	TRAN,M(F) V(F),M(L) V(L),P(L)	M(T),M(E) M(T)+M(E)	M(F),M(T) M(E),M(LU M(T)+M(E)
RESEARCH METHODOLOGY	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.
STATISTICAL ANALYSIS		JONCKHEER S,TUKEY B	ANOVA JONCKHEER	JONCKHEER TUKEY B	N	N	ANOVA	ANOVA, TUKEY B

Holstein and Berry (1972) used Nelson's general model to examine the "where" labor assignment rule. The LONQ "where" rule reassigned the worker to the longest queue in the shop. The work flow "where" rule used the work flow in the shop to determine where to place the operator. Both of these "where" rules used SHOP and LABOR information. The "when" rule was JOB, which means that operators were eligible for reassignment at the end of each job. They found that the work flow "where" labor assignment rule reduced the number of labor transfers, but it increased the mean flow time of the job compared to LONQ.

Weeks and Fryer (1977) used Fryer's model without the interdivisional labor transfers to examine the impact of due date rules on the shop. The interaction of due date rules and job dispatch rules had a significant impact on the variance of flow time. They also found that the JOB "when" labor assignment rule was the best rule to reduce flow time.

Weeks (1979) and Weeks and Fryer (1976) examined the development of due date assignment rules to estimate individual job flow times with different shop structures. They found that the shop structure influenced the ability to meet due dates.

Weeks and Wingler (1979) used the same simulation model as Weeks and Fryer (1976) but they incorporated the possibility that the production scheduler could be a risk adverse utility maximizer. They then used stochastic dominance (SD) to order the outcomes of the experiment to account for the probability

distribution of the outcomes. They found that based on SD measures a risk adverse scheduler would never use the SOT dispatching rule.

Treleven and Elvers (1985) examined both static and dynamic shops without interdivisional transfer of labor. They examined the impact of eleven "where" labor assignment rules in a shop with non zero labor transfer times. Their findings were that the choice of the "where" labor assignment rule impacted only the number of labor transfers and had no significant impact on the other performance measures.

Treleven (1987) examined three "when" labor assignment rules: JOB, QUE and PULL. The QUE assignment rule means that a laborer is eligible for reassignment only when all the work in the laborer's current queue is completed. A pull assignment rule initiates the release of labor when there is a need for a laborer at another work center. He found that the "when" labor assignment rule can have a significant impact on the shop performance. All of these labor assignment rules used SHOP and LABOR information. The PULL rule does incorporate the "where" decision into the "when" decision rules to pull workers to the bottleneck work centers.

Maggard, Lesso, Hogg and Phillips (1973, 1976), Maggard, Lesso, Keating, and Wexler (1974) and Hogg, Phillips, Maggard and Lesso (1975a) modeled different examples of a DRC system with the primary purpose of demonstrating how to model labor blocking in DRC shops and to examine the impact of job dispatching rules. Their main emphasis was on the development of a simulation

TABLE 2-10
SUMMARY OF DRC ASSEMBLY SHOP

	ROCHETTE &SADOWSKI 1976	RUSSELL & TAYLOR 1985
SHOP CHARACTERISTICS		
# OPERATORS	13	6
# MACHINES	1,2,4/WC	9
# WORK CENTERS	13	5
# DIVISIONS	4	1
OPERATORS		
TRANSFER DELAY TIME	N	Y
CROSS-TRAINING	SOME	Y
JOB DUE DATE	FLOW	TWK
PLANNING SYSTEM	NEG. EXP.	NEG. EXP.
OPERATING POLICY		
DISPATCH RULE		
SHOP	FCFS,SLK, SRO,FISFS NOP,EDD,	FISFS,SOT EDD,RWK, SP,BS,LP NOP
INTERDIVISIONAL "WHERE" ASSIGN.	N	N
SHOP, LABOR	FCFS	LONQ
"WHEN" ASSIGN.		
SHOP, LABOR	JOB	QUE
PERFORMANCE MEASURES	M(F),M(T)	M(F),M(T) P(T),RMS ASSEM DEL
RESEARCH METHODOLOGY	SIMUL.	SIMUL.
STATISTICAL ANALYSIS	ANOVA	ANOVA

language which could be used to simulate queuing systems with resource limitations.

Hogg, Phillips and Maggard (1977) examined two different "where" labor assignment rules with three different levels of labor cross training. They found that a homogeneous work force is superior to a heterogeneous work force but that much of the benefit of labor cross training can be obtained with a limited amount of cross training.

Fryer (1973) examined the ability of workers to transfer between divisions in a two echelon shop. They found that the interdivisional transfer of labor was the most significant factor in shop performance. Both the intradivisional "when" rule and the interdivisional "when" rule were more important than the interdivisional "where" rule. Also, the intradivisional "when" rule was more important than the intradivisional "where" rule. Job dispatching was important, but often the "when" labor rule was more important than the job dispatch rule.

Gunther (1979, 1981) examined the impact of server transfer delays on the mean flow time using different combinations of "when" and "where" labor assignment rules. The shop had one server and two work centers of one machine each. Gunther concluded that increasing server transfer times increased the mean flow time and the flow time variance and that all labor assignment rules which acted to reduce the number of transfers improved performance.

2.3.1.3 OPERATING POLICIES IN ASSEMBLY JOB SHOPS

An assembly job shop has both serial and parallel operations (Russell and Taylor, 1985). There have been two experiments dealing with assembly DRC shops. They are summarized in Table 2-10.

Rochette and Sadowski (1976) examined the performance of eight different job dispatching rules in a shop with four divisions in which each had their own assembly process. A fifth division assembled only component parts. Workers could work at more than one work center but with a loss of efficiency. The "when" labor assignment rule was JOB and the "where" labor assignment rule was to stay in the current work center if jobs were available, otherwise to transfer to work centers with the highest priority job based on the dispatching rules. They concluded that job dispatching rules had a major influence on flow time and that job tardiness and work force cross training improved performance.

Russell and Taylor (1985) examined eleven different job dispatching rules. The workers had limited cross training but there was no loss of efficiency at any work center that they were trained to operate. The "where" labor assignment rule was LONQ. They found that the SOT job dispatching rule did not perform well. However as the number of components being assembled decreased, SOT's performance improved.

2.3.1.4 SHOP DESIGN ISSUES

The design issues identified by Treleven (1989), include 1) the cross training[1] of the workers, 2) the machine staffing level, 3) the labor utilization levels, and 4) the shop size. This literature is summarized in Table 2-11.

Fryer (1974a, 1974b, 1975 and 1976) extended Nelson's model by using a shop with work centers organized into divisions. The experiments varied the number of divisions and examined intra and interdivisional "when" and "where" labor assignment rules. They found that the interdivisional "when" rules significantly influenced performance of the shop, but that the interdivisional "where" rules had no impact. Also, the intradivisional "when" rules impacted the variance of the shop performance while the intradivisional "where" rules had little impact on the variance of the shop performance.

Maggard, Hogg and Phillips (1980) used a "when" labor assignment rule which reassigned a laborer after the completion of each job. They used a "where" labor assignment rule which assigned an eligible laborer to the machine which had a job in the queue the longest, and another rule assigning labor to its most efficient machine. They examined the impact on shop performance of cross training of the work force. They discovered that cross training of workers helped if operators were always assigned to the work center whose queue contained the job which had waited the longest. However if workers were always assigned to

---

[1]. Cross training of workers determines how many machines a worker can be assigned to operate and what their efficiency will be on each machine.

TABLE 2-11
SUMMARY OF DRC SHOP DESIGN

	FRYER 1974 a	FRYER 1974 b	FRYER 1975	FRYER 1976	MAGGARD ET AL. 1980
SHOP CHARACTERISTICS					
# OPERATORS	12	12	2,4,12	6	2
# MACHINES	24	24	4,8,24	24	3
# WORK CENTERS	12	12	4,12	12	3
# DIVISIONS	3	3	1,3	1,2,3,4,6	1
OPERATORS					
TRANSFER DELAY TIME	N	N	N	N	N
CROSS-TRAINING	Y	Y	Y	Y	4 LEVELS
JOB DUE DATE	-	-	-	-	
PLANNING SYSTEM	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.	NEG. EXP.
OPERATING POLICY					
DISPATCH RULE					
SHOP	FISFS	FISFS	FISFS,SOT	FISFS,SOT	FCFS
INTERDIVISIONAL	Y	Y	Y	Y	N
"WHERE" ASSIGN.					
SHOP, LABOR	FISFS, LONQ	FISFS, LONQ	FISFS, LONQ	LONQ	FISFS EFF.
"WHEN" ASSIGN.					
SHOP, LABOR	JOB, QUE	JOB, QUE 2 INT	JOB, QUE	JOB, QUE	JOB
PERFORMANCE MEASURE	M(F),V(F) TRAN(ID) TRAN(WD)	M(F),V(F) TRAN(ID) TRAN(WD)	M(F),V(F) TRAN(ID) TRAN(WD)	M(F),V(F) TRAN(ID) TRAN(WD)	M(LU), M(W)
RESEARCH METHODOLOGY	SIMUL.	SIMUL.	SIMUL.	SIMUL.	SIMUL.
STATISTICAL ANALYSIS	N	N	N	JONCKHEER	

TABLE 2-12
SUMMARY OF DRC IMPLEMENTATION LITERATURE

	ALLEN 1963	BULKIN, COLLEY & STEINHOFF 1966	HOGG, PHILLIPS, MAGGARD & LESSO 1975a JOB SHOP	HOGG, PHILLIPS, MAGGARD & LESSO 1975a RETAIL	LeGRANDE 1966	STEINHOFF 1966
SHOP CHARACTERISTICS						
# OPERATORS	40	400	3	2	400	400
# MACHINES	78	1000	4		1000	1000
# WORK CENTERS	18	120	4	3	120	120
# DIVISIONS	1	1	1	1	1	1
EARLY COMPLETION OPERATORS		-				
TRANSFER DELAY TIME	N	N	N	N	N	N
CROSS-TRAINING	2 LEVELS	Y	MEDIUM	N	Y	Y
JOB DUE DATE	ACTUAL	ACTUAL			ACTUAL	ACTUAL
PLANNING SYSTEM	ACTUAL	ACTUAL		NEG. EXP.	ACTUAL	ACTUAL
OPERATING POLICY						
DISPATCH RULE						
SHOP	SOT, LOT	SLK		FCFS	SOT,SRO, FCFS,EDD, ESD	EDD
NO INFORMATION	RANDOM				RANDOM	
"WHERE" ASSIGN.						
SHOP, LABOR	DISPATCH	SLK	FCFS,EFF	FCFS	DISPATCH	EDD
"WHEN" ASSIGN.						
SHOP, LABOR	JOB	QUE		JOB	JOB	JOB
PERFORMANCE MEASURE	M(LU)	TARDINESS	FLOW		NJC,PLT MFT,VFT MQL,MQT CCQ,MU,LU	
RESEARCH METHODOLOGY			SIMUL.	SIMUL.	SIMUL.	SIMUL.
STATISTICAL ANALYSIS			-	-	-	

the work center where they were the most efficient, cross training of the workers had little influence on shop performance.

Maggard et al. (1973), Elvers and Treleven (1985), Treleven and Elvers (1985), Hogg et al (1975), Fryer (1975) and Weeks (1979) examined staffing levels in shops. They found that staffing levels of 50% to 75% (e.g. the number of operators is 50% the number of machines) were the most efficient because staffing above this level resulted in increased labor idleness.

Researchers demonstrated that shop performance improves rapidly as cross training increased. Nelson (1967), and Park and Bobrowski (1989) also demonstrated that cross training of workers produced large improvements in shop performance.

2.3.1.5 DESCRIPTIVE RESEARCH ABOUT DRC SHOP

This research is summarized in Tables 2-12 to 2-13. This work is important for one of two reasons. First, some of this work (e.g. Allen, 1963) developed a model of the DRC shop based on field studies. Second, some of this work applied models developed with simulations to actual shops (e.g. Hogg et al., 1975a).

TABLE 2-13
SUMMARY OF DRC DESCRIPTIVE RESEARCH LITERATURE

LABOR MACHINE CONSTRAINED JOB SHOP	SERVICE SECTOR
REITER (1966) HARRIS (1965)	BEGLEY, KAINEN & MAGGARD (1983)

Allen (1963) simulated a large DRC job shop at General Electric. He evaluated the importance of alternative routings, cross training of labor, "where" labor assignment rules and job dispatching rules. He concluded that cross training and alternative routings were the best means to improve labor utilization and that job dispatching had relatively little effect on shop performance.

Harris (1965) examined a job shop with 160 machines and a smaller set of laborers who were assigned to different machines as needed. He rejected the hypothesis that the job arrival distribution was poisson or that the service times were exponential. He proposed that job shops are actually dual resource constrained shops but did not develop a model to continue this study.
Bulkin, Colley and Steinhoff (1966) examined and simulated the operation of Hughes Aircraft Company. This work helped establish the basis in the literature for the need to examine DRC shops. The company had 1,000 machines and 400 workers. They did not develop any unique scheduling rules for the DRC environment.

LeGrande (1966) also examined and simulated a division of Hughes Aircraft Company which was labor and machine constrained. He was more interested in developing a simulation than in testing labor assignment rules, but again, this work served as justification for later work on DRC shops.

Hogg et al. (1975a) gives two examples of actual DRC shops. They tested the labor assignment rules in both shops using a simulation of conditions in the

shops. They found the simulations to be effective aids in assessing the benefits of cross training and staffing levels.

2.3.5 SUMMARY OF DRC RESEARCH

Past efforts to use analytical methods to study the DRC shop has not yielded much insight. Most research into the DRC shop has relied on simulation of the shop and statistical analysis of the output.

The research does indicate that the "when" rules have a major impact on system performance and that the "where" rules examined do not significantly impact system performance in a generalized random job shop (Treleven, 1989).

The three "when" rules that have received the most study are: 1) JOB, 2) QUE, and 3) PULL. JOB, QUE, and PULL all use LABOR and SHOP information, but there are important differences in their operation. JOB results in a larger number of transfers than QUE; QUE gives the manager less control over transfers than JOB; and, PULL is an intermediate rule between JOB and QUE. The PULL rule compares the ratio of laborers at a work center to jobs in the queue at the work center to determine when a laborer is eligible for a transfer. Studies of the "when" rule found that in a shop without labor transfer delay that JOB is the best performing rule (Treleven, 1989). However Gunther (1979) and Treleven (1987) found that in shops with a delay in labor transfer the PULL and QUE rules perform better than the JOB rule.

2.4 SUMMARY OF LITERATURE REVIEW

Figure 2-6 attempts to capture the significant aspects of the ORR and DRC literature to date. Six variables discussed in the literature are shown on the left. These variables are:

- Load leveling in ORR rule;
- Load leveling by the planning system;
- Labor cross training;
- Job dispatching rules;
- Labor staffing level;
- and the "when" labor assignment rules.

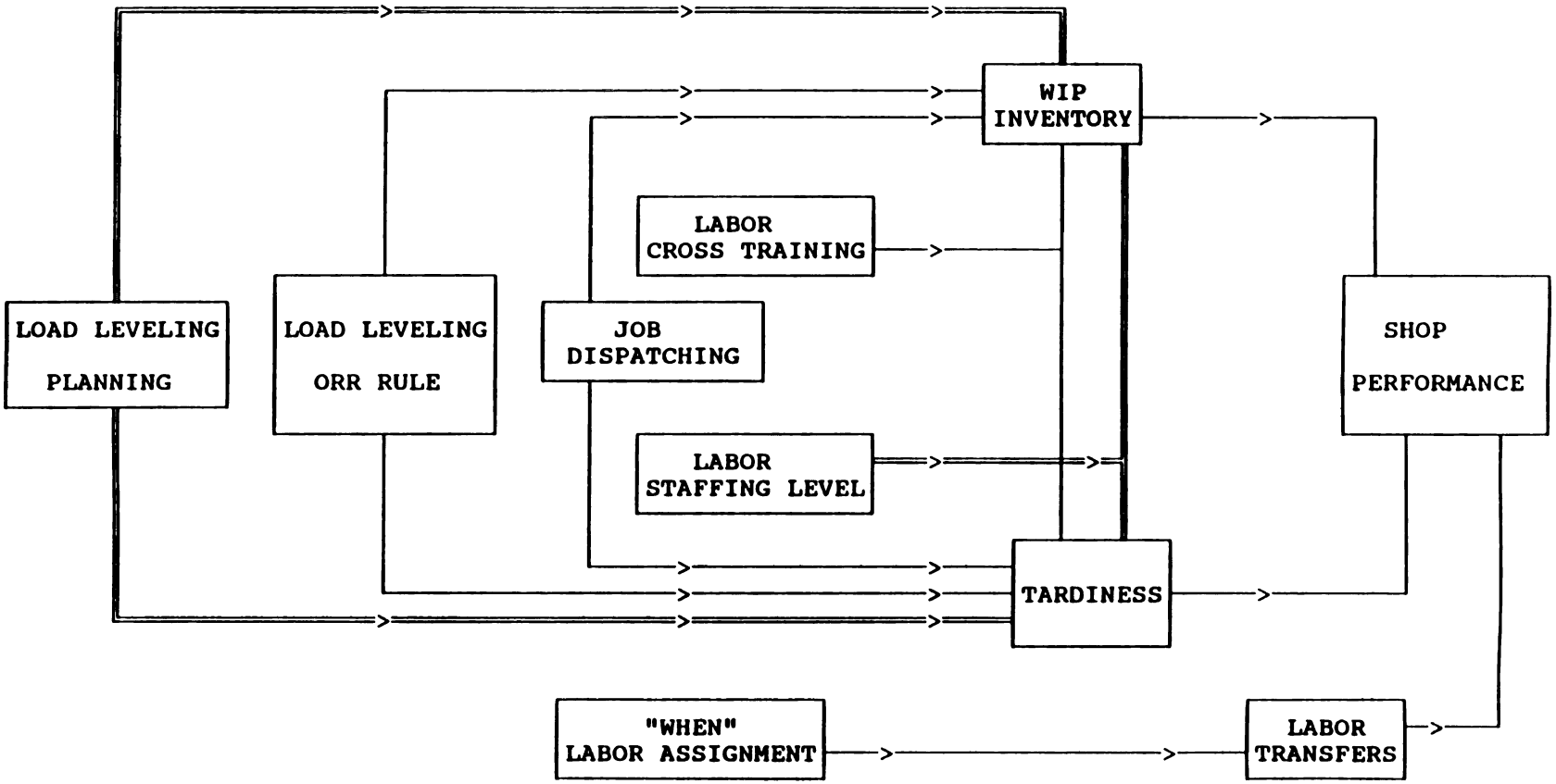
There are four outcome variables shown on the right. They are:

- Total inventory;
- Tardiness;
- The number of labor transfers;
- and, the shop performance.

The first three of these variables are intermediate outcomes. This means that the changes in total inventory, tardiness, and the number of labor transfers influence the shop performance directly and the input variables effect shop performance through these intermediate variables. So, shop performance is actually a composite of these intermediate measures.

Each of these six input variables is discussed individually.

FIGURE 2-6
SUMMARY OF ORR RESEARCH IN A DRC SHOP



- Load leveling by the planning system influences both tardiness and inventory levels. There has been little investigation of this issue.
- Load leveling by the ORR rule decreases the level of inventory in the system but can also increase the amount of tardiness.
- The labor staffing level influences both the inventory level and tardiness because as the number of operators increase, they tend to interfere with each other more often.
- The level of operator cross training has a major influence on the amount of inventory and the amount of tardiness.
- The "when" labor assignment rule influences the number of labor transfers which influence the performance of the shop.
- The choice of the job dispatching rule influences the inventory level and the tardiness performance of the shop.

Researchers are including more variables that represent decisions managers must make into their research model of the job shop. Ragatz and Mabert (1988) included the ORR rule, with finished goods inventory and cost measurements of the shop performance. They found that in all cases the SFC system with ORR performed as well as or better than the SFC system without ORR. When Bobrowski and Park (1989) and Park and Bobrowski (1990) introduced a labor constraint into job shops, they found that those shops using ORR clearly performed better than shops without ORR mechanisms.

Table 2-14 shows the information content of the ORR rule, job dispatch rule, and labor assignment rules. These are the SFC rules shown in Figure 2-6. The other input variables in Figure 2-6 (labor staffing level, labor cross training and the planning system) are not rules, but are part of the shop design issues. In examining Table 2-14, it is clear that none of the research to date has used all three sources of information in the shop: POOL, SHOP and LABOR.

This research examines the use of SHOP, POOL and LABOR information by the ORR rules and the "when" labor assignment rules. It also examines the use of SHOP and POOL information in the job dispatching rule. The experimental design to do this is explained in the next two chapters.

TABLE 2-14
INFORMATION CONTENT OF SHOP FLOOR CONTROL RULES

TYPES OF INFORMATION	ORDER REVIEW/ RELEASE MECHANISM	JOB DISPATCH RULES	"WHEN" LABOR ASSIGNMENT	"WHERE" LABOR ASSIGNMENT
SHOP, POOL, LABOR				
SHOP, POOL	FFL,MIP,NES,WLS, AGGWINQ,WCEDD,MIL, BFL,LC,VECTSIM,DIR			
SHOP, LABOR			QUE,JOB,PULL,INT	LONQ,FISFS,SOT,EFF, FCFS,WORK FLOW,EDD, PSOT,PLNQ,PFCFS,SRO SLK
SHOP	MNJ	FCFS,SOT,CR,MODD, SRO,EDD,CEDD,FISFS SLK,NOP,RWK,SP,BS LP,LOT,ESD		
POOL	FCFS,BIL,CR			
NO INFORMATION	IMM	RANDOM	RANDOM	RANDOM

CHAPTER 3

3.1 INTRODUCTION TO THE RESEARCH METHODOLOGY

The method used to generate the experimental data was computer simulation of a hypothetical dual resource constrained job shop. This hypothetical shop was subjected to a series of experiments and the output was analyzed using ANOVA, a priori hypotheses and post hoc multiple comparisons.

Seven aspects of the research methodology are examined in this chapter:

1. The selection of simulation as the research methodology is justified.
2. The validity of this research is examined.
3. The shop environment is described. This includes an explanation of the research assumptions and a description of the shop operating rules.
4. An overview of how SIMAN, the simulation language used in this experiment, processes the simulation code.
5. Three simulation design issues are analyzed in detail. They are:
    - Verification of the simulation.
    - Calculation of the batch mean size.
    - Removal of initialization bias.
6. The performance measurements which were used in the experiment are examined.
7. The data collection techniques used in this experiment are explained.

Each of these seven aspects of the research is examined in turn.

3.2 JUSTIFICATION OF RESEARCH METHODOLOGY

The system being studied is very complex. Prior attempts to examine this problem using analytical approaches have not been successful (e.g. Takacs, 1968). Maggard et al. (1980) suggested that because this system allows labor blocking to occur, it would not be feasible to examine the problem analytically. Ragatz (1985, p.29) stated that an analytical solution was not appropriate for this type of problem because some of the control rules require real time information. For these reasons, computer simulation was selected as the research methodology.

The study was conducted using a discrete event simulation model of the shop. The simulation was written using SIMAN (Pegden, 1987). SIMAN is a FORTRAN based simulation language which contains subroutines for event timing and the statistical calculations. This special purpose simulation language was used to reduce the model development time and to reduce the possibility of errors resulting from the record keeping aspects of the simulation.

3.3 EXPERIMENTAL VALIDITY

As with all experiments it is important to ensure that this research methodology was valid. Cook and Campbell (1979, p.39) list four major types of validity that the researcher must be concerned about. They are:

1. Statistical conclusion validity is concerned with whether the sample size is large enough to detect an effect and whether covariation really exists at a given α level (Cook and Campbell, 1979, p.41).
2. Internal validity is concerned with whether there is a causal relationship exists between the variables (Cook and Campbell, 1979, p.50).
3. Construct validity is concerned with whether the observed treatment effect is actually due to other constructs in the shop (Cook and Campbell, 1979, p.59).
4. External validity is concerned with the ability to generalize the results of this experiment to particular shop environments and across shop environments (Cook and Campbell, 1979, p.71).

3.3.1 STATISTICAL CONCLUSION VALIDITY

Using a simulation of a hypothetical DRC job shop could effect the ability to infer that there is covariance between variables in four ways. They are:

- Elimination of threats due to the unreliability of the treatment implementation because all treatments are implemented in the same way (Cook and Campbell, 1979, p.43).
- Eliminate threats due to random features in the experimental setting (Cook and Campbell, 1979, p.44) by limiting randomness to the four random number streams.
- Low statistical power was prevented by obtaining a sample size large enough to detect effects. As will be discussed later, a pilot run was done to estimate the variance and the half width of the effect. The necessary sample size was calculated using the data in Table 3-1. The largest variance was 72.18 with a standard deviation of 8.5. Using a half width (d) of 2 the following calculations were made:

$$x - d \leq \mu \leq x + d$$

so that $d = z_{(1-\alpha/2)}\, s/\sqrt{N} = 10$

where N = sample size

z = standard normal deviate = 1.96

$\alpha = .05$

Substituting into the formula above for s=8.5 and d=2 gives

$$N = (1.96)^2(s^2)/d^2$$

$$= (1.96)^2(8.5)^2/(2)^2 = 69.3$$

TABLE 3-1
RUNS TO DETERMINE SAMPLING CONDITIONS

PERFORMANCE MEASURE IS FLOW TIME
MIL & CMS SETTINGS: FACTOR 1 = 5; FACTOR 2 = 2
MLC SETTING: FACTOR 1 = 1. DUE DATE FACTOR = 9.50
SMOOTH PLANNING SYSTEM

ORR RULE	WHEN RULE	JOB DISP RULE	NUMB JOBS TRUNC	INDEP BATCH SIZE	NORM BATCH SIZE	VARIAN FLOW TIME	MEAN FLOW TIME
1	1	1	100	50	50	4.660	27.110
1	1	2	601	50	100	3.080	27.453
1	1	3	100	50	50	3.980	25.089
1	2	1	100	50	50	5.080	27.509
1	2	2	851	50	50	4.930	28.046
1	2	3	100	50	50	4.300	25.050
1	3	1	100	50	50	4.570	27.612
1	3	2	500	50	50	4.380	28.192
1	3	3	100	50	50	3.070	22.751
2	1	1	1452	50	50	72.180	34.230
2	1	2	3701	50	400	6.240	38.772
2	1	3	100	100	100	53.760	31.742
2	2	1	100	100	100	64.860	35.321
2	2	2	500	50	400	4.220	39.378
2	2	3	100	50	50	56.110	30.318
2	3	1	100	100	800	3.390	40.748
2	3	2	500	100	100	22.220	39.063
2	3	3	100	100	800	12.700	35.725
3	1	1	100	50	50	23.810	30.655
3	1	2	4001	50	800	6.840	32.094
3	1	3	100	50	50	22.120	30.863
3	2	1	150	50	50	24.500	31.007
3	2	2	500	50	200	4.870	30.744
3	2	3	100	50	50	21.760	30.726
3	3	1	100	50	50	25.280	30.929
3	3	2	500	50	800	2.710	31.588
3	3	3	125	50	50	24.760	31.080
4	1	1	100	50	400	.670	37.020
4	1	2	500	50	400	.590	37.260
4	1	3	100	50	100	2.560	37.244
4	2	1	100	50	100	2.840	37.362
4	2	2	500	50	50	3.000	37.499
4	2	3	100	50	100	1.860	37.044
4	3	1	100	50	400	.660	36.991
4	3	2	500	50	200	1.460	37.834
4	3	3	100	50	100	2.520	37.240

TABLE 3-1 (cont'd)

PERFORMANCE MEASURE IS FLOW TIME
MIL & CMS SETTINGS: FACTOR 1 = 5; FACTOR 2 = 2
MLC SETTING: FACTOR 1 = 1. DUE DATE FACTOR = 9.50
EXPONENTIAL PLANNING SYSTEM

ORR RULE	WHEN RULE	JOB DISP RULE	NUMB JOBS TRUNC	INDEP BATCH SIZE	NORM BATCH SIZE	VARIAN FLOW TIME	MEAN FLOW TIME
1	1	1	1977	100	100	31.360	29.980
1	1	2	1751	100	4600	.140	29.349
1	1	3	477	100	100	33.160	28.053
1	2	1	100	1600	1600	26.180	32.677
1	2	2	1901	100	1600	13.760	33.826
1	2	3	152	100	100	15.250	27.107
1	3	1	100	100	100	22.490	29.735
1	3	2	500	100	2400	25.680	35.149
1	3	3	100	100	100	26.270	27.930
2	1	1	1575	50	400	8.380	23.683
2	1	2	500	100	800	20.420	43.377
2	1	3	275	100	2400	15.900	24.174
2	2	1	325	50	800	6.300	23.191
2	2	2	1751	100	800	24.640	43.837
2	2	3	300	100	2400	14.190	24.037
2	3	1	100	100	800	12.340	25.392
2	3	2	851	100	800	14.040	42.969
2	3	3	175	100	800	9.470	22.419
3	1	1	277	25	50	20.790	29.131
3	1	2	100	50	50	17.870	29.211
3	1	3	152	25	50	20.090	29.275
3	2	1	100	25	50	20.530	29.282
3	2	2	500	50	800	4.560	33.299
3	2	3	100	25	200	4.200	29.681
3	3	1	100	25	800	2.390	30.896
3	3	2	500	50	800	9.650	34.364
3	3	3	100	50	100	6.690	29.106
4	1	1	100	50	50	4.450	35.336
4	1	2	100	50	50	4.740	35.431
4	1	3	100	50	200	1.360	35.673
4	2	1	100	50	50	5.260	35.640
4	2	2	100	50	50	5.020	35.520
4	2	3	100	50	100	2.270	35.405
4	3	1	100	50	50	4.960	35.662
4	3	2	100	50	400	1.140	36.142
4	3	3	100	50	50	5.310	35.527

The sample size of 69.3 was taken as the lower bound on the sample size. The experiments actually used a sample size of 100 to ensure there would be adequate power.

- Violation of statistical assumptions is possible using simulation. The data was analyzed using ANOVA and pair-wise multiple comparisons. The assumptions are homogeneity of variance, normal residuals and independent samples. Before conducting any analysis the data was examined to ensure that these assumptions were met.

There are two real threats to the independence of the samples that are acute when using simulation. The first is that there is autocorrelation in the sample. The actions taken to prevent this are discussed later. The second is the use of common random number streams to reduce variance (Kleijnen, 1974; Law and Kelton, 1982).

To maintain independence in the sample as required for the analysis of variance, one random number steam was different for each treatment as suggested by Mihram (1974). This allowed the use of a repeated measures ANOVA model for analysis as will be discussed in chapter 5.

3.3.2 INTERNAL VALIDITY

Using simulation eliminates many common threats to internal validity. These threats are history, instrumentation, mortality, diffusion of treatments, and compensatory equalization of treatments (Cook and Campbell, 1979, pp.51-55). Each is discussed below:

- Simulation prevents the occurrence of an exogenous event in the shop while data is being collected. So, simulation prevents threats due to history.
- The possibility that the instrumentation or performance measurements could change between observations is controlled by the simulator.
- The occurrence of mortality because a subject (e.g. work center or shop) drops out of the experiment is controlled by the simulator.
- Diffusion of treatments is prevented in the simulation by coding the treatments correctly.
- Compensatory equalization of treatments cannot occur because the simulator writes the code for the experiment.

All of these threats to internal validity are controlled by simulating a hypothetical job shop. This results in high internal validity.

3.3.3 CONSTRUCT VALIDITY

Construct validity is threatened by the presence of confounding variables. Confounding occurs if a particular cause and effect construct can be explained in terms of another construct (Cook and Campbell, 1979, pp.64-70). The threats to construct validity could occur in a hypothetical job shop are:

- Inadequate pre-operational explication of constructs, which is not affected by the use of simulation. Chapter 2 contains a careful elucidation of constructs from the past literature to prevent this problem.
- That generalizability is restricted across constructs. This is a real threat that exists when using simulation. It occurs because the simulation models only part of the shop, so that the effect of the treatments on other aspects of the shop are not measured.

Consequently, construct validity is a concern of the simulator. To obtain construct validity in this study, effort has been taken to collect the relevant measures and to use the relevant constructs from the prior literature.

3.3.4 EXTERNAL VALIDITY

Statistical interactions are a threat to the external validity of the experiment. If there is an interaction then the result cannot be generalized across classes involved in the interaction (Cook and Campbell, 1979, p.73). Significant interactions between the main variables will be identified with statistical tests.

However, possible interactions of the setting and the treatments are a threat to a simulation experiment. The question of whether the setting of this experiment in a DRC job shop of limited size can influence the outcome is now addressed.

Past researchers have found that hypothetical studies of the job shop are relatively insensitive to many differences between the setting of the hypothetical shop and the setting of a real shop. These differences include the structure of the shop (number of divisions), the size of the shop (number of work centers), the number of machines in each work center, and the pattern of the work flow in the shop (Baker and Dzielinski, 1963; Fryer, 1974a; Fryer, 1975; Weeks, 1979). Also, researchers have found that shop performance is insensitive to changes in the job routings (Elvers and Treleven, 1985).

3.3.5 SUMMARY OF DISCUSSION ABOUT VALIDITY

There is a countervailing relationship among these types of validity. Using a particular research methodology to increase one type of validity tends to decrease other types of validity. The priority among the validity types depends on the purpose of the research. Since this research is orientated towards theory development, it is more important that the methodology provide statistical power than high external validity (Cook and Campbell, 1979, p.83). Consequently, simulation of a hypothetical job shop is an appropriate research methodology for this study.

3.4 DESCRIPTION OF THE SHOP ENVIRONMENT

The shop environment is described in three sections. First some key definitions are given. Second, the key assumptions made in the design of the shop are described. And, third the shop operating rules are described.

3.4.1 KEY DEFINITIONS

Two definitions were used repeatedly in the shop operating rules and in the experimental variables. They were the definition of criticality and operation due dates. Each definition is addressed below.

3.4.1.1 MACHINE CRITICALITY

The purpose of the term criticality is to identify those work centers which have a larger work load than the other centers. Criticality is a relative measure, because it contrasts the load of one work center against another work center's load. The load of an individual work center is calculated by summing all the work which is scheduled for the work center but is not completed. This total work load includes work in the shop and the work in the pool. All the work centers are then ranked from the largest load to the smallest load. The work center with the largest work load is the most critical work center. The work center with the smallest work load is the least critical work center. The work center criticality is calculated at the beginning of each shift (i.e. every eight hours).

The concept of criticality is used in this research to identify potential bottlenecks in the job shop. Recent research (Umble and Srikanth, 1990) suggests that controlling the flow of work through bottlenecks or potential bottlenecks is an important control point for job shop managers.

The concept of criticality is used in the following experimental variables:

- The Critical Labor Release (CLR) "when" labor assignment rule;
- the Critical Machine Selection (CMS) ORR rule;
- the Modified Load Control (MLC) ORR rule; and,
- the Critical Machine Due Date (CMDD) rule.

These experimental variables are explained in detail in chapter 4.

3.4.1.2 CALCULATION OF OPERATION DUE DATES

Operation due dates (ODD) were established using the same method as Park (1987,p.55). First, the proportion of work not completed on the job was multiplied by the flow allowance for the job. Second this result was added to the release date of the job. The flow allowance (FLOW) was the due date of the job minus the release date of the job.

$$FLOW = DD_{JOB} - RD_{JOB}$$

$$ODD = (PROPORTION\ WORK\ COMPLETED)*FLOW + RD_{JOB}$$

The ODD were calculated as needed for the modified operation due date (MODD) dispatching rule and the critical machine due date (CMDD) dispatching rule. Both of these experimental variables are explained in great detail in chapter 4.

3.4.2 RESEARCH ASSUMPTIONS

This hypothetical job shop had both labor and machine constraints. The shop had six different work centers, each with two identical machines as previously shown in Figure 1-3. Serving those work centers were six operators who were all equally efficient at each work center.

Once a job started processing on a machine it continued processing until completed. Several assumptions were made about the availability of information in the shop:

- First it was assumed that the information was complete. The only uncertainty in the shop was the exact amount of work arriving from the shop.
- Second, it was assumed that all information was accurate.
- Third, it was assumed that all information was current.
- Finally, it was assumed that all information was immediately accessible.

There were only three sources of information in the shop as illustrated in Figure 1-4. It was assumed that all of these sources of information (POOL, SHOP and LABOR) were immediately available everywhere in the shop.

Other assumptions included: no set up, no breakdowns, no preemption, no lot splitting, no labor transfer delay, no material transfer time, and no alternative routings.

3.4.3 SHOP OPERATING RULES

The average shop workload was set so that 45 percent of machine capacity was utilized and 90 percent of labor capacity was utilized. The shop operated one shift a day (8 hours per shift) and five days equal one week (40 hours). This was consistent with past research by Bobrowski and Park (1989), and Ragatz and Mabert (1988).

Each work center had one queue and two identical machines. Jobs could not be processed until a laborer was assigned to a machine, and only one laborer could be assigned to a machine. Once a laborer was assigned to a free machine at a work center, the job dispatch rule was used to select a job from the queue to begin processing on the machine.

When the job completes processing, it transfers immediately to the queue of the next machine on its route or to FGS inventory if it has finished processing at each machine on its route.

The shop operating rules are discussed in four categories:

1. Order creation in the planning system.
2. The ORR control point.
3. The Labor Resource Management.
4. The job dispatching rules,

Each of these is described separately below.

3.4.3.1 ORDER CREATION IN THE PLANNING SYSTEM

The routing for the job through the shop is set during the planning stage and is not subsequently altered. The routings are generated randomly without replacement, so that a job cannot visit a work center more than once. Each job has an equal probability of visiting any work center. The number of operations assigned to a job is drawn from an uniform distribution with a minimum of 3 and a maximum of 5.

Each job was assigned a due date based on the number of operations[1] in the job. A similar method was previously used by Ragatz and Mabert (1988). Furthermore, its use has been recommended by Orlicky (1975, p. 83). The due date was set so that 20% of the jobs were tardy with the following experimental settings: UNCONTROLLED planning system; immediate release of jobs to the shop floor; reassignment of the machine operator only when there were no jobs left in the queue; and, the dispatch rule was first-come-first-served (FCFS). These experimental variables are explained in detail in chapter 4. This corresponds to the tight due date setting in Ragatz (1985).

The setting of the parameter used in the due date rule to ensure that 20% of the jobs were tardy was obtained with the Fibonnaci response surface methodology algorithm. The code for this algorithm is in Appendix B. Fibonnaci is a search algorithm which searches for the optimal value for a single parameter by

---

[1]. $DD_j = TARR + k*NOP_j$; where DD_j = due date of job j; TARR = time arrival in the shop, which is the first day of the week for all jobs; k = planning factor; NOP_j is the number of operations of job j.

running the simulation to obtain a set of points. The algorithm selects a new point to improve this set of points and continues to select points until it finds no further improvement.

All processing times and order due dates were assigned when the job was created by the planning system. The processing time for each operation was drawn from a negative exponential distribution with a mean of 1.0. This assumption has been investigated in prior research and it has been found that the service distribution is not critical when comparing scheduling rules (Baker, 1974, p.215).

3.4.3.2 ORR CONTROL POINT

Using the ORR rules, a job was released to the gateway work center (first work center on its route). The ORR control point received jobs from the planning system once a week or every 40 hours. Prior research has not indicated that the timing convention has a major effect on the shop performance (Melnyk and Ragatz, 1989), so a continuous timing convention was used. The trigger mechanism and the job selection mechanism were experimental variables that are discussed in chapter 4.

3.4.3.3 LABOR RESOURCE MANAGEMENT

Workers are assigned to machines using two labor assignment rules. The "when" labor assignment rule determines when a worker is to be released from one machine and returned to the labor pool. The "where" labor assignment rule determines to which machine an operator in the labor the pool is assigned. The

"when" rule is an experimental variable and is explained in detail in chapter 4. Past research found the "where" rule to have no significant influence on shop performance (Treleven, 1989), so the same "where" rule was used for each treatment.

The "where" rule used in this research is called the critical labor assignment (CLA) rule. The CLA assigns labor to the most critical machine which has work in its queue. If machines are tied in terms of criticality (defined below), the laborer is assigned to the machine with the most work in its queue. This "where" rule is very similar to the longest queue rule used by Nelson (1978), which assigns a laborer to the machine with the most work in the queue. Laborers are transferred from the machine to the labor pool, and from the pool back to a machine without incurring a transfer delay. However, there is a transfer cost incurred for each transfer of the laborer from the labor pool to a machine. This is consistent with prior research by Bobrowski and Park (1989).

3.4.3.4 JOB DISPATCHING RULES

The job dispatching rules control the progress of the job through the shop after it is released by the ORR rule. This is an experimental variable and the details of the dispatching rules used are explained in chapter 4.

When a job is finished processing on a machine it transfers to either finished goods (FGS) inventory or to the next machine on its route with no transfer delay. This practice assumes a make-to-order shop with forbidden early shipments. It also reflects the reality of many actual job shops which are

prohibited from shipping their orders early. This restriction was found in past research to change the relative performance of some job shop operating rules (Kanet and Christy, 1984).

3.5 PROCESSING OF THE SIMULATION CODE

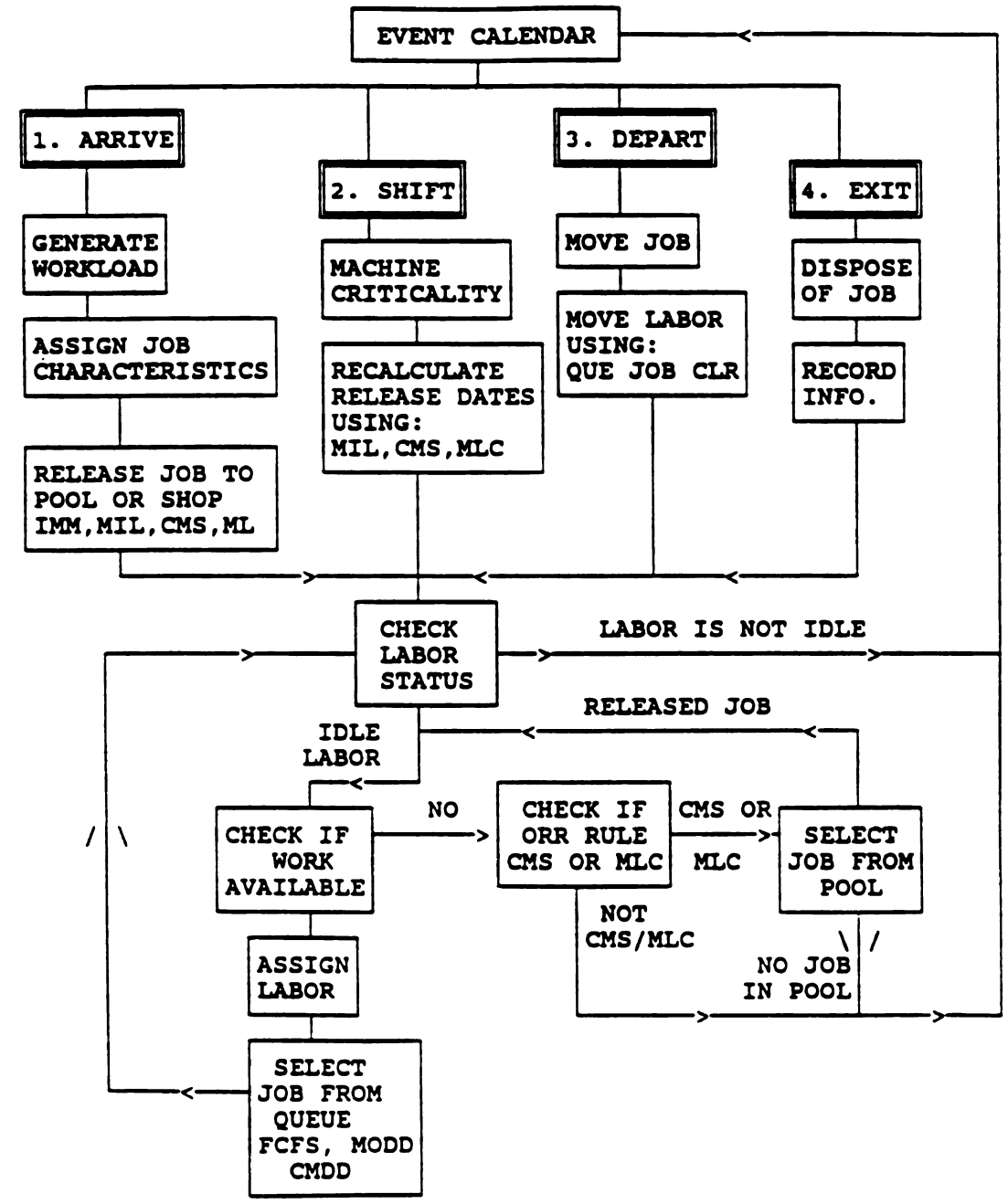
The simulation is written in SIMAN (Pegden, 1988) with user written subroutines in fortran. SIMAN simulates a sequence of activities by moving entities through a series of events (Pegden, Shannon, Sadowski, 1990, p.62). This section briefly describes how jobs flow through a series of events during the course of the simulation.

Before describing these events, a brief overview of the code is provided. The simulation consists of three different sections of code. These are:

- The SIMAN Model code.
- The SIMAN Experiment code.
- The user written fortran subroutines.

The SIMAN Model code is shown in Appendix B, part 1. This block of code is required in SIMAN to execute the program (Pegden, 1987, p.45). The SIMAN Experiment code is shown in Appendix B, part 1. It contains the elements for each treatment (e.g. establishes which planning system and which ORR rules are to be used etc.). It also sets default values, and specifies the counters, tallies and time persistent variables for record keeping (Pegden, 1987, p.47). The user written fortran subroutines are in Appendix B, part 2. These contain

FIGURE 3-1
FLOWCHART OF SIMULATION CODE



═══ INDICATES AN EVENT ON THE EVENT CALENDAR

the code for each event and for the logic of the control system used in the experiment.

An abbreviated flow chart in Figure 3-1 shows the four events that each job flows through with double boxes. These events are:

- Arrive Event.
- Shift Event.
- Depart Event.
- Exit Event.

There are other events not listed which are used to control the simulation. These events are not discussed here because they do not involve the jobs. A complete list of events is shown in Appendix A, part 2.

Two other aspects of the simulation are explained before discussing the four events above. First, the X() array variables and their role in controlling the simulation processing and setting the experimental parameters is discussed. Second, a section of the simulation logic that is used in all the events is explained.

3.5.1 X() ARRAY

All the X() array variables are shown in Appendix A part 2. These variables are global and are not related to a specific entity (Pegden, 1987, p.20). The left hand column of Appendix A shows the array location. The middle

column gives a description of the variable. The third column shows the fortran subroutine or SIMAN block where the value of this variable is changed.

The X() variables are used to determine which level of experimental factors are being run, to set the value of parameters used in the experiment and to collect historical information about the status of the shop.

3.5.2 COMMON LOGIC IN ALL EVENTS

At the end of every event the status of the operators is checked. If an operator is idle, the "where" labor assignment rule is used to assign the operator to a work center if possible. If the operator cannot be assigned to a work center, a check is made to see if the ORR rule is either CMS or MLC. Both the CMS and the MLC rules hold jobs in the order pool until there is idle operator. If there are jobs in the order pool the CMS and MLC rules will examine the pool and select a job for release to the shop floor. If all the operators are busy or if there is no work to assign to an idle operator, the simulation finds that no idle laborer (if any) can be assigned to a machine, the simulation returns to the event calendar to process the next event on the calendar.

3.5.3 ARRIVE EVENT

The arrive block represents the output of the planning system. This event is run every 40 hours (one week) of simulated time. This event first determines the number of jobs to create (determined by the type of planning system) and

then assigns characteristics to each job. The jobs are then sent to the order pool where a check is made to see if work should be released to the floor. The release decision is made by the ORR rule in use. The common logic described above is used to assign any idle operators and the event ends.

Work flows through the simulation model starting with the creation of a set of jobs. This job creation including the determination of the job characteristics, and the release of the job to the order pool occurs in the Arrive event.

The characteristics of the job are described by its attributes. In SIMAN, the attributes are assigned to individual entities which in this simulation represent jobs (Pegden, 1987, p.19). So each job carries these attributes with it the entire time its being processed.

In this model, each job is assigned 20 attributes (the attribute structure is shown in Appendix A). This exhibit describes each attribute in the middle column across from the attribute name (e.g. A(1)). The right hand column shows the subroutine where the attribute is either set or changed.

The attributes include static job characteristics such as routings, operation processing times, number of operations, and due dates. It also includes dynamic information about jobs such as current operation, release dates, time started in the system, remaining processing time and the time the job started on the floor.

Once the job is released from the order pool, its progress through the shop is controlled by other events in particular the Depart event.

3.5.4 DEPART EVENT

The Depart event is scheduled whenever a job begins processing on a machine. The Depart event represents those activities that occur when a job finishes processing. These activities include the freeing of the machine, transferring the job, selecting a job from the queue for processing with the job dispatching rule, and checking to see if the operator should be reassigned. Jobs are transferred either to the queue of the next work center in their routing or to finished goods inventory. The decision to reassign labor is made using the appropriate "when" rule. If the operator is eligible for reassignment the "where" labor assignment rule determines where to reassign the operator.

At this point the subroutine uses the common logic described above and the event ends. The next event controls activities which occur at the beginning of each shift (every 8 hours).

3.5.5 SHIFT EVENT

The Shift event represents the beginning of a shift in the hypothetical job shop. It initiates processing of all the activities that take place at the beginning of each shift. These include recalculating the release dates for all the jobs in the order pool, and determining the criticality of each work center. When these activities are complete the common logic described above is processed and the event ends.

The recalculation of the release dates is done using the appropriate logic for the ORR rule in use. This logic is described in chapter 4. The method used to determine the criticality of a work center was explained earlier.

The next event initiates processing of activities that occur when a job leaves the FGS inventory.

3.5.6 EXIT EVENT

The Exit event is scheduled when a job enters the FGS inventory. This event removes a job from FGS inventory, disposes of the entity that represents the job and records all information and performance measures related to the job.

The logic common to all events is then processed and the next event on the calendar is processed.

3.6 SIMULATION DESIGN ISSUES

The previous discussion focused on the organization and the logic of the software written to simulate the job shop. This section is concerned with the methods used to ensure that the output data produced by this software is actually representative of the hypothetical job shop and is suitable for statistical analysis. Three issues will be discussed:

- Verification of simulation code.
- Removal of initialization bias.
- Determination of sample batch size.

3.6.1 VERIFICATION

Law and Kelton (1982, pp.334-337) define verification as the demonstration that a simulation model performs as intended. The techniques used in this research to verify the performance of the simulation were:

1. Modular programming of the simulation model to allow separate debugging;
2. A structured walk through of the program;
3. Comparison of the model to known output.

3.6.1.1 MODULAR PROGRAMMING

Modular means that the code was written in sections which could operate and be tested on their own. Each module performs a simple task by itself. Modules are combined to perform complex tasks. These modules or subroutines were combined one by one. This allowed them to be debugged individually and as a group.

The simulation code is shown in Appendix B. It is apparent from this code and from the flow chart in Figure 3-1 that the program consists of a series of linked modules. Once the entire code was written, and all the modules were combined, the model was examined as written to see if the results were reasonable.

After completing the coding and debugging the simulation, 29,000 jobs for each of the 72 cells were generated. The results are shown in Table 3-2. No attempt was made to set the planning factors so that the performance of the

treatments were optimized. The range of the flow times were from 22.751 to 40.748. The largest variance was 103.59 and the smallest variance was .660. It was concluded that these results did not indicate an error in the simulation code.

The next step was to conduct a structured walk through to ensure that there were no errors in the code.

3.6.1.2 STRUCTURED WALK THROUGH

In the structured walk-through, the programmer explains each line of code to a knowledgeable individual not involved in the programming. This was done while the program was stepped through each set of experimental conditions.

SIMAN's "Interactive Debugger" allowed interruption of the processing to view the status of the experiment (Systems Modeling Corporation, 1988, p.II-11). SIMAN's tracer (Systems Modeling Corporation, 1988, p.III-145) displayed the value of the system variables over time as a series of jobs for each experimental setting were run. During each event, SIMAN was paused. Then the relevant code for that setting of variables was examined by Paul Blossom, a fellow Ph.D. student with experience programming in fortran and SIMAN. Using the code, the actions of SIMAN through the event were predicted. The current system status and the trace were then consulted to ensure that the code had behaved as predicted.

This structured walk through provided initial evidence that the code was debugged. Additional verification was obtained by comparing the simulation output to other known output.

TABLE 3-2
VERIFICATION RUNS OF SIMULATION

PERFORMANCE MEASURE IS COST/WEEK: SMOOTH PLANNING SYSTEM
TRUNCATE 3000 JOBS: BATCH SIZE 50: DUE DATE FACTOR = 9.50
PARAMETERS: MIL & CMS 1 = 5: 2 = 2. MLC 1[1] = 1.

ORR RULE	WHEN RULE	JOB DISP RULE	BATCH SIZE	VARIANCE OF COST	AVERAGE OF COST
1	1	1	50	170.880	298.504
1	1	2	50	724.490	286.073
1	1	3	50	531.880	290.500
1	2	1	50	658.050	371.322
1	2	2	50	854.760	326.892
1	2	3	50	708.850	333.437
1	3	1	50	825.580	357.914
1	3	2	50	825.910	323.248
1	3	3	50	687.030	319.638
2	1	1	50	172.590	315.038
2	1	2	50	799.330	303.001
2	1	3	50	678.970	309.253
2	2	1	50	836.930	390.709
2	2	2	50	1014.970	349.272
2	2	3	50	939.240	358.142
2	3	1	50	911.260	375.759
2	3	2	50	1056.300	345.113
2	3	3	50	871.630	346.336
3	1	1	50	694.690	320.548
3	1	2	50	1410.920	346.759
3	1	3	50	1369.250	347.409
3	2	1	50	1445.960	367.170
3	2	2	50	1416.110	363.501
3	2	3	50	1284.720	367.499
3	3	1	50	1622.170	366.041
3	3	2	50	1475.730	363.888
3	3	3	50	1480.800	363.482
4	1	1	50	570.700	302.248
4	1	2	50	1253.510	326.838
4	1	3	50	1438.400	330.461
4	2	1	50	1509.630	348.472
4	2	2	50	1507.270	343.908
4	2	3	50	1305.050	348.353
4	3	1	50	1643.020	345.514
4	3	2	50	1492.220	346.090
4	3	3	50	1561.920	343.972

[1]. At the time of this run, the MLC rule used only one parameter. It was later changed to use two parameters.

3.6.1.3 COMPARISON OF SIMULATION RESULTS TO KNOWN RESULTS

By making appropriate simplifications in the model, the code could be compared to three standards existing in the literature. These were:

- The standard M/M/1 and M/M/2 queuing models;
- The results from Ragatz (1985);
- The results from Park (1987).

3.6.1.3.1 Comparison to M/M/s Standard

The next step was to run the model first as a M/M/1 and then as a M/M/2 system using the same random number stream. To simplify the simulation, the labor constraints were removed, the number of machines were reduced, and early completions were allowed. The results were summarized in Table 3-3. For a M/M/1 system: the theoretical flow time was 10.0. The simulation mean flow time as a M/M/1 system was 9.32 with a variance of 7.61. For the M/M/2 system: the theoretical flow time was 5.26. The simulation mean flow time as a M/M/2 system was 4.94 with a variance of 6.94. It was concluded that there were no significant differences between the theoretical flow times and the simulation results for either system using a t-test.

3.6.1.3.2 Comparison to Ragatz (1985)

The labor constraint was dropped from the simulation to compare it to Ragatz (1985). Table 3-4 compares this simulation's results with some of the results obtained by Ragatz (1985). This simulation is not an exact duplicate of Ragatz model. One difference is that this simulation allows a maximum of 5

operations per job while Ragatz's simulation allowed a maximum of 8 operations per job. Both simulations operated at 87% of capacity and both simulations accumulated new arrivals in a pre-shop file until the beginning of the next week.

Two representative settings of Ragatz's simulation were used as the basis of comparison to check the validity of this model. For both settings, the dispatch rule was FCFS. For the first setting, there was no ORR rule in effect (IMM) and for the second setting the MIL ORR rule was used. The simulations were compared on the basis of their In-Shop Queue time per job and their average number of jobs on the shop floor. In addition, theoretical values for the in-shop queue time per job were calculated for both simulations when the IMM ORR was used.

The theoretical In Shop Queue Time per job for both this model and the Ragatz model were calculated as if both models were a network of M/M/1 queues (Jackson, 1957). This simulation results corresponds closely to Ragatz results for the number of jobs in the queue.

The simulation results for the amount of In-Shop Queue Time (34.48) were not statistically different from Ragatz (28.07) results using a t-test. These theoretical results were compared to the simulation results as shown in Table 3-2. The theoretical In-Shop Queue Time calculations assumed continuous arrivals. However, both simulation models batched arrivals and released them at the beginning of the week, which increased their In-Shop Queue time compared to the theoretical value. The theoretical In-Shop Queue time for the Ragatz model

TABLE 3-3
COMPARISON OF SIMULATION TO THEORETICAL RESULTS

M/M/1 AND M/M/2 SYSTEMS

FACTOR SETTINGS	M/M/1 THEORETICAL FLOW TIME	M/M/1 SIMULATION FLOW TIME MEAN/VAR	M/M/2 THEORETICAL FLOW TIME	M/M/2 SIMULATION FLOW TIME MEAN/VAR
IMM,QUE,FCFS	10.0	9.32/7.61	5.26	4.94-/6.94

λ = MEAN ARRIVAL RATE = 0.9 JOBS/HR
μ = MEAN SERVICE RATE = 1.0 JOBS/HR
IMM = IMMEDIATE RELEASE OF JOBS FROM ORDER POOL
QUE = REASSIGNMENT OF LABOR WHEN QUE IS EMPTY
JOB = REASSIGNMENT OF LABOR WHEN JOB IS COMPLETE
CLR = REASSIGNMENT OF LABOR BASED ON MACHINE CRITICALITY
FCFS = FIRST COME FIRST SERVED JOB DISPATCHING RULE

M/M/1 SYSTEM[1]

FLOW TIME = $1/(\mu - \lambda)$
where λ = 1.8 JOBS/HR
μ = 1.0 JOBS/HR

M/M/2 SYSTEM

FLOW TIME = $\{\mu(\lambda/\mu)^2/([2-1]![2\mu - \lambda]^2)\}P_0(t) + 1/\mu$
where $P_0(t) = \{(\Sigma^{s-1}_{n=0}[1/n!][\lambda/\mu]^n) + ([1/s!][\lambda/\mu)^s[s\mu/(s\mu-\lambda)])\}^{-1}$
n = NUMBER OF UNITS IN THE SYSTEM
λ = 1.8 ARRIVALS PER HOUR
μ = 1 SERVICE PER HOUR
s = 2 SERVERS
$\lambda/s\mu = 1.8/2 = .9 < 1$
SOLVING $P_0(t) = (19)^{-1}$
FLOW TIME = 5.26

[1]. All formulas are from Panico (1969).

TABLE 3-4
COMPARISON OF THIS SIMULATION TO RAGATZ (1985)

PERFORMANCE MEASURE	LABOR ASSIGNMENT	RAGATZ (1985) IMM/FCFS	RAGATZ (1985) MIL/FCFS	SIMULATION IMM/FCFS AVG/VAR	SIMULATION MIL/FCFS AVG/VAR
IN-SHOP QUEUE TIME/JOB	QUE	28.07	20.73	35.48/22.62	29.01/28.16
	THEORETICAL	21.96		26.86	
NUMBER JOBS ON SHOP FLOOR	QUE	43.38	33.19	39.14/36.51	32.08/37.70

CALCULATIONS ARE BASED ON M\M\1 TIME IN QUEUE USING JACKSON'S DECOMPOSITION THEOREM:

$$E_t = \text{TIME IN QUEUE} = \lambda/\mu(\mu-\lambda)$$

	RAGATZ	THIS MODEL
UTILIZATION	.87	.87
MEAN NUMBER OF OPERATIONS	3.39	4.0
PROCESSING TIME/OPER	.924	1.0
ARRIVALS/HOUR AT MACH	1.39	1.088
ARRIVAL (λ)	.945	.8704
SERVICE RATE (μ)	1.08	1.0
E_t FOR OPERATION	6.477	6.716
E_t FOR JOB	21.96	26.86

is 21.96 hours per job and it is 26.86 for this simulation compared to 28.07 and 35.48 for Ragatz and this model respectively. Comparing the results of this simulation and Ragatz's simulation when the MIL rule was used, the In-Shop Queue time for both models decreased. The decrease was 7.34 for Ragatz and for this simulation 6.47, corresponding very closely to Ragatz measures. There is a close correspondence between the two simulations for the Number of Jobs on the Shop Floor. The decrease in the Number of Jobs on the shop floor comparing MIL to IMM is in the same direction. These similarities suggest that the code produces results similarly to those generated by the Ragatz code.

3.6.1.3.3 Comparison to Park (1987)

This simulation was compared to Park's (1987) simulation because Park (1987) examined ORR rules in a DRC shop. To compare this simulation to Park (1987), the distributions for the arrival of jobs and processing times were adjusted. The output of the simulations are compared in Table 3-5.

The In-Shop Queue time for this simulation model was higher than that obtained by Park, however this was expected because Park used only five work centers and this shop used six work centers. Also, Park averaged 3 operations per job while this simulation averaged 4 operations per job.

The Time In Order Pool measure for Park includes the time the job waited in the planning system before entering the shop. If this average time (4.01) was subtracted from the time in pool for the MIL job selection, the difference of 6.91 was very close to the results of this simulation which was 6.552. These results

TABLE 3-5
COMPARISON OF PARK (1987) TO THIS MODEL

PERFORMANCE MEASURE	LABOR ASSIGNMENT	PARK (1987) IMM/MODD	PARK (1987) MIL/MODD	SIMULATION IMM/MODD AVG/VAR	SIMULATION MIL/MODD AVG/VAR
IN-SHOP QUEUE TIME/JOB	JOB	8.09	7.01	19.915/0.85	17.772/7.92
TIME IN ORDER POOL/JOB	JOB	4.01*	10.92	0.0	6.552/0.36

*PARK INCLUDED TIME THE JOB SPENT WAITING IN THE PLANNING SYSTEM AS TIME IN THE ORDER POOL.

indicate that this simulation and Park's (1987) simulation produce comparable output.

Summary: The modular programming, structured walk through and comparison of this simulation code to known standards strongly support the belief that the simulation code does not contain errors in programming. Next, the steps taken to ensure that the sample data meets the necessary statistical assumptions are discussed.

3.6.2 INITIALIZATION BIAS

The hypotheses in this experiment are concerned with the performance of a DRC job shop in steady state. However, when the simulation starts processing, the shop is empty. This creates a negative bias on the measurements. Kleijnen (1987,p.110) recommends the use Schruben, Singh and Tierney's (1983) technique to eliminate initialization bias. This technique assumes that the last half of a long time series was stationary and that the purpose was to identify the point at which the first half becomes stationary. First the variance and the average of the second half of the time series are calculated. Second, an arbitrary batch size of 50 was chosen to calculate the running average of the first half of the data. This running average was compared with the average of the second half of the time series using a t-test with a 95% confidence interval. If the test value was not significant, then the null hypothesis that there was no difference between means of the first part of the time series and the second part of the time

series was accepted. If the null hypothesis was rejected, the first batch of jobs used in the moving average calculation was dropped and the running average was recalculated.

Kleijnen (1987) recommends that prior to starting the calculations that those jobs which are obviously not stationary be dropped. After inspection of the data, the first 100 jobs were dropped before beginning the calculations. If the initialization bias ended in less than 500 jobs, the first 500 jobs were truncated. To ensure that an adequate sample size was available, 32,000 jobs were run for each treatment in the experiment. The results for each treatment are shown in Table 3-1, where the largest number of jobs that needed to be eliminated was 2,252. For most cells, the initialization bias was removed within 100 jobs. The fortran code for this routine was in Appendix C.

3.6.3 BATCH SIZE

The batch size effects the amount of autocorrelation and the normality of the sample. Each of these is addressed separately.

3.6.3.1 AUTOCORRELATION

After the bias of the transient phase was removed, the batch size needed to ensure independence was determined. The independence of the sample was threatened by autocorrelation between the batch means. Independence was obtained by increasing the batch size until the batches were uncorrelated (Law and Carson, 1979).

A procedure, recommended by Paul Rubin, was developed which uses the Box-Pierce Q statistic (Makridakis, Wheelwright and McGee, 1983, pp.355-388) to determine when autocorrelation between batch means was statistically insignificant for a lag of 20. As Kleijnen (1987,p.67) recommended a minimum of 100 batches were used to calculate the autocorrelation for a given batch size.

The autocorrelation for lag p was calculated as:

$$\rho_{p,m} = \frac{\sum_{i=1}^{k-1} (Y_{i,m} - \overline{Y_k})(Y_{i+p,m} - \overline{Y_k})}{\sum_{i=1}^{k} (Y_{i,m} - \overline{Y_k})^2}$$

where: $\rho_{p,m}$ was the autocorrelation of batch means with lag p and size m;

k was the total number of batch means;

$Y_{i,m}$ was the batch mean of i = 1 ... k;

Y_k was the mean of k batch means.

For these runs, k was always 100 batch means and the lags (p) were calculated from 1 to 20.

The Q statistic was calculated as:

$$Q = N\sum_{p=1}^{20} r_{p,m}^2$$

where: $r^2_{p,m}$ was the calculated autocorrelation of lag p = 1, ..., 20 for batch size m;

N was the total number of observations of batch means of size m.

The number of degrees of freedom for Q was $p_{max} = 20$.

The test was performed with $\alpha = 0.05$, and $\xi^2_{(.950,20)} = 31.41$. The null hypothesis was that the batch means are autocorrelated. A two sided test was taken until H_o was rejected. If H_o was accepted, the batch size was doubled and the Q statistic was recalculated for another 100 batches. The test compares the Box-Pierce Q statistic (Makradakis, Wheelwright and McGee, 1983,p.390) to the critical ξ^2 value. The fortran code for these subroutines was in Appendix C.

3.6.3.2 NORMALITY

The batch size also influences the normality of the sample. Normality was tested using a technique developed by Filliben (1975) as recommended by Paul Rubin. The technique works by testing the correlation of the batch means with their normalized scores and increasing the batch size until the correlation was high enough to indicate normality (Ryan, Joiner and Ryan, 1982, p. 48). The null hypothesis was that the means are normal. First, the data was sorted into

ascending order, and the order statistic medians are calculated. Then Phi inverse of the order statistic was calculated using an approximation recommended by Ryan, Joiner and Ryan (1982). The test statistic was the correlation of the original data and the normalized scores. This correlation was compared to the critical correlation tables developed by Filliben. If the correlation exceeds the critical value, the null hypothesis that the scores are normal was accepted. If the null hypothesis was rejected, the batch size was increased and the calculations were repeated.

This procedure was tested using the sample problem in Ryan, Joiner, Ryan (1982, p.47) where it produced comparable results. It was also tested with another set of data which was analyzed using the NSCORES command in Minitab (Ryan, Joiner, Ryan, 1982). Its results agreed with Minitab results. Finally, the results were checked by producing a histogram of the output. The histogram agreed that the output could be normal. The fortran code for this subroutine was in Appendix C.

The pilot runs indicate that the batch size needed for normality was significantly larger than the batch size needed for independence. The results of these runs are shown in Table 3-1.

3.6.6 SUMMARY

About 40 weeks data (2,160 jobs) was truncated to eliminate initialization bias in this experiment. The batch size for all treatments was set at 90 weeks

(or about 4860 jobs) to eliminate autocorrelation and obtain normality. The sample size was set at 100 to ensure that there would be adequate statistical power to distinguish treatment effects.

3.7 PERFORMANCE MEASURES

Cost was selected as the main performance measure. There was support in the literature for the use of total cost as the performance measure. Blackstone, Phillips and Hogg (1982) argued that the only relevant measure of performance when evaluating job dispatching rules was cost effectiveness. Jones (1973) points out that cost measures allow for consideration of economic tradeoffs. Jones uses a cost structure with costs for idle machines, WIP, late jobs and long lead times. Jones found that the choice of the cost structure favors different job dispatching rules. Maggard, Hogg and Phillips (1980) used cost as the performance measure in a DRC shop. They calculated the cost based on the cost of machine centers time, the cost of labor, the cost of the mean service rate and the cost of jobs waiting. Ragatz and Mabert (1988) used a cost structure composed of late delivery and inventory costs with a ratio of 20 to 1. Bobrowski (1989) used a cost structure composed of late delivery costs and inventory costs set at a ratio of 30 to 1. Bobrowski and Park (1989) used a cost structure composed of late delivery, inventory and labor transfer costs with a ratio of 10/1/1.

Total cost was used as a performance measure by Ragatz (1985) and Park (1987). Cost measures were composed of lateness costs, inventory costs and labor transfer costs which incorporate the more traditional job shop performance measures. Inventory costs incorporate WIP and FGS inventory, earliness measures, and flow time measures. Lateness costs incorporate tardiness and variance of tardiness measures. Labor transfer costs incorporate the number of labor transfers and costs of delays in labor transfers.

The cost structure used in this experiment was 20/1/10 for the late delivery costs, inventory holding costs and labor transfer costs respectively. This means that the tardy cost per hour was 20 times as large as the charge for each hour of inventory in stock. It means that the charge to transfer a worker was ten times as large as the charge for keeping one hour of inventory.

Inventory holding costs are computed by multiplying the number of inventory hours by the cost per hour ($0.10/hour). Inventory hours are computed by summing all the hours of work completed and still in the shop whether in the form of WIP or FGS.

Late delivery costs are computed by summing the number of hours of work late by the number of hours the work was late in each period and multiplying this by the cost of late work ($1.00/hour). Labor transfer costs are assessed as a charge per transfer ($0.10).

The total cost of processing jobs was summed for each week (40 simulated hours). The average cost per job was calculated and stored. When 90 weeks

of data was collected the average cost per week was calculated and stored as one sample.

3.7.1 SECONDARY PERFORMANCE MEASURES

To aid in interpretation, ten other performance measures were collected. These are:

- mean tardiness per tardy job;
- the 90th percentile of tardiness;
- mean flow time per job;
- the 90th percentile of flow time;
- mean time in shop queue per job;
- mean time in the order pool per job;
- the mean time in the system per job;
- the mean percent of jobs tardy per week;
- the work-in-process inventory; and,
- the finished goods inventory.

The 90th percentile of tardiness indicates the consistency of meeting due dates. The mean flow time is a measure of the time the job actually spends on the shop floor. It does not include time the job spends in the order pool waiting to be released or time in finished goods inventory. The 90th percentile of flow time was an estimate of lead time stability.

The in shop queue time per job measures the amount of time the job spends waiting on the shop floor. The pool time per job was the amount of time that the job spends in the order pool before being released to the floor. The mean time in the system per job is the sum of the job's flow time and its time in the order pool.

3.8 DATA COLLECTION

The simulation was run on the microcomputer using SIMAN (Pegden, 1988). It was not possible to obtain the desired 100 samples for each treatment in one long run. To ensure that common random numbers were maintained for 3 of the 4 random number streams, five separate runs were made for each treatment. A different set of seeds for the three common random number streams were used for each of the 5 runs. The fourth independent stream used a different seed for each run for each treatment.

At the beginning of each run 40 weeks of data or about 2160 jobs were truncated to eliminate the initialization bias. After all the data was collected the means and variance for each cell were calculated, a scatter diagram of the cost was drawn for each cell. The outliers in the data were inspected to determine if they were due to experimental error. However, no outliers were identified as experimental error.

3.9 SUMMARY

This research used a simulation of a hypothetical job shop to examine the effect on shop performance of the experimental variables. The shop operating assumptions, the simulation code, and the processing of the code were described. Finally, the methods used to reduce variance and improve the statistical power were explained. Also, the methods used to obtain normal, independent data were explained.

The main performance measure was total cost. Total cost was used because it was a composite of several independent performance measures (tardiness, inventory and labor transfers).

Chapter 4 will examine the experimental design and explain the experimental variables. Chapter 4 will also present the results of the initial pilot runs used to establish the parameters for the ORR rules.

CHAPTER 4

4.1 INTRODUCTION TO EXPERIMENTAL DESIGN

This chapter summarizes the experimental issues that are addressed by this research. It also discusses the experimental variables, the hypotheses, the experimental design, and the pilot runs conducted to set the parameters for the experiment.

4.2 EXPERIMENTAL VARIABLES

A full factorial design with four experimental factors was used:

1. The type of planning system: two (2) levels,
2. Order Review/Release Rules: four (4) levels,
3. "When" Labor Assignment Rules: three (3) levels,
4. Job Dispatching Rules: three (3) levels.

Each of these experimental variables will now be discussed.

4.2.1 THE TYPE OF PLANNING SYSTEM

Two different types of planning system were represented:

- the UNCONTROLLED planning system did not level the work load before sending it to the shop. This planning system was simulated with a poisson distribution (mean of 54) to determine the number of jobs sent from planning to the shop each week. The variance of

this planning system represented a measure of the fluctuation from week to week in the number of jobs sent from planning to the shop. As noted in chapter 2 this is the most common type of planning system used in the past research.

- the SMOOTH planning system sent a level load of work from planning to the shop. This planning system was simulated using a uniform distribution with a minimum of 53 jobs and a maximum of 55 jobs a week. The simple capacity planning procedures described by Schmitt, Berry and Vollmann (1985) or the more complex smoothing procedure described by Kingsman, Tatsiopoulos and Hendry (1989) could achieve this smooth schedule. In this system both the peaks (high loads) and valleys (low loads) were leveled or smoothed.

These two planning systems were picked to represent extremes that practitioners generally encounter. The UNCONTROLLED planning system creates a shop load with many "peaks" and "valleys", while the SMOOTH system creates a flat load. Melnyk, Ragatz and Fredendall (1990) tested the SMOOTH planning system in a machine constrained system and found that it made a significant improvement in shop performance. Both planning systems released jobs into the shop only at the beginning of the week. A week was defined as starting every 40 hours.

4.2.2 ORDER REVIEW/RELEASE RULES

This research examined the trigger mechanisms and the job selection mechanisms of the ORR rule. Parameters were used in three of the ORR rules. The method used to select these parameters is explained separately in this chapter. This discussion is focused on why these particular ORR rules were selected and how these rules operated.

- Immediate Release (IMM): This ORR rule meant that all jobs are released immediately to the shop floor when they arrive from the planning system. Consequently, there is no trigger mechanism or job selection mechanism. This is a naive rule and is included to allow comparison with the other rules. Also, some prior research (e.g. Baker, 1984; Kanet, 1988) found that immediate release of jobs to the floor could achieve better performance than achieved with some ORR rules.
- Modified Infinite Loading (MIL): MIL was used in this research because it performed well in prior research (Ragatz and Mabert, 1988). MIL used only POOL and SHOP information and did not use LABOR information. It calculated the release date of each job in the order pool. The release date is based on the job's due date, the job's processing time and the number of jobs currently in the queues on the job's route.

Ragatz and Mabert (1988) and Bobrowski and Park (-1989)[1] found that MIL was the best performing job selection mechanism they tested. As discussed in chapter 2, these researchers used slightly different versions of the MIL rule. This research combined portions of Bobrowski and Park's (1989) and Ragatz and Mabert's (1988) versions of MIL. Mathematically, the release date (RD) in this research was calculated as:

$$RD_j = DD_j - X_6*PROC_j - X_7*Q_j$$

where: DD_j was the due date of job j; X_6 and X_7 were planning factors or parameters; Q_j was the number of jobs currently in the queues at machines on the route of job j; and, PROC was the total processing time for job j.

Unlike the Ragatz and Mabert (1988) MIL rule, which used the number of operations in the job, this rule used the job's processing time to calculate the release date. In contrast to the Bobrowski and Park (1989) MIL rule, which measured congestion in the queues by totaling the processing time of the jobs in the queues, this rule counted the number of jobs in the queues to calculate the release date.

---

[1]. Bobrowski and Park refer to their version of MIL as BIL. It is almost identical to the MIL rule used by Ragatz and Mabert (1988).

The MIL rule included the trigger mechanism in the calculation of the release date. The release date moved forward or backwards based on the congestion on the shop floor.

At the beginning of each shift, the MIL rule calculated the release date for each job in the order pool. If the release date was within the next shift (next 8 hours), then the job was scheduled to enter the shop at the release date. If the release date occurred after the end of the shift, the job would remain in the order pool and a new release date would be calculated at the beginning of the next shift.

- Critical Machine Selection (CMS): The CMS rule was created because it calculates the release date for each job exactly the same as MIL. However, the independent trigger mechanism used with the CMS rule used LABOR information. By using this trigger mechanism, the CMS rule used POOL, SHOP and LABOR information in determining which job to release to the floor.

  There were four characteristics of the CMS rule not shared by the MIL rule. They were:

  1. Jobs were released only when an operator was idle and there were no work centers to which the operator could be transferred.

2. All jobs in the order pool scheduled to be released prior to the end of the shift were eligible for release if there was idle labor.

3. The CMS rule selected the highest priority job whose first operation was the most critical work center with an idle machine. If more than one job was eligible to be assigned to this work center, the job with the earliest release date was selected.

4. Jobs were not automatically released into the shop from the pool even if their release date was current unless they were selected for release as described in point number 3 above. The release date for these jobs not released was recalculated at the beginning of the next shift.

- Modified Load Conversion (MLC): The MLC rule was selected because it used the same trigger mechanism as the CMS rule. Work was released only if an operator was idle and could not be assigned elsewhere. Consequently MLC used POOL, SHOP and LABOR information. However, it used a different job selection mechanism than the CMS rule used.

  Also, just as the CMS rule did, the MLC rule examined all jobs in the order pool eligible for release and selected the job whose first work center was the most critical work center with an

idle machine. If more than one eligible job was assigned to this work center, the job with the earliest release date was selected and released to the shop.

The job selection mechanism in the MLC rule used a finite loading heuristic. It estimated the future loads at each work center over time to calculate the release date of the job. This heuristic was based on the load conversion techniques developed by Bechte (1982, 1988a, 1988b). This load conversion heuristic attempted to obtain the benefits of finite scheduling without performing all the calculations.

The MLC rule estimated the release date of a job by "discounting" future work loads destined for each work center. This discounting resulted in a "net present value" for the future work at each work center. This net present value was then used to calculate a flow allowance for each job. The release date was then calculated by subtracting the flow allowance and the job's processing time from the job's due date.

The specific calculations to obtain the MLC release date are as follows:

1. The discount factor, DF_I, for the next five operations in sequence is calculated using the discount rate, X_{10}[1]:

$$DF(I) = (X_{10})^{I-1}$$

where I equals 1,2,3,4,5 to represent the operation's position in the job's routing. Because there were at the most five operations in a job's routing, only five discount factors need to be calculated.

2. Each job in the order pool, on a machine or in a queue was examined and the discounted load that its operations placed on each machine were calculated. This was done by multiplying the processing requirements of each operation in the job by the discount factor for that operation's order in the routing for operations not completed. For example,

$$DF(\text{operation 1}) = (X_{10})^{1-1} = 1$$

$$DF(\text{operation 2}) = (X_{10})^{2-1} = X_{10}$$

$$DF(\text{operation 3}) = (X_{10})^{3-1} = (X_{10})^2$$

Note that if a job completed the first two operations, then the discount factor for the third operation would be 1. If job "AA" had three operations (1,2 and 3) that had to be processed in numerical sequence, then the discount factor for operation 1 is 1.

---

[1]. The discount rate, X_{10}, is constrained between 0 and 1. At 0, the work is evaluated as being 0. At 1, the work is not discounted.

The discount factor for operation 2 is X_{10}. The discount factor for operation 3 is $(X_{10})^2$.

3. The discounted load (DLOAD) of each work center was calculated as the sum of all the discounted processing time of the jobs. The discounted load at work center was calculated as below:

$$DLOAD(WC) = \sum_{jobs} DF(I)*PROCESSING_{jobs}$$

where PROCESSING was the processing time for the appropriate operation of the job.

4. Since the congestion at each work center was the discounted load for the work center, the congestion on a job's route was the sum of the discounted loads at each work center in the job's routing. The sum of the discounted loads for the remaining operations of the job (2 through 5) were divided by the congestion parameter X_9 as shown below:

$$CONGESTION(JOB) = DLOAD(WC_1) + \sum_{i=2..5} DLOAD(WC_i)/X_9$$

5. The release date of each job in the order pool was calculated by subtracting the processing time of the job, and the congestion the job would experience from the due date of the job. The formula is:

$$RD = DD - PROCESSING - CONGESTION$$

Note that the greatest weight was placed on the congestion at the first machine on the job's routing since that estimate of congestion was the most accurate.

To illustrate how this rule works, a small example is given in Appendix D.

4.2.3 "WHEN" LABOR ASSIGNMENT MECHANISMS

The "when" labor assignment rule determined when an operator was eligible to be transferred to a new machine. Note that an operator could be eligible for transfer, but not be transferred because the operator was already at the best machine according to the logic of the "where" rule. The three "when" labor assignment rules tested were:

- QUE: The QUE "when" labor assignment rule allowed an operator to transfer to another work center only when there were no jobs in the queue of the operator's current work center. This rule was selected because it has often been the best rule in prior research (Treleven, 1989).
- JOB: The JOB "when" labor assignment rule allowed an operator to transfer to a new work center whenever the operator completed a job. The JOB rule used SHOP and LABOR information and has also been frequently used in prior DRC research (Treleven, 1989). It is at the opposite end of the spectrum from the QUE rule.

- Critical Labor Release (CLR): The CLR "when" labor assignment rule was developed for this research to incorporate POOL information as well as SHOP and LABOR information into "when" labor assignment decision. CLR was a compound "when" rule because it used either the JOB or QUE rules for the decision depending on the criticality of the work center the operator was currently assigned to. If a work center was critical then the "when" labor assignment rule for the work center was QUE. If a work center was not critical then the "when" labor assignment rule was JOB.

  The rationale for the CLR rule was that an operator should remain at the critical work centers to prevent bottlenecks from developing. And further, that an operator should leave non critical work centers to go to critical work centers as needed. The concept of criticality and its use of POOL information was explained in chapter three.

These "when" labor assignment rules provide a representative range of different methods of making the "when" labor assignment decision. The JOB and QUE methods are at opposite ends of the spectrum, while the CLR rule provides a mid-range method.

4.2.4 JOB DISPATCHING RULES

Three job dispatching rules were considered in this research. Each rule is discussed below:

- First Come First Served (FCFS): The FCFS job dispatching rule processes jobs in a queue in the order in which the job arrived in the queue. FCFS was used as an experimental variable because Wight (1970), Plossl and Wight (1973), Bechte (1982, 1988a, 1988b) and Sandman and Hayes (1982) suggested that this simple rule would be effective if a sophisticated ORR mechanism was used. This rule also served as a basis of comparison for the other job dispatching rules. In comparison to the other job dispatching rules, this rule used the least information.
- Modified Operation Due Date (MODD): MODD was used as an experimental variable because Bobrowski and Park (1989), Baker and Kanet (1983) and Baker (1984) found that the rule performed very well. The MODD for each job j was calculated as:

$$MODD_j = MAX\{TNOW + OPT_j, ODD_j\}$$

where: TNOW was the time now; OPT_j was the time to process the next operation of job j; ODD_j was the operation due date for the next operation of job j; and MAX{ } selected the maximum of the choices in the brackets.

The operator then processes the job with the smallest $MODD_j$. The MODD dispatching rule uses the SOT job dispatching rule if the operation is late. If the operation is early or on time, it uses the operation due date (ODD) priority rule. This tends to speed up late jobs and to slow down early jobs. This rule uses more information than the FCFS rule, but all the information used is SHOP information. The method used to establish operation due dates was explained in chapter 3.

- Critical Machine Due Date (CMDD): CMDD is a compound dispatching rule similar to MODD. It also uses either the SOT or the ODD job dispatching rules, but it differs from MODD in two ways:

  1. By using the concept of criticality defined in chapter 3, the CMDD rule utilized the POOL information. Depending on the criticality of the individual work center, the CMDD rule determined whether the SOT or ODD job dispatching rule should be used. In contrast, the MODD rule determined whether SOT or ODD should be used based on the characteristics of an individual job.

  2. The CMDD rule uses the same dispatching rule for every job in the queue of a work center. If a work center is critical, <u>all</u> jobs in its queue are prioritized with the SOT dispatching rule. If a work center is not critical, <u>all</u> jobs in its queue are prioritized with

the ODD rule. This is in contrast to the MODD rule which applies the SOT and ODD rules to different jobs in the same queue.

The goal of the CMDD dispatching rule is to move jobs through the bottleneck work centers quickly and keep them on schedule in the other work centers. Each work center uses the same job dispatching rule for an entire shift. This job dispatching rule uses both SHOP and POOL information.

The a priori hypotheses which these experimental variables were selected to investigate will now be discussed.

4.3 A PRIORI HYPOTHESES

The following hypotheses were proposed prior to the start of experimentation. The hypotheses were formulated as non orthogonal linear contrasts with an individual comparison confidence level of 0.05.

These hypotheses were proposed to examine the general research questions discussed in section 1.3. The focus of these hypotheses was 1) to determine the value of using additional information to make SFC decisions; 2) to determine the benefit of using planning systems to level the work load sent from the planning system into the shop; and 3) to investigate whether there was any benefit to finite loading of each work center using the ORR rule.

The expectations of this research were that using additional sources of information to make SFC decisions would improve performance. It was also

expected that increased variance in the workload received by the shop would result in performance deterioration. Finally, it was expected that using a finite loading ORR rule would improve performance when supported by a job dispatching rule which enforced on time processing of the jobs.

The statistical analysis will be discussed fully in chapter 5, but it needs to be mentioned here that the analysis consists of three stages. First an ANOVA model will be fitted to determine if the factors are significant and to establish whether there are significant higher order interactions. If there are no significant higher order interactions, the critical value of the a priori hypotheses will be calculated. However if the higher order interactions are significant, the weights for these particular linear contrasts will not be meaningful (Kirk, 1982, p.371). Bonferroni multiple comparisons will then be used to conduct a post hoc analysis of the data (Kirk, 1982, p.106).

The i and j subscripts in the hypotheses below refer to the row and column number respectively in Table 4-1.

Hypothesis 1: The first hypothesis was that treatments which used the SMOOTH planning system would always perform better than treatments which used the UNCONTROLLED planning system. The proposed contrast was:

H_o: $\psi_1 \leq 0$

H_A: $\psi_1 > 0$

TEST STATISTIC: $\psi_1 = \{\Sigma^9_{i=1} \Sigma^8_{j=5} \mu_{ij}\}/36 - \{\Sigma^9_{i=1} \Sigma^4_{j=1} \mu_{ij}\}/36$

TABLE 4-1
CELLS IN MAIN EXPERIMENT: LABELED BY ROW, COLUMN

Labor Assign.	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM								ROW
		SMOOTH				UNCONTROLLED				
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC	
QUE	FCFS	11	12	13	14	15	16	17	18	1
	MODD	21	22	23	24	25	26	27	28	2
	CMDD	31	32	33	34	35	36	37	38	3
JOB	FCFS	41	42	43	44	45	46	47	48	4
	MODD	51	52	53	54	55	56	57	58	5
	CMDD	61	62	63	64	65	66	67	68	6
CLR	FCFS	71	72	73	74	75	76	77	78	7
	MODD	81	82	83	84	85	86	87	88	8
	CMDD	91	92	93	94	95	96	97	98	9
COLUMN NUMBER		1	2	3	4	5	6	7	8	

INDIVIDUAL CELLS ARE REFERENCED IN THE TEXT BY ROW AND COLUMN NUMBER. CELL CONTENTS ARE ITS ROW AND COLUMN NUMBER.

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 2: The use of labor information in ORR rules results in improved shop performance compared to the other ORR rules being examined if the planning system is SMOOTH. The proposed contrast is:

H_o: $\psi_2 \geq 0$

H_A: $\psi_2 < 0$

TEST STATISTIC: $\psi_2 = \{\Sigma^9_{i=1}\Sigma^3_{j=4}\mu_{ij}\}/18 - \{\Sigma^9_{i=1}\mu_{i2}\}/9$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 3: The use of labor information in the ORR rule results in improved shop performance when the planning system is UNCONTROLLED. The proposed contrast is:

H_o: $\psi_3 \geq 0$

H_A: $\psi_3 < 0$

TEST STATISTIC: $\psi_3 = \{\Sigma^9_{i=1}\Sigma^8_{j=7}\mu_{ij}\}/18 - \{\Sigma^9_{i=1}\mu_{i6}\}/9$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 4: Using the SMOOTH planning system with an ORR rule which loads the shop to a finite limit (the time phasing in MLC) improves performance compared to the CMS ORR rule. The proposed contrast is:

H_o: $\psi_4 \geq 0$

H_A: $\psi_4 > 0$

TEST STATISTIC: $\psi_4 = \{\Sigma^9_{i=1}\mu_{i3}\}/9 - \{\Sigma^9_{i=1}\mu_{i4}\}/9$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 5: Using the UNCONTROLLED planning system with an ORR rule which loads the shop to a finite limit (time phasing in MLC) improves performance compared to the CMS ORR rule. The proposed contrast is:

H_o: $\psi_5 \geq 0$

H_A: $\psi_5 > 0$

TEST STATISTIC: $\psi_5 = \{\Sigma^9_{i=1}\mu_{i7}\}/9 - \{\Sigma^9_{i=1}\mu_{i8}\}/9$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 6: The use of ORR mechanisms in SFC never results in worse shop performance than immediately releasing jobs to the shop floor from the planning system. The proposed contrast is:

H_o: $\psi_6 \leq 0$

H_A: $\psi_6 > 0$

TEST STATISTIC: $\psi_6 = \{\Sigma^9_{i=1}\mu_{i1} + \Sigma^9_{i=1}\mu_{i5}\}/18 - \{\Sigma^9_{i=1}\Sigma^4_{j=2}\mu_{ij} + \Sigma^9_{i=1}\Sigma^8_{j=6}\mu_{ij}\}/54$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 7: Using POOL information in the job dispatching rules improves shop performance in shops using the SMOOTH planning system. The proposed contrast is:

H_o: $\psi_7 \geq 0$

H_A: $\psi_7 < 0$

TEST STATISTIC: $\psi_7 = \{\Sigma^4_{j=1}\mu_{3j}+\Sigma^4_{j=1}\mu_{6j}+\Sigma^4_{j=1}\mu_{9j}\}/12 - \{\Sigma^9_{i=1;i\neq 3,6,9}\Sigma^4_{j=1}\mu_{ij}\}/24$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 8: Using POOL information in the job dispatching rules improves shop performance in shops using the UNCONTROLLED planning system. The proposed contrast is:

H_o: $\psi_8 \geq 0$

H_A: $\psi_8 < 0$

TEST STATISTIC: $\psi_8 = [\Sigma^8_{j=5}\mu_{3j}+\Sigma^8_{j=5}\mu_{6j}+\Sigma^8_{j=5}\mu_{9j}]/12 - \{\Sigma_{i\neq 3,6,9}\Sigma^8_{j=5}\mu_{ij}\}/24$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 9: Using POOL information in the "when" labor assignment rules improves shop performance in shops using the SMOOTH planning system. The proposed contrast is:

H_o: $\psi_9 \geq 0$

H_A: $\psi_9 < 0$

TEST STATISTIC: $\psi_9 = [\Sigma^9_{i=7} \Sigma^4_{j=1} \mu_{ij}]/12 - [\Sigma^6_{i=1} \Sigma^4_{j=1} \mu_{ij}]/24$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

Hypothesis 10: Using POOL information in the "when" labor assignment rules improves shop performance in shops using the UNCONTROLLED planning system. The proposed contrast is:

H_o: $\psi_{10} \geq 0$

H_A: $\psi_{10} < 0$

TEST STATISTIC: $\psi_{10} = [\Sigma^9_{i=7} \Sigma^8_{j=5} \mu_{ij}]/12 - [\Sigma^6_{i=1} \Sigma^8_{j=5} \mu_{ij}]/24$

where μ_{ij} = treatment i refers to the rows and j to the columns in Table 4-1.

4.4 EXPERIMENTAL DESIGN

A full factorial design was used to avoid the presence of confounding variables if the higher order interactions were significant. This resulted in a total of seventy-two treatments in this experiment as shown in Table 4-1.

All treatments in columns 2,3,4,6,7 and 8 included the estimation of planning factors or parameters in their ORR rule. This meant that 54 cells required estimates of the planning factors and that 18 cells had no planning

factors to estimate. All the planning factors were determined by using the Complex algorithm as explained by Kuester and Mize (1973, p.368). This method is explained in the next section.

4.5 PILOT RUNS

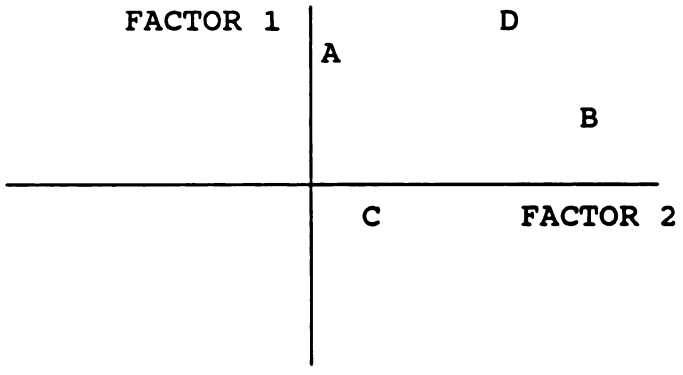
The pilot runs were made to establish the proper settings for the parameters used in the MIL, CMS and MLC ORR rules. All the treatments in columns 2,3,4,6,7 and 8 of Table 4-1, required estimation of parameter values. To ensure that the results were not biased, the values of these parameters were selected using a response surface methodology technique called Complex.

4.5.1 COMPLEX ALGORITHM

Figure 4-1 illustrates how the Complex algorithm as developed by Kuester and Mize (1973, p.367) works. Because these ORR rules require only two planning factors each, Complex conducted the search in two dimensions.

Complex first scattered an initial set of planning factors through the feasible region as illustrated in Figure 4-1 where A, B, and C each represent a set of values for the planning factors selected at random. The simulation was run until 1000 jobs were completed at each of the points A, B and C. The cost performance measure at all three points were collected and compared with the highest cost point being discarded.

FIGURE 4-1
COMPLEX ALGORITHM



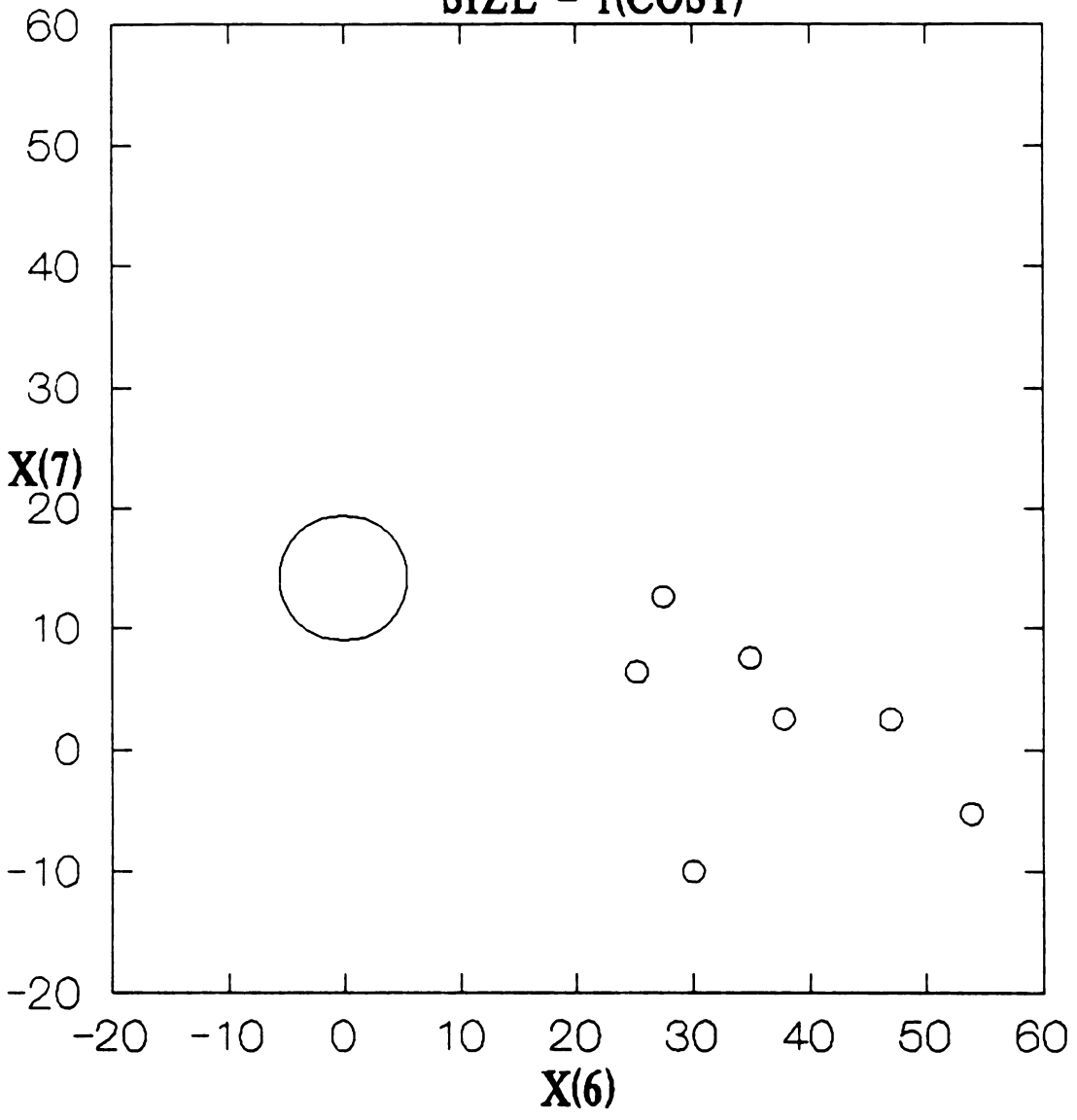
After discarding the highest cost point, the algorithm selected a new point in the feasible region to evaluate by moving a random distance through the centroid between the two low cost points. In Figure 4-1, the discarded point was C and the new point was D. These three points (A, B and D) were then evaluated. One of the three points was discarded and a new point was selected for testing. This process of discarding the highest point and selecting a new point continued until the algorithm failed to find any further improvement in cost. The final three points formed a triangle which contained the lowest cost setting for the parameters.

It was difficult using the algorithm to map the response surface for the parameters. The objective function was found using a simulation and was not calculated from a function. After much experimentation with the algorithm, the mean of each sample from the simulation was treated as a point estimate, despite its variance. It was impractical because of the large amount of computer time

FIGURE 4-2

CELL 2111: COST STRUCTURE 20/1/1

SIZE = f(COST)



MIL: RD = DD - X(6)*PROC - X(7)*CONG

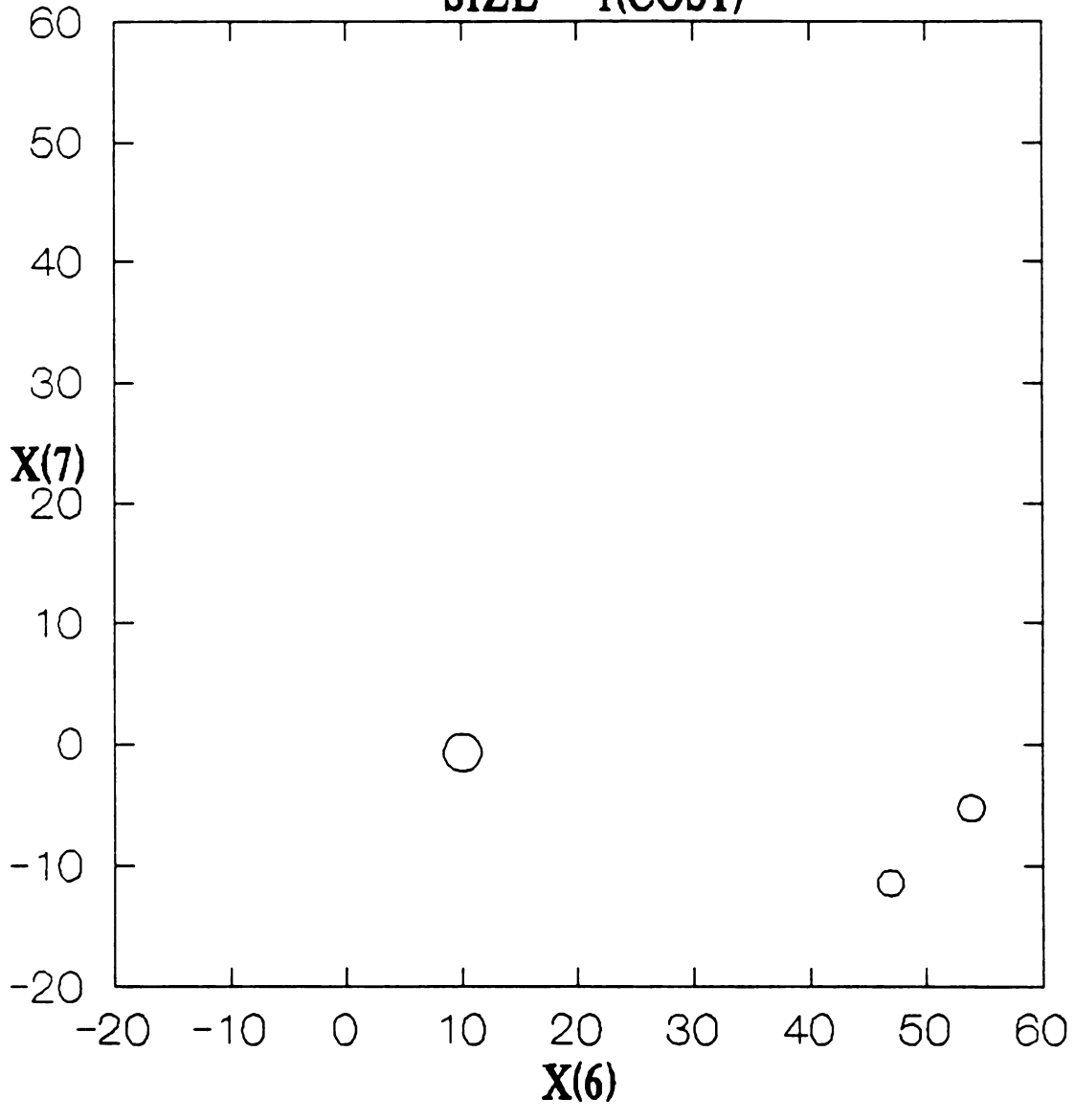
LEGEND FOR CELL LABEL

DIGIT 1=ORR, 2=MIL, 3=CMS, 4=MLC
DIGIT 2=WHEN: 1=QUE, 2=JOB, 3=CLR
DIGIT 3=DISPATCH: 1=FCFS, 2=MODD, 3=CMDD
DIGIT 4=PLANNING:1=SMOOTH,2=UNCONT

FIGURE 4-3

CELL 2112: COST STRUCTURE 20/1/1

SIZE = f(COST)



MIL: RD = DD - X(6)*PROC - X(7)*CONG

LEGEND FOR CELL LABEL
DIGIT 1=ORR, 2=MIL, 3=CMS, 4=MLC
DIGIT 2=WHEN: 1=QUE, 2=JOB, 3=CLR
DIGIT 3=DISPATCH: 1=FCFS, 2=MODD, 3=CMDD
DIGIT 4=PLANNING:1=SMOOTH,2=UNCONT

involved in gathering the samples to obtain enough sample to reduce the confidence intervals to a size where they did not overlap. However, by using multiple starting points for the complex algorithm, it was determined that these sample means provided a stationary map of the surface.

The data points checked by Complex were graphed to select the parameter values. After using alternative types of graphs, a "Bubble Graph" was constructed using Sygraph version 1.1 (Wilkinson, 1988) to represent the points. In this type of graph, the size of the bubble represents the cost for the combination of parameters represented on the axis. Larger bubbles indicate the higher cost treatments.

A minimum of two runs of the Complex algorithm was made for each treatment. Each run used different starting points and proceeded until there was no further reductions in cost achieved by selecting new points to test. This was done to help ensure that a global optimum was located. When there was still doubt about the best set of parameters, 20 samples of size 100 were run at each possible sets of parameters. Finally to ensure that the parameters selected with the algorithm were close to optimum, two additional runs were made to collect 100 samples each of size 100 weeks. The first run used the parameter settings identified by the algorithm. The second run used a different set of parameters that performed well. In all cases the planning factor values initially selected were adequate.

4.5.2 PARAMETER SELECTION

The selection of the parameters for the MIL, CMS and MLC ORR rules is discussed separately for each of these rules.

4.5.2.1 MIL PARAMETERS

Figures 4-2 and 4-3 were representative of the cost of using different combinations of parameters in the MIL rule. These graphs show that the response of the cost to changes in the parameter values was flat for a large range. For all the graphs the selection of a negative X(6) or a X(6) close to zero results in large costs. But with the exception of several points any combination of X(6) and X(7) where X(6) was 10 or greater gives little difference in cost. All X(6) and X(7) parameter values were set to 10 and -.7 respectively because this seemed to be a low cost point for all MIL cells.

4.5.2.2 CMD PARAMETERS

Figures 4-4 to 4-5 were examples of the bubble graphs used to select the parameters for the CMS rule. Again the response of the cost to changes in the parameter values was flat for a large range. All the X(6) and X(7) parameter values were set to 10 and -.7 respectively because this seemed to be a common low cost point for all of these cells.

4.5.2.3 MLC PARAMETERS

Figures 4-6 to 4-7 were used to select the planning factors for the MLC rule. The cost differences between the selection of planning factors was not as apparent as in the MIL and CMS factors. These factors were selected to corre-

spond to flat points. In many cases the selection was arbitrary because there were so many possible points of equal value. Points were selected by testing the identified low point against other points using larger runs and using the point which was the lowest cost for the main experiments.

All the parameters selected for use in the main experiment are shown in Table 4-2.

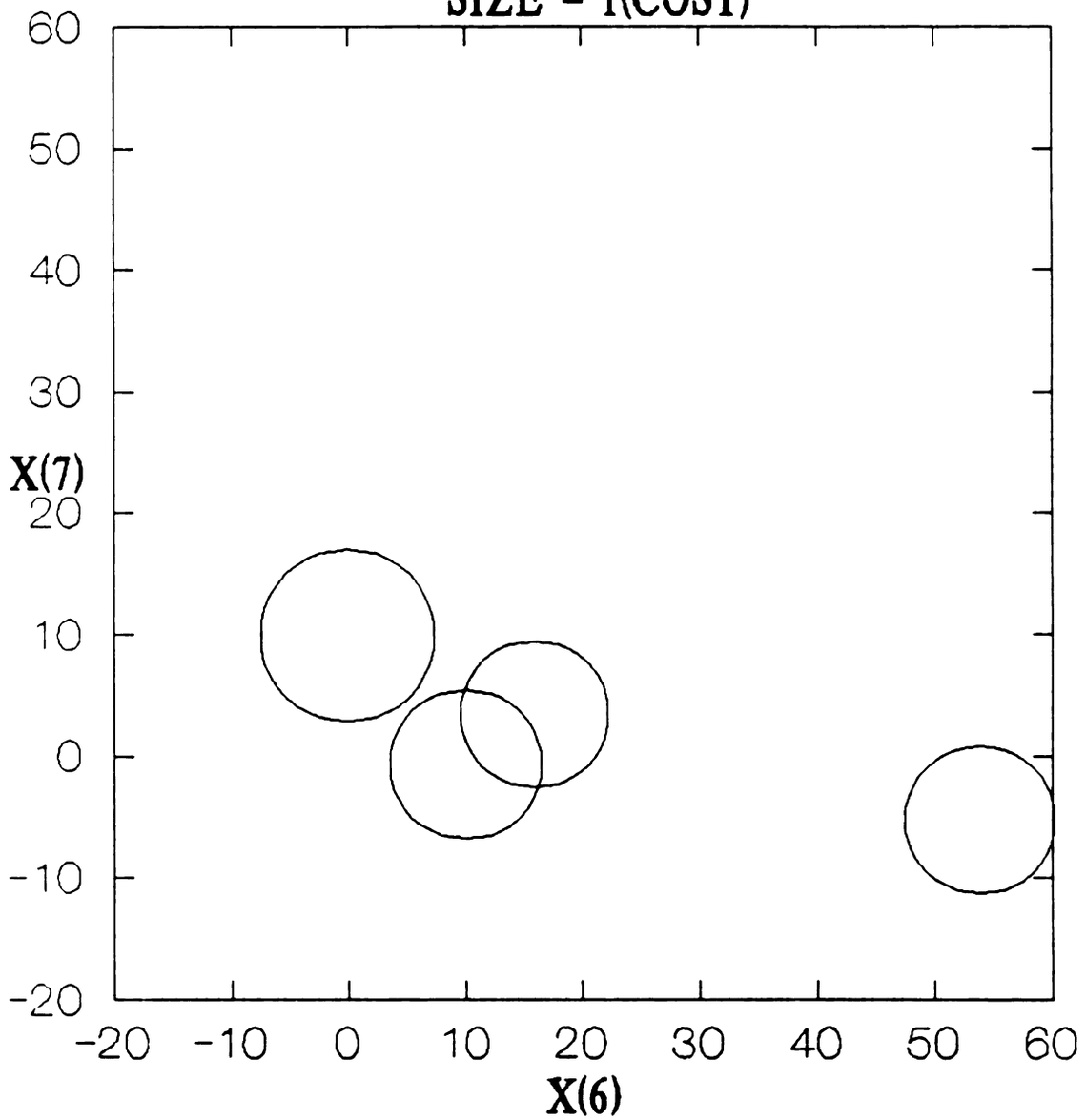
4.6 SUMMARY

The four experimental variables and each level of the variables were explained and discussed. The a priori hypotheses were then presented and explained. These hypotheses were based on the general research discussed in chapter 1. The full factorial design was used to avoid having higher order interactions confound other variables in the experiment.

FIGURE 4-5

CELL 3212: COST STRUCTURE 20/1/1

SIZE = f(COST)

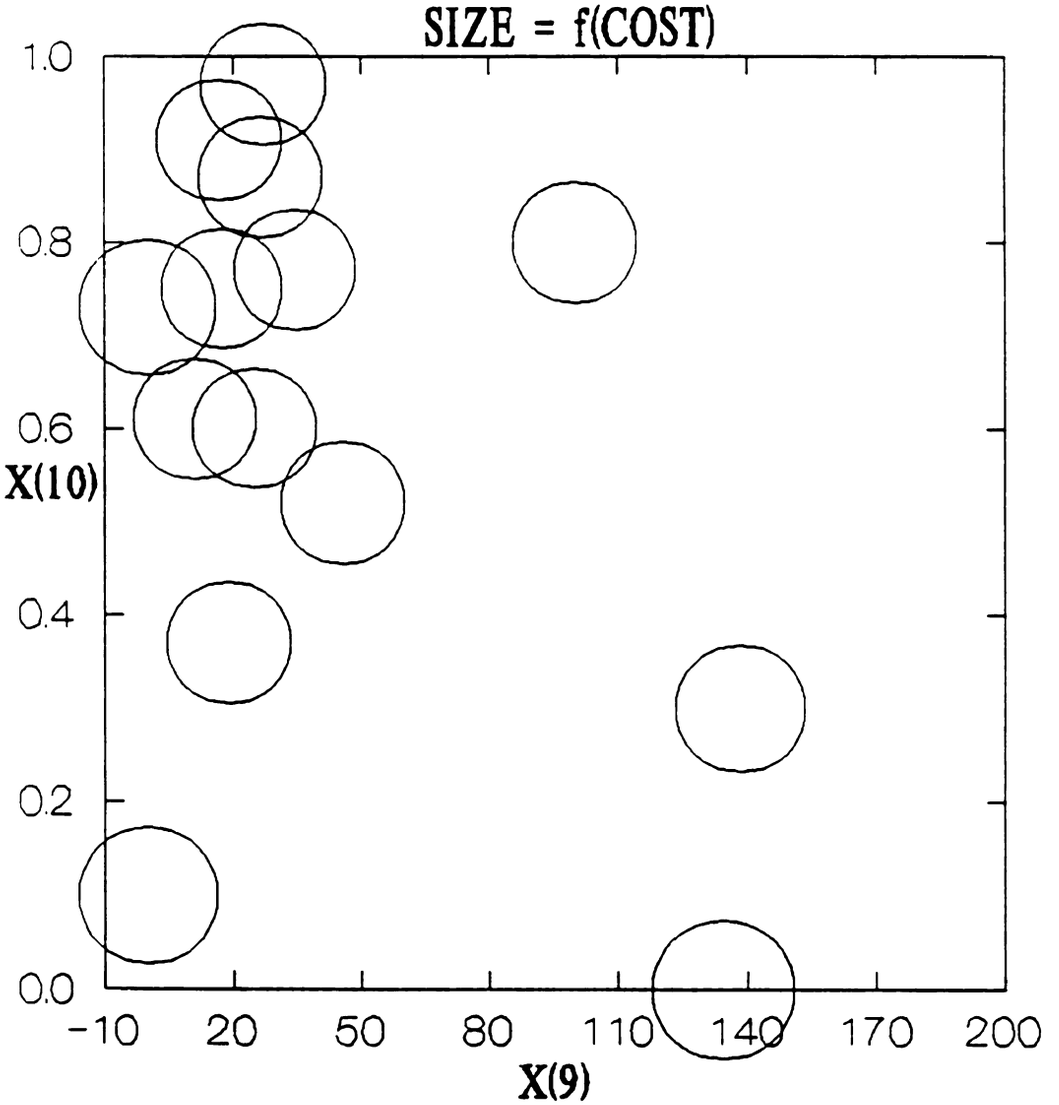


CMS: RD = DD - X(6)*PROC - X(7)*CONG

LEGEND FOR CELL LABEL
DIGIT 1=ORR. 2=MIL. 3=CMS. 4=MLC
DIGIT 2=WHEN: 1=QUE. 2=JOB. 3=CLR
DIGIT 3=DISPATCH: 1=FCFS. 2=MODD. 3=CMDD
DIGIT 4=PLANNING:1=SMOOTH.2=UNCONT

FIGURE 4-6

CELL 4311: COST STRUCTURE 20/1/1



MLC: RD=DD - PROC - DLOAD(WC1) -SUM{DLOAD(WC2 to WC5)/X(9)}
where DLOAD = SUM{(WORK/STAGE)*X(10)^(STAGE-1)}

LEGEND FOR CELL LABEL
DIGIT 1=ORR. 2=MIL. 3=CMS. 4=MLC
DIGIT 2=WHEN: 1=QUE. 2=JOB. 3=CLR
DIGIT 3=DISPATCH: 1=FCFS. 2=MODD. 3=CMDD
DIGIT 4=PLANNING:1=SMOOTH.2=UNCONT

TABLE 4-2
PARAMETERS IN EACH TREATMENT CELL

when Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM												
		SMOOTH PLANNING							UNCONTROLLED PLANNING					
		IMM	MIL		CMS		MLC		IMM	MIL		CMS		MLC	
			X6	X7	X6	X7	X9	X10		X6	X7	X6	X7	X9	X10
QUE	FCFS		10	-.7	10	-.7	40	.8		10	-.7	10	-.7	25	.6
	MODD		10	-.7	10	-.7	20	.4		10	-.7	10	-.7	190	.8
	CMDD		10	-.7	10	-.7	10	.5		10	-.7	10	-.7	150	.4
JOB	FCFS		10	-.7	10	-.7	15	.5		10	-.7	10	-.7	80	.8
	MODD		10	-.7	10	-.7	40	.8		10	-.7	10	-.7	45	.6
	CMDD		10	-.7	10	-.7	20	.4		10	-.7	10	-.7	50	.5
CLR	FCFS		10	-.7	10	-.7	5	.5		10	-.7	10	-.7	190	.9
	MODD		10	-.7	10	-.7	40	.8		10	-.7	10	-.7	50	.6
	CMDD		10	-.7	10	-.7	35	.7		10	-.7	10	-.7	40	.8

RELEASE DATES ARE CALCULATED USING THESE PARAMETERS AS SHOWN BELOW:

MIL: RD = DD - X(6)*PROC - X(7)*CONG

CMS: RD = DD - X(6)*PROC - X(7)*CONG

MLC: $RD = DD - PROC - DLOAD(WC_1) - \Sigma_{i=2}^{5}\{DLOAD(WC_i)/X(9)\}$

where $DLOAD(WC) = \Sigma_{j=1}^{5}(OPN_PROC_j)*X(10)^{j-1}$

LEGEND:

PROC = TOTAL PROCESSING TIME OF JOB; CONG = CONGESTION AT WORK CENTERS;

DLOAD(WC) = DISCOUNTED LOAD AT WORK CENTER;

OPN_PROC = OPERATION PROCESSING TIME.

v.2

MICHIGAN STATE UNIVERSITY LIBRARIES
3 1293 00908 2136

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
OCT 30 2001		

MSU Is An Affirmative Action/Equal Opportunity Institution
c:\circ\datedue.pm3-p.1

AN EXPERIMENTAL
OPERATING UNDER

AN EXPERIMENTAL INVESTIGATION OF INFORMATION USE IN A JOB SHOP OPERATING UNDER DUAL RESOURCE CONSTRAINTS: A SIMULATION STUDY

Volume II

By

Lawrence Dean Fredendall

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Department of Management

1991

This cha ter

he sensitivity nal

s the mean tot l c

otal cost. The e c

(in Table 5-2). he

number of labo tra

Ten aux lary

4. These perfo man

mental results. The

- T ble
- T ble
- T ble
- T ble
- T ble
- T ble
- T ble
- T ble
- T ble
- Ta ble
- Ta ble

CHAPTER 5

5.1 INTRODUCTION TO EXPERIMENTAL RESULTS

This chapter analyzes the results of the main simulation experiments and the sensitivity analysis. The simulation results consist of two sets of data. First is the mean total cost per week in Table 5-1 and second are the components of total cost. These component costs are the mean hours of tardy work per week (in Table 5-2), the mean total inventory per week (in Table 5-3), and the mean number of labor transfers per week (in Table 5-4).

Ten auxiliary performance measures were collected as discussed in chapter 4. These performance measures were collected to aid in explaining the experimental results. The measures are:

- Table 5-5 contains the mean number of hours tardy per tardy job.
- Table 5-6 contains the mean 90th percentile of tardiness.
- Table 5-7 contains the mean flow time per job.
- Table 5-8 contains the mean 90th percentile of flow time.
- Table 5-9 contains the mean time in the shop queue per job.
- Table 5-10 contains the mean time in the order pool per job.
- Table 5-11 contains the mean time in the system per job.
- Table 5-12 contains the mean percent of jobs tardy per week.
- Table 5-13 contains the mean finished goods inventory per week.
- Table 5-14 contains the mean work-in-process inventory per week.

TABLE 5-1
MEAN TOTAL COST/WEEK AND STANDARD DEVIATION*

PLANNING SYSTEM/JOB SELECTION MECHANISM			
SMOOTH		UNCONTROLLED	

TABLE 5-1
MEAN TOTAL COST/WEEK AND STANDARD DEVIATION*

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM: SMOOTH — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC	UNCONTROLLED — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC
QUE	FCFS	44.271	39.825	52.989	41.064	102.524	78.539	75.453	66.440
		(3.346)	(2.236)	(1.970)	(1.912)	(26.775)	(19.338)	(11.563)	(12.441)
	MODD	42.313	37.571	52.812	42.081	85.921	75.143	75.041	65.151
		(2.711)	(2.276)	(2.092)	(2.844)	(21.451)	(18.804)	(11.426)	(12.441)
	CMDD	47.109	41.160	53.140	35.987	95.463	81.165	75.329	73.670
		(3.034)	(2.332)	(1.989)	(1.746)	(23.027)	(19.085)	(11.516)	(13.830)
JOB	FCFS	47.289	45.200	54.144	38.716	107.058	83.395	76.563	67.224
		(3.297)	(2.165)	(2.018)	(1.860)	(27.403)	(19.196)	(11.534)	(12.725)
	MODD	46.701	44.146	53.878	51.740	90.921	81.830	75.749	64.690
		(2.956)	(3.577)	(2.004)	(36.480)	(22.381)	(18.819)	(11.296)	(13.643)
	CMDD	40.041	46.776	54.000	43.401	97.131	85.953	76.236	68.420
		(3.306)	(2.265)	(2.053)	(2.868)	(21.834	(18.208)	(11.644)	(13.971)
CLR	FCFS	45.996	43.961	53.971	39.699	105.740	82.228	76.135	68.280
		(3.301)	(2.132)	(1.815)	(1.974)	(27.404)	(19.284)	(11.343)	(12.245)
	MODD	46.052	42.201	53.634	41.699	90.419	80.254	75.613	64.906
		(2.869)	(2.121)	(2.160)	(2.030)	(17.578)	(19.009)	(11.192)	(13.816)
	CMDD	48.825	45.577	53.806	41.981	96.493	85.073	77.004	64.584
		(3.034)	(2.514)	(1.970)	(1.869)	(21.376)	(19.076)	(12.084)	(12.893)

* ENTRIES ARE COST (STANDARD DEVIATION)
QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE
IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-2
MEAN HOURS TARDY PER WEEK

	PLANNING SYSTEM/JOB SELECTION MECHANISM	
	SMOOTH	UNCONTROLLED
"WHEN"	ORDER REVIEW/RELEASE RULE	ORDER REVIEW/RELEASE RULE
Job		

TABLE 5-2
MEAN HOURS TARDY PER WEEK

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM: SMOOTH — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC	UNCONTROLLED — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC
QUE	FCFS	14.492	11.457	12.951	6.992	44.376	31.375	24.059	20.757
	MODD	13.954	10.475	12.502	8.589	36.124	29.898	23.862	19.448
	CMDD	16.392	12.298	12.651	4.822	40.894	32.884	23.983	24.802
JOB	FCFS	14.453	11.475	12.607	5.850	44.653	31.077	24.041	19.876
	MODD	14.297	11.279	12.459	6.777	37.363	30.748	23.616	19.002
	CMDD	16.279	12.512	12.511	8.644	40.190	32.702	23.855	21.169
CLR	FCFS	14.396	11.440	12.655	5.895	44.582	31.113	23.963	20.461
	MODD	14.475	10.713	12.473	6.885	37.572	30.378	23.690	19.256
	CMDD	16.158	12.463	12.553	7.014	40.356	32.830	24.370	18.775

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 2-1
MEAN HOURS TOTAL INVENTORY PER WEEK

TABLE 5-3
MEAN HOURS TOTAL INVENTORY PER WEEK

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	110.824	110.447	110.867	101.487	109.050	106.058	104.883	78.957
	MODD	111.023	110.204	110.885	79.707	108.712	105.131	104.879	92.404
	CMDD	111.190	110.273	110.893	93.850	109.555	105.722	104.914	70.364
JOB	FCFS	110.865	110.495	110.883	89.292	109.115	106.239	104.998	93.096
	MODD	111.079	110.323	110.878	101.449	108.280	105.446	104.983	84.679
	CMDD	111.327	110.414	110.875	79.723	109.146	105.739	104.944	78.278
CLR	FCFS	111.019	110.502	110.883	102.592	109.161	106.260	104.968	95.060
	MODD	111.107	110.270	110.878	101.536	108.362	105.213	104.932	84.353
	CMDD	111.294	110.422	110.887	101.502	109.167	105.840	104.862	90.712

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-4
MEAN NUMBER OF LABOR TRANSFERS PER WEEK

PLANNING SYSTEM/JOB SELECTION MECHANISM	
SMOOTH	

TABLE 5-4
MEAN NUMBER OF LABOR TRANSFERS PER WEEK

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	33.048	58.658	167.196	169.313	28.665	51.835	168.474	170.294
	MODD	33.027	56.018	167.186	169.329	28.021	48.350	168.298	170.141
	CMDD	32.056	55.264	167.491	169.580	27.189	48.239	168.711	170.308
JOB	FCFS	72.968	112.006	178.420	180.864	68.401	106.162	179.814	181.635
	MODD	70.000	105.555	178.725	179.181	63.669	97.902	180.188	182.182
	CMDD	63.513	107.105	178.904	181.408	58.366	99.754	180.315	182.554
CLR	FCFS	61.024	100.318	175.732	176.494	56.596	93.756	177.122	178.526
	MODD	59.915	97.471	176.007	177.757	54.398	89.769	177.389	179.595
	CMDD	53.788	96.092	176.116	178.032	48.644	88.284	177.782	179.632

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5.5
MEAN HOURS TARDY PER TARDY JOB

"RULE"	Job Selection Mechanism	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
SPT		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
EDD		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
MOD		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
CR		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
S/OPN		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
S/RPC		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
S/SLK		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
WINQ		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
COVERT		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
ATC		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
RANDOM		1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1

TABLE 5-5
MEAN HOURS TARDY PER TARDY JOB

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	3.109	4.759	4.361	6.186	7.894	9.710	9.682	9.549
	MODD	3.817	3.831	4.331	4.776	9.262	8.145	9.680	9.127
	CMDD	3.770	4.141	4.311	5.174	9.094	8.394	9.699	9.500
JOB	FCFS	3.161	4.823	4.376	5.070	7.877	10.075	9.751	9.164
	MODD	3.919	3.946	4.332	9.659	8.359	8.112	9.798	9.519
	CMDD	3.809	4.433	4.321	4.908	8.596	8.684	9.759	9.695
CLR	FCFS	3.204	4.848	4.366	5.384	7.938	10.052	9.722	9.143
	MODD	3.930	3.940	4.334	6.166	8.413	8.132	9.740	9.651
	CMDD	3.817	4.422	4.298	6.081	8.597	8.870	9.910	9.431

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-6
MEAN 90th PERCENTILE OF TARDINESS*

"WHEN"	Job Dispatch	PLANNING SYSTEM/JOB SELECTION MECHANISM	
		SMOOTH	UNCONTROLLED
Labor		ORDER REVIEW/RELEASE RULE	ORDER REVIEW/RELEASE RULE

TABLE 5-6
MEAN 90th PERCENTILE OF TARDINESS*

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	6.743	9.907	8.533	15.387	19.513	24.203	25.033	23.961
	MODD	7.916	8.807	8.510	11.626	23.179	19.907	25.379	23.290
	CMDD	7.805	8.583	8.459	12.738	22.742	20.304	25.237	23.615
JOB	FCFS	6.703	10.024	8.552	12.474	19.343	25.216	25.450	23.802
	MODD	7.961	8.518	8.488	16.344	19.888	19.833	25.797	23.838
	CMDD	7.631	9.392	8.497	11.866	20.644	20.796	25.569	24.243
CLR	FCFS	6.818	9.974	8.580	14.010	19.452	25.225	25.363	23.984
	MODD	7.995	8.531	8.510	15.382	20.124	19.922	25.518	24.435
	CMDD	7.664	9.326	8.478	15.386	20.486	21.348	26.355	24.116

* TARDINESS DATA IS COLLECTED ONLY FOR JOBS WHICH WERE TARDY

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-7
MEAN FLOW TIME PER JOB*

"WHEN"	Job	PLANNING SYSTEM/JOB SELECTION MECHANISM	
		SMOOTH	UNCONTROLLED
		ORDER REVIEW/RELEASE RULE	

TABLE 5-7
MEAN FLOW TIME PER JOB*

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	27.292	17.579	4.830	4.764	30.584	19.467	4.833	4.777
	MODD	27.896	17.487	4.836	4.769	30.509	19.052	4.829	4.801
	CMDD	25.393	16.851	4.763	4.678	28.337	18.602	4.764	4.709
JOB	FCFS	27.560	17.418	4.825	4.739	30.868	19.373	4.829	4.777
	MODD	28.361	18.008	4.832	4.916	30.873	19.696	4.823	4.779
	CMDD	25.923	16.970	4.767	4.691	28.459	18.722	4.773	4.708
CLR	FCFS	27.529	17.424	4.824	4.790	30.851	19.375	4.830	4.791
	MODD	28.368	17.450	4.833	4.790	30.878	19.050	4.824	4.781
	CMDD	25.888	16.983	4.773	4.707	28.490	18.776	4.776	4.713

* FLOW TIME BEGINS WHEN THE JOB IS RELEASED FROM THE ORDER POOL

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-8
*MEAN 90th PERCENTILE OF FLOW TIME**

PLANNING SYSTEM/JOB SELECTION MECHANISM			
	SMOOTH		UNCONTROLLED
	ORDER RELEASE/RELEASE RULE		ORDER RELEASE/RELEASE RULE
	1	2	3
1	1.4	1.4	1.4
2	1.4	1.4	1.4
3	1.4	1.4	1.4
4	1.4	1.4	1.4
5	1.4	1.4	1.4
6	1.4	1.4	1.4
7	1.4	1.4	1.4
8	1.4	1.4	1.4
9	1.4	1.4	1.4
10	1.4	1.4	1.4
11	1.4	1.4	1.4
12	1.4	1.4	1.4
13	1.4	1.4	1.4
14	1.4	1.4	1.4
15	1.4	1.4	1.4
16	1.4	1.4	1.4
17	1.4	1.4	1.4
18	1.4	1.4	1.4
19	1.4	1.4	1.4
20	1.4	1.4	1.4
21	1.4	1.4	1.4
22	1.4	1.4	1.4
23	1.4	1.4	1.4
24	1.4	1.4	1.4
25	1.4	1.4	1.4
26	1.4	1.4	1.4
27	1.4	1.4	1.4
28	1.4	1.4	1.4
29	1.4	1.4	1.4
30	1.4	1.4	1.4
31	1.4	1.4	1.4
32	1.4	1.4	1.4
33	1.4	1.4	1.4
34	1.4	1.4	1.4
35	1.4	1.4	1.4
36	1.4	1.4	1.4
37	1.4	1.4	1.4
38	1.4	1.4	1.4
39	1.4	1.4	1.4
40	1.4	1.4	1.4
41	1.4	1.4	1.4
42	1.4	1.4	1.4
43	1.4	1.4	1.4
44	1.4	1.4	1.4
45	1.4	1.4	1.4
46	1.4	1.4	1.4
47	1.4	1.4	1.4
48	1.4	1.4	1.4
49	1.4	1.4	1.4
50	1.4	1.4	1.4
51	1.4	1.4	1.4
52	1.4	1.4	1.4
53	1.4	1.4	1.4
54	1.4	1.4	1.4
55	1.4	1.4	1.4
56	1.4	1.4	1.4
57	1.4	1.4	1.4
58	1.4	1.4	1.4
59	1.4	1.4	1.4
60	1.4	1.4	1.4
61	1.4	1.4	1.4
62	1.4	1.4	1.4
63	1.4	1.4	1.4
64	1.4	1.4	1.4
65	1.4	1.4	1.4
66	1.4	1.4	1.4
67	1.4	1.4	1.4
68	1.4	1.4	1.4
69	1.4	1.4	1.4
70	1.4	1.4	1.4
71	1.4	1.4	1.4
72	1.4	1.4	1.4
73	1.4	1.4	1.4
74	1.4	1.4	1.4
75	1.4	1.4	1.4
76	1.4	1.4	1.4
77	1.4	1.4	1.4
78	1.4	1.4	1.4
79	1.4	1.4	1.4
80	1.4	1.4	1.4
81	1.4	1.4	1.4
82	1.4	1.4	1.4
83	1.4	1.4	1.4
84	1.4	1.4	1.4
85	1.4	1.4	1.4
86	1.4	1.4	1.4
87	1.4	1.4	1.4
88	1.4	1.4	1.4
89	1.4	1.4	1.4
90	1.4	1.4	1.4
91	1.4	1.4	1.4
92	1.4	1.4	1.4
93	1.4	1.4	1.4
94	1.4	1.4	1.4
95	1.4	1.4	1.4
96	1.4	1.4	1.4
97	1.4	1.4	1.4
98	1.4	1.4	1.4
99	1.4	1.4	1.4
100	1.4	1.4	1.4
101	1.4	1.4	1.4
102	1.4	1.4	1.4
103	1.4	1.4	1.4
104	1.4	1.4	1.4
105	1.4	1.4	1.4
106	1.4	1.4	1.4
107	1.4	1.4	1.4
108	1.4	1.4	1.4
109	1.4	1.4	1.4
110	1.4	1.4	1.4
111	1.4	1.4	1.4
112	1.4	1.4	1.4
113	1.4	1.4	1.4
114	1.4	1.4	1.4
115	1.4	1.4	1.4
116	1.4	1.4	1.4
117	1.4	1.4	1.4
118	1.4	1.4	1.4
119	1.4	1.4	1.4
120	1.4	1.4	1.4
121	1.4	1.4	1.4
122	1.4	1.4	1.4
123	1.4	1.4	1.4
124	1.4	1.4	1.4
125	1.4	1.4	1.4
126	1.4	1.4	1.4
127	1.4	1.4	1.4
128	1.4	1.4	1.4
129	1.4	1.4	1.4
130	1.4	1.4	1.4
131	1.4	1.4	1.4
132	1.4	1.4	1.4
133	1.4	1.4	1.4
134	1.4	1.4	1.4
135	1.4	1.4	1.4
136	1.4	1.4	1.4
137	1.4	1.4	1.4
138	1.4	1.4	1.4
139	1.4	1.4	1.4
140	1.4	1.4	1.4
141	1.4	1.4	1.4
142	1.4	1.4	1.4
143	1.4	1.4	1.4
144	1.4	1.4	1.4
145	1.4	1.4	1.4
146	1.4	1.4	1.4
147	1.4	1.4	1.4
148	1.4	1.4	1.4
149	1.4	1.4	1.4
150	1.4	1.4	1.4
151	1.4	1.4	1.4
152	1.4	1.4	1.4
153	1.4	1.4	1.4
154	1.4	1.4	1.4
155	1.4	1.4	1.4
156	1.4	1.4	1.4
157	1.4	1.4	1.4
158	1.4	1.4	1.4
159	1.4	1.4	1.4
160	1.4	1.4	1.4
161	1.4	1.4	1.4
162	1.4	1.4	1.4
163	1.4	1.4	1.4
164	1.4	1.4	1.4
165	1.4	1.4	1.4
166	1.4	1.4	1.4
167	1.4	1.4	1.4
168	1.4	1.4	1.4
169	1.4	1.4	1.4
170	1.4	1.4	1.4
171	1.4	1.4	1.4
172	1.4	1.4	1.4
173	1.4	1.4	1.4
174	1.4	1.4	1.4
175	1.4	1.4	1.4
176	1.4	1.4	1.4
177	1.4	1.4	1.4
178	1.4	1.4	1.4
179	1.4	1.4	1.4
180	1.4	1.4	1.4
181	1.4	1.4	1.4
182	1.4	1.4	1.4
183	1.4	1.4	1.4
184	1.4	1.4	1.4
185	1.4	1.4	1.4
186	1.4	1.4	1.4
187	1.4	1.4	1.4
188	1.4	1.4	1.4
189	1.4	1.4	1.4
190	1.4	1.4	1.4
191	1.4	1.4	1.4
192	1.4	1.4	1.4
193	1.4	1.4	1.4
194	1.4	1.4	1.4
195	1.4	1.4	1.4
196	1.4	1.4	1.4
197	1.4	1.4	1.4
198	1.4	1.4	1.4
199	1.4	1.4	1.4
200	1.4	1.4	1.4
201	1.4	1.4	1.4
202	1.4	1.4	1.4
203	1.4	1.4	1.4
204	1.4	1.4	1.4
205	1.4	1.4	1.4
206	1.4	1.4	1.4
207	1.4	1.4	1.4
208	1.4	1.4	1.4
209	1.4	1.4	1.4
210	1.4	1.4	1.4
211	1.4	1.4	1.4
212	1.4	1.4	1.4
213	1.4	1.4	1.4
214	1.4	1.4	1.4
215	1.4	1.4	1.4
216	1.4	1.4	1.4
217	1.4	1.4	1.4
218	1.4	1.4	1.4
219	1.4	1.4	1.4
220	1.4	1.4	1.4
221	1.4	1.4	1.4
222	1.4	1.4	1.4
223	1.4	1.4	1.4
224	1.4	1.4	1.4
225	1.4	1.4	1.4
226	1.4	1.4	1.4
227	1.4	1.4	1.4
228	1.4	1.4	1.4
229	1.4	1.4	1.4
230	1.4	1.4	1.4
231	1.4	1.4	1.4
232	1.4	1.4	1.4
233	1.4	1.4	1.4
234	1.4	1.4	1.4
235	1.4	1.4	1.4
236	1.4	1.4	1.4
237	1.4	1.4	1.4
238	1.4	1.4	1.4
239	1.4	1.4	1.4
240	1.4	1.4	1.4
241	1.4	1.4	1.4
242	1.4	1.4	1.4
243	1.4	1.4	1.4
244	1.4	1.4	1.4
245	1.4	1.4	1.4
246	1.4	1.4	1.4
247	1.4	1.4	1.4
248	1.4	1.4	1.4
249	1.4	1.4	1.4
250	1.4	1.4	1.4
251	1.4	1.4	1.4
252	1.4	1.4	1.4
253	1.4	1.4	1.4
254	1.4	1.4	1.4
255	1.4	1.4	1.4
256	1.4	1.4	1.4
257	1.4	1.4	1.4
258	1.4	1.4	1.4
259	1.4	1.4	1.4
260	1.4	1.4	1.4
261	1.4	1.4	1.4
262	1.4	1.4	1.4
263	1.4	1.4	1.4
264	1.4	1.4	1.4
265	1.4	1.4	1.4
266	1.4	1.4	1.4
267	1.4	1.4	1.4
268	1.4	1.4	1.4
269	1.4	1.4	1.4
270	1.4	1.4	1.4
271	1.4	1.4	1.4
272	1.4	1.4	1.4
273	1.4	1.4	1.4
274	1.4	1.4	1.4
275	1.4	1.4	1.4
276	1.4	1.4	1.4
277	1.4	1.4	1.4
278	1.4	1.4	1.4
279	1.4	1.4	1.4
280	1.4	1.4	1.4
281	1.4	1.4	1.4
282	1.4	1.4	1.4
283	1.4	1.4	1.4
284	1.4	1.4	1.4
285	1.4	1.4	1.4
286	1.4	1.4	1.4
287	1.4	1.4	1.4
288	1.4	1.4	1.4
289	1.4	1.4	1.4
290	1.4	1.4	1.4
291	1.4	1.4	1.4
292	1.4	1.4	1.4
293	1.4	1.4	1.4
294	1.4	1.4	1.4
295	1.4	1.4	1.4
296	1.4	1.4	1.4
297	1.4	1.4	1.4
298	1.4	1.4	1.4
299	1.4	1.4	1.4
300	1.4	1.4	1.4
301	1.4	1.4	1.4
302	1.4	1.4	1.4
303	1.4	1.4	1.4
304	1.4	1.4	1.4
305	1.4	1.4	1.4
306	1.4	1.4	1.4
307	1.4	1.4	1.4
308	1.4	1.4	1.4
309	1.4	1.4	1.4
310	1.4	1.4	1.4
311	1.4	1.4	1.4
312	1.4	1.4	1.4
313	1.4	1.4	1.4
314	1.4	1.4	1.4
315	1.4	1.4	1.4
316	1.4	1.4	1.4
317	1.4	1.4	1.4
318	1.4	1.4	1.4
319	1.4	1.4	1.4
320	1.4	1.4	1.4
321	1.4	1.4	1.4
322	1.4	1.4	1.4
323	1.4	1.4	1.4
324	1.4	1.4	1.4
325	1.4	1.4	1.4
326	1.4	1.4	1.4
327	1.4	1.4	1.4
328	1.4	1.4	1.4
329	1.4	1.4	1.4
330	1.4	1.4	1.4
331	1.4	1.4	1.4
332	1.4	1.4	1.4
333	1.4	1.4	1.4
334	1.4	1.4	1.4
335	1.4	1.4	1.4
336	1.4	1.4	1.4
337	1.4	1.4	1.4
338	1.4	1.4	1.4
339	1.4	1.4	1.4
340	1.4	1.4	1.4
341	1.4	1.4	1.4
342	1.4	1.4	1.4
343	1.4	1.4	1.4
344	1.4	1.4	1.4
345	1.4	1.4	1.4
346	1.4	1.4	1.4
347	1.4	1.4	1.4
348	1.4	1.4	1.4
349	1.4	1.4	1.4
350	1.4	1.4	1.4
351	1.4	1.4	1.4
352	1.4	1.4	1.4
353	1.4	1.4	1.4
354	1.4	1.4	1.4
355	1.4	1.4	1.4
356	1.4	1.4	1.4
357	1.4	1.4	1.4
358	1.4	1.4	1.4
359	1.4	1.4	1.4
360	1.4	1.4	1.4
361	1.4	1.4	1.4
362	1.4	1.4	1.4
363	1.4	1.4	1.4
364	1.4	1.4	1.4
365	1.4	1.4	1.4
366	1.4	1.4	1.4
367	1.4	1.4	1.4
368	1.4	1.4	1.4
369	1.4	1.4	1.4
370	1.4	1.4	1.4
371	1.4	1.4	1.4
372	1.4	1.4	1.4
373	1.4	1.4	1.4
374	1.4	1.4	1.4
375	1.4	1.4	1.4
376	1.4	1.4	1.4
377	1.4	1.4	1.4
378	1.4	1.4	1.4
379	1.4	1.4	1.4
380	1.4	1.4	1.4
381	1.4	1.4	1.4
382	1.4	1.4	1.4
383	1.4	1.4	1.4
384	1.4	1.4	1.4
385	1.4	1.4	1.4
386	1.4	1.4	1.4
387	1.4	1.4	1.4
388	1.4	1.4	1.4
389	1.4	1.4	1.4
390	1.4	1.4	1.4
391	1.4	1.4	1.4
392	1.4	1.4	1.4
393	1.4	1.4	1.4
394	1.4	1.4	1.4
395	1.4	1.4	1.4
396	1.4	1.4	1.4
397	1.4	1.4	1.4
398	1.4	1.4	1.4
399	1.4	1.4	1.4
400	1.4	1.4	1.4
401	1.4	1.4	1.4
402	1.4	1.4	1.4
403	1.4	1.4	1.4
404	1.4	1.4	1.4
405	1.4	1.4	1.4
406	1.4	1.4	1.4
407	1.4	1.4	1.4
408	1.4	1.4	1.4
409	1.4	1.4	1.4
410	1.4	1.4	1.4
411	1.4	1.4	1.4
412	1.4	1.4	1.4
413	1.4	1.4	1.4
414	1.4	1.4	1.4
415	1.4	1.4	1.4
416	1.4	1.4	1.4
417	1.4	1.4	1.4
418	1.4	1.4	1.4
419	1.4	1.4	1.4
420	1.4	1.4	1.4
421	1.4	1.4	1.4
422	1.4	1.4	1.4
423	1.4	1.4	1.4
424	1.4	1.4	1.4
425	1.4	1.4	1.4
426	1.4	1.4	1.4
427	1.4	1.4	1.4
428	1.4	1.4	1.4
429	1.4	1.4	1.4
430	1.4	1.4	1.4
431	1.4	1.4	1.4
432	1.4	1.4	1.4
433	1.4	1.4	1.4
434	1.4	1.4	1.4
435	1.4	1.4	1.4
436	1.4	1.4	1.4
437	1.4	1.4	1.4
438	1.4	1.4	1.4
439	1.4	1.4	1.4
440	1.4	1.4	1.4
441	1.4	1.4	1.4
442	1.4	1.4	1.4
443	1.4	1.4	1.4
444	1.4	1.4	1.4
445	1.4	1.4	1.4
446	1.4	1.4	1.4
447	1.4	1.4	1.4
448	1.4	1.4	1.4
449	1.4	1.4	1.4
450	1.4	1.4	1.4
451	1.4	1.4	1.4
452	1.4	1.4	1.4
453	1.4	1.4	1.4
454	1.4	1.4	1.4
455	1.4	1.4	1.4
456	1.4	1.4	1.4
457	1.4	1.4	1.4
458	1.4	1.4	1.4
459	1.4	1.4	1.4
460	1.4	1.4	1.4
461	1.4	1.4	1.4
462	1.4	1.4	1.4
463	1.4	1.4	1.4
464	1.4	1.4	1.4
465	1.4	1.4	1.4
466	1.4	1.4	1.4
467	1.4	1.4	1.4
468	1.4	1.4	1.4
469	1.4	1.4	1.4
470	1.4	1.4	1.4
471	1.4	1.4	1.4
472	1.4	1.4	1.4
473	1.4	1.4	1.4
474	1.4	1.4	1.4
475	1.4	1.4	1.4
476	1.4	1.4	1.4
477	1.4	1.4	1.4
478	1.4	1.4	1.4
479	1.4	1.4	1.4
480	1.4	1.4	1.4
481	1.4	1.4	1.4
482	1.4	1.4	1.4
483	1.4	1.4	1.4
484	1.4	1.4	1.4
485	1.4	1.4	1.4
486	1.4	1.4	1.4
487	1.4	1.4	1.4
488	1.4	1.4	1.4
489	1.4	1.4	1.4
490	1.4	1.4	1.4
491	1.4	1.4	1.4
492	1.4	1.4	1.4
493	1.4	1.4	1.4
494	1.4	1.4	1.4
495	1.4	1.4	1.4
496	1.4	1.4	1.4
497	1.4	1.4	1.4
498	1.4	1.4	1.4
499	1.4	1.4	1.4
500	1.4	1.4	1.4

TABLE 5-8
MEAN 90th PERCENTILE OF FLOW TIME*

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	36.411	30.965	8.767	8.353	44.486	34.433	8.723	8.406
	MODD	36.450	31.937	8.865	8.422	42.941	35.416	8.822	8.513
	CMDD	35.982	31.623	8.728	8.286	42.976	34.826	8.707	8.388
JOB	FCFS	36.438	31.034	8.766	8.305	45.073	34.634	8.746	8.411
	MODD	36.575	32.269	8.881	8.796	42.958	35.744	8.848	8.476
	CMDD	36.208	31.641	8.750	8.315	42.764	34.527	8.730	8.373
CLR	FCFS	36.427	31.055	8.768	8.391	44.988	34.634	8.749	8.434
	MODD	36.606	32.048	8.879	8.469	43.114	35.282	8.836	8.480
	CMDD	36.184	31.624	8.767	8.344	42.938	34.624	8.746	8.387

* FLOW TIME BEGINS WHEN THE JOB IS RELEASED FROM THE ORDER POOL

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-9
MEAN TIME IN QUEUE ON SHOP FLOOR PER JOB*

"WHEN" / Job	PLANNING SYSTEM/JOB SELECTION MECHANISM			
	SMOOTH		UNCONTROLLED	
	ORDER REVIEW/RELEASE RULE		ORDER REVIEW/RELEASE RULE	

TABLE 5-9
MEAN TIME IN QUEUE ON SHOP FLOOR PER JOB*

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	23.290	13.577	0.834	0.770	26.581	15.465	0.836	0.780
	MODD	23.894	13.484	0.840	0.774	26.508	15.051	0.833	0.806
	CMDD	21.391	12.849	0.767	0.693	24.335	14.600	0.767	0.714
JOB	FCFS	23.558	13.416	0.830	0.744	26.866	15.371	0.833	0.782
	MODD	24.359	14.007	0.838	0.922	26.872	15.695	0.828	0.784
	CMDD	21.921	12.969	0.772	0.695	24.457	14.719	0.777	0.713
CLR	FCFS	23.528	13.421	0.828	0.795	26.849	15.372	0.835	0.795
	MODD	24.366	13.447	0.837	0.796	26.877	15.049	0.828	0.787
	CMDD	21.886	12.981	0.778	0.712	24.488	14.773	0.775	0.718

* THIS DOES NOT INCLUDE TIME IN THE ORDER POOL
QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-10
MEAN TIME IN ORDER POOL PER JOB

PLANNING SYSTEM/JOB SELECTION MECHANISM		
	SMOOTH	UNCONTROLLED
"WILLY"		

TABLE 5-10
MEAN TIME IN ORDER POOL PER JOB

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	0	10.243	18.118	16.919	0	11.618	20.504	21.776
	MODD	0	10.288	18.147	20.691	0	11.549	20.505	19.317
	CMDD	0	10.110	18.141	18.150	0	11.371	20.516	23.447
JOB	FCFS	0	10.132	18.125	18.907	0	11.509	20.479	19.235
	MODD	0	9.581	18.157	17.740	0	10.605	20.502	20.671
	CMDD	0	10.093	18.133	20.703	0	11.358	20.520	21.900
CLR	FCFS	0	10.136	18.125	16.829	0	11.501	20.471	18.926
	MODD	0	10.181	18.149	16.932	0	11.410	20.493	20.742
	CMDD	0	10.100	18.144	16.951	0	11.374	20.611	19.639

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5.11
MEAN TIME IN SYSTEM PER JOB

"WHEN"	Job	PLANNING SYSTEM/JOB SELECTION MECHANISM			
		SMOOTH		UNCONTROLLED	
		ORDER REVIEW/RELEASE RULE		ORDER REVIEW/RELEASE RULE	
		IMM	MLL	IMM	MLC
Machine Idle	SPT	1.00	1.00	1.00	1.00
	EDD	1.00	1.00	1.00	1.00
	S/OPN	1.00	1.00	1.00	1.00
	CR	1.00	1.00	1.00	1.00
Machine Busy	SPT	1.00	1.00	1.00	1.00
	EDD	1.00	1.00	1.00	1.00
	S/OPN	1.00	1.00	1.00	1.00
	CR	1.00	1.00	1.00	1.00

TABLE 5-11
MEAN TIME IN SYSTEM PER JOB

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	27.292	27.822	22.948	21.683	30.584	31.085	25.337	26.553
	MODD	27.896	27.775	22.983	25.460	30.509	30.601	25.334	24.118
	CMDD	25.393	26.961	22.904	22.828	28.337	29.973	25.280	28.156
JOB	FCFS	27.560	27.550	22.950	23.646	30.868	30.882	25.308	24.012
	MODD	28.361	27.589	22.989	22.656	30.873	30.301	25.325	25.450
	CMDD	25.923	27.063	22.900	25.394	28.459	30.080	25.293	26.608
CLR	FCFS	27.529	27.560	22.949	21.619	30.851	30.876	25.301	23.717
	MODD	28.368	27.631	22.982	21.722	30.878	30.460	25.317	25.523
	CMDD	25.888	27.083	22.917	21.658	28.490	30.150	25.387	24.352

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5.12
MEAN PERCENT JOBS TARDY PER WEEK

"WHEN" Labor Assignment	Job Dispatch	PLANNING SYSTEM/JOB SELECTION MECHANISM					
		SMOOTH				UNCONTROLLED	
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE	
		IMM	MIL	CMS	MLC	IMM	MIL

TABLE 5-12
MEAN PERCENT JOBS TARDY PER WEEK

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	7.3	12.4	10.0	4.1	19.8	23.2	16.1	9.9
	MODD	7.5	10.2	9.8	4.5	16.0	20.0	15.8	9.6
	CMDD	7.5	11.5	9.9	2.8	16.8	22.1	15.9	11.5
JOB	FCFS	7.4	12.2	10.0	3.3	20.4	22.8	16.0	9.9
	MODD	8.9	10.5	9.9	4.0	17.8	20.1	15.7	4.0
	CMDD	8.2	12.1	9.8	4.5	16.8	22.7	15.9	9.2
CLR	FCFS	7.4	12.2	10.0	3.5	20.3	22.8	15.9	10.0
	MODD	9.0	10.4	9.8	4.0	18.0	20.5	15.7	10.1
	CMDD	8.1	12.0	9.9	9.2	16.9	22.8	16.1	9.3

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-11
MEAN HOURS FINISHED GOODS INVENTORY/WEEK

"WHEN" Labor	Job Dispatch	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTHED LOAD				UNCONTROLLED LOAD			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
Infinite	FCFS	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	SPT	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	EDD	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	S/OPN	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	CR	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
Finite	FCFS	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	SPT	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	EDD	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	S/OPN	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	CR	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000

TABLE 5-13
MEAN HOURS FINISHED GOODS INVENTORY/WEEK

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTHED LOAD				UNCONTROLLED LOAD			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	59.336	70.702	99.499	90.953	49.465	62.226	93.883	68.237
	MODD	58.231	68.804	99.389	68.973	50.609	60.036	93.662	81.485
	CMDD	68.291	74.587	99.561	83.445	59.606	65.890	93.999	59.751
JOB	FCFS	58.418	72.574	99.480	78.843	48.606	63.954	94.030	82.433
	MODD	57.354	69.986	99.456	89.541	49.785	61.441	93.803	73.840
	CMDD	67.121	75.657	99.502	69.282	59.238	67.169	94.012	67.688
CLR	FCFS	58.475	72.520	99.511	92.027	48.726	63.947	93.992	84.351
	MODD	57.297	70.330	99.401	90.769	49.855	61.521	93.746	73.492
	CMDD	67.146	75.523	99.521	91.031	59.044	67.056	93.906	80.107

* TARDINESS DATA IS COLLECTED ONLY FOR JOBS WHICH WERE TARDY
QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE
CMDD = CRITICAL MACHINE DUE DATE
IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION

TABLE 5-14
MEAN WORK IN PROCESS INVENTORY/WEEK

"WHEN" Labor	Job Dispatch	PLANNING SYSTEM/JOB SELECTION MECHANISM: SMOOTHED LOAD — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC	UNCONTROLLED LOAD — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC

TABLE 5-14
MEAN WORK IN PROCESS INVENTORY/WEEK

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTHED LOAD				UNCONTROLLED LOAD			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	51.488	39.746	10.905	10.534	59.585	43.832	11.000	10.720
	MODD	52.792	41.400	11.134	10.734	58.104	45.096	11.217	10.919
	CMDD	42.899	35.786	10.810	10.405	49.949	39.832	10.915	10.613
JOB	FCFS	52.447	37.922	10.857	10.449	60.508	42.285	10.969	10.663
	MODD	53.726	40.337	11.107	11.166	58.494	44.005	11.181	10.839
	CMDD	44.206	34.756	10.816	10.441	49.908	38.750	10.932	10.590
CLR	FCFS	52.544	37.982	10.869	10.565	60.434	42.313	10.976	10.709
	MODD	53.810	39.940	11.114	10.767	58.507	43.692	11.186	10.861
	CMDD	44.148	34.898	10.838	10.470	50.123	38.785	10.957	10.606

* TARDINESS DATA IS COLLECTED ONLY FOR JOBS WHICH WERE TARDY
QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE
CMDD = CRITICAL MACHINE DUE DATE
IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION

After com

measures, a sens

structure on the

because the com

number of labor

by the appropria

The analys

assumptions. The

analysis was con

Before tes

simulation runs w

1. Fit

2. Che

3. Che

4. Sel

as

was

fie

this

neo

After completing the analysis of the main and the auxiliary performance measures, a sensitivity analysis was conducted to examine the impact of the cost structure on the mean total cost per week. This did not require additional runs because the component costs (hours of tardy work, total inventory, and the number of labor transfers) were saved. These cost components were multiplied by the appropriate cost ratios to represent different cost structures.

5.1.1 OVERVIEW OF ANALYSIS

The analysis consisted of first determining if the data met all the statistical assumptions. Then the a priori hypotheses were examined and finally a post hoc analysis was conducted.

Before testing the a priori hypotheses, the data generated during the simulation runs was initially analyzed using a four step procedure:

1. Fit raw data to ANOVA model;
2. Check residuals for normality;
3. Check residuals for homogeneity of variances; and,
4. Select appropriate transformation if data fails to satisfy conditions as described in Steps 2 and 3. The purpose of this procedure was to ensure that the data used for the subsequent analysis satisfied the ANOVA assumptions. The two assumptions examined in this chapter were that the residuals were normal and had homogeneous variances (Scheffe, 1959, pp.331-369). The third assump-

tic

ch

After ens

steps were taken.

- Sig
  and
  tion
  the
  var
- The
  mai
  This
  not
  p.3
- Pos
  ma

Before be

calculations wi

5

This sect

unless stated o

tion of the ANOVA, that the data be independent, was addressed in chapter 3.

After ensuring that all the ANOVA assumptions were met, the following steps were taken to analyze the data:

- Significant higher order interactions were investigated statistically and graphically because these interactions prevent direct interpretation of the main effects. The significant interactions indicated that the effect of one variable was dependent on the presence of other variables (Pedhazur, 1982, p.350).
- The a priori hypotheses originally formulated as linear contrasts of main effects were investigated with pair-wise multiple comparisons. This was done because the weights in the original contrasts were not meaningful given the significant interactions (Pedhazur, 1982, p.362).
- Post hoc multiple comparisons were conducted on all the performance measures to identify any apparent patterns.

Before beginning the analysis, technical considerations in the statistical calculations will be explained.

5.1.2 OVERVIEW OF STATISTICAL CALCULATIONS

This section lists the statistical tests performed on the data. In all cases unless stated otherwise, the confidence level (α) used in conducting the tests was

0.05 and the test

er. This section

and briefly discuss

- Th
  dat
  640
- The
  (SA
- The
  wit
  154
- The
  usi
  PR
- Al
  Sy
  m

The Bon

allows the fami

α is divided by

for the individ

0.05 and the tests were run on a Hitachi Data Systems model ASEX-80 computer. This section also discusses the software packages used to conduct the tests and briefly discusses any special concerns about particular tests.

- The ANOVA analysis and Bonferroni multiple comparisons were done using SAS version 6.06, Proc GLM (SAS/STAT, 1988, p.549-640).
- The Friedman test was done with SAS version 6.06, Proc Rank (SAS Institute, 1988, pp.293-299).
- The Kruskal-Wallis test was done using the ranked data and run with the SAS procedure PROC ANOVA (SAS/STAT, 1988, pp. 125-154).
- The Shapiro-Wilk W statistic and the kurtosis values were obtained using SAS version 6.06, Proc Univariate (SAS Institute, 1988, pp.405-420).
- All graphical analysis was done using the CPLOT procedure in Sygraph version 1.0 (Wilkinson, 1987, pp.231-303). It was run a micro-computer.

The Bonferroni multiple comparisons procedure was used because it allows the family-wise type 1 error rate to be controlled at $\alpha = 0.05$. To do this, α is divided by the total number of comparisons being made to determine the α for the individual comparison (Kirk, 1982, pp.106-109).

As explain

to reduce varian

set of random nu

treatment. This

to treat the comm

design.

The comm

operations per jo

fourth random nu

new jobs arriving

The three

variable in the A

error variability

5.2 A

The mea

effects, full fact

streams. This A

randomly and

and Kutner, 19

Table 5-15.

As explained in chapter three, common random number streams were used to reduce variance. To ensure the independence required by ANOVA, a separate set of random number was used for the fourth random number stream in each treatment. This technique, recommended by Mihram (1974), allows the analyst to treat the common random number streams as a block in the experimental design.

The common random number streams were the streams for the number of operations per job, the processing time per operation and the job routings. The fourth random number stream used in the experiment determined the number of new jobs arriving at the system each week.

The three common random number streams were treated as a blocking variable in the ANOVA. If the blocking variable is significant, the experimental error variability is reduced (Neter, Wasserman and Kutner, 1985, p.902).

5.2 ANALYSIS OF FIT OF ANOVA MODEL TO COST DATA

The mean total cost per week performance data was fitted to a fixed effects, full factorial ANOVA model blocked on the common random number streams. This ANOVA model was used because the factor levels were not chosen randomly and because all interactions could be significant (Neter, Wasserman and Kutner, 1985, p.682). The ANOVA table for the cost data summary is in Table 5-15.

Before in

examined to de

variance and no

The mean

different ways.

- Firs
  a p
  whi
  lack
- Sec
  me
  the
  an
  the
  Will
- Thi
  kur
  the
  kur
  trea

Before interpreting the data in the ANOVA table, the residuals were examined to determine whether the data met the assumptions of homogeneous variance and normality.

5.2.1 CHECK RESIDUALS FOR NORMALITY

The mean total cost data was not normal. This was established in three different ways.

- First a normal probability plot was examined for each cell. This is a plot of the residuals against their expected values. The plots, which were not reproduced here, were not linear. This indicates lack of normality (Neter, Wasserman and Kutner, 1985, p.118).
- Second the Shapiro-Wilk W statistic was calculated for each treatment. The Shapiro-Wilk W is the ratio of the estimated variance to the corrected sum of squares variance. It is distributed between 0 and 1 and small values support rejection of the null hypothesis that the sample is normal (SAS Institute, 1988, p.413). The Shapiro-Wilk W values for the data are shown in Table 5-16.
- Third, the kurtosis value for each treatment was calculated. The kurtosis measures the relative likelihood of extreme values being in the tail of the distribution. A normal distribution would have a kurtosis of 0 (Lewis and Orav, 1989, p.113). The kurtosis for each treatment is shown in Table 5-16.

In Table

with an '*'. Of

treatments also

data is not norm

The hom

Hartley test exa

variance. As the

ratio approaches

that the data be

Neter, W

transformations

data which did

All three trans

- P
- S
- T

In Table 5-16 the Shapiro-Wilk W's, below the critical value, are marked with an '*'. Of the 72 treatments, 42 treatments are not normal. These 42 treatments also had a large kurtosis value providing additional evidence that the data is not normal.

5.2.3 CHECK RESIDUALS FOR HOMOGENEITY

The homogeneity of variance is checked using the Hartley test. The Hartley test examines the ratio of the maximum variance to the minimum variance. As the maximum and minimum variances approach each other, their ratio approaches 1. The Hartley test was not conducted because the test requires that the data be normal (Neter, Wassermann and Kutner 1985, p.622).

5.2.4 SELECT APPROPRIATE TRANSFORMATION

Neter, Wasserman and Kutner (1985, pp.615-617) suggested several transformations (log, square root and reciprocal) as being the most useful for data which did not meet the ANOVA assumptions of normality and homogeneity. All three transformation were tested using the following procedure:

- First the data was transformed.
- Second, the transformed data was fitted to the ANOVA model.
- Third, the residuals were then examined to see if they were normal.

TABLE 5-15
ANOVA TABLE: DEPENDENT VARIABLE IS COST

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
	120	1045413.0507	17914.1944	163.53	0.0001

TABLE 5-15
ANOVA TABLE: DEPENDENT VARIABLE IS COST

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	3045413.0507	17914.1944	163.53	0.0001
ERROR	7029	770027.2168	109.5500		
CORRECTED TOTAL	7199	3815440.2676			

R-SQUARE	C.V.	ROOT MSE	GRAND MEAN
0.798181	16.57895	10.4666154	63.13194

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
ORR RULE	3	292044.7541	97348.2513	888.62	0.0001
WHEN ASSIGNMENT	2	8265.1635	4132.5818	37.72	0.0001
DISPATCH RULE	2	12491.8260	6245.9130	57.01	0.0001
PLANNING SYSTEM	1	2128614.9857	2128614.9857	19430.53	0.0001
BLOCKING FACTOR	99	334705.5498	3380.8641	30.86	0.0001
ORR*WHEN	6	6819.0003	1136.5000	10.37	0.0001
ORR*DISPATCH	6	15113.2628	2518.8771	22.99	0.0001
ORR*PLANNING	3	217307.0064	72435.6688	661.21	0.0001
WHEN*DISPATCH	4	272.7242	68.1811	0.62	0.6465
WHEN*PLAN	2	294.0212	147.0106	1.34	0.2614
DI3PATCH*PLAN	2	7944.0863	3972.0431	36.26	0.0001
ORR*WHEN*DISPATCH	12	1085.4821	90.4568	0.83	0.6240
ORR*WHEN*PLAN	6	1052.0693	175.3449	1.60	0.1426
ORR*DISPATCH*PLAN	6	13650.4406	2275.0734	20.77	0.0001
WHEN*DISPATCH*PLAN	4	1723.2873	430.8218	3.93	0.0034
ORR*WHEN*DISPATCH*PLAN	12	4019.3908	335.7825	3.07	0.0002

TABLE 5-16
SHAPIRO-WILKS W AND THE (KURTOSIS) FOR COST

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	.97252	.97573	.98374	.95503*	.87827*	.86013*	.90309*	.89523*
		(-.263)	(-.499)	(0.541)	(-.429)	(7.865)	(9.794)	(5.598)	(2.317)
	MODD	.96391*	.97299	.97906	.98699	.86789*	.85808*	.89238*	.86883*
		(-.160)	(0.514)	(-.196)	(0.268)	(8.182)	(9.717)	(5.336)	(5.763)
	CMDD	.98522	.98728	.97321	.96335*	.86961*	.86740*	.89005*	.89658*
		(0.730)	(0.103)	(-.363)	(0.743)	(8.693)	(8.585)	(6.114)	(4.663)
JOB	FCFS	.97799	.98188	.97822	.97003	.87269*	.84657*	.90269*	.89112*
		(-.065)	(0.007)	(-.151)	(-.256)	(7.791)	(10.89)	(5.058)	(4.274)
	MODD	.97389	.86844*	.98944	.35220*	.89592*	.87403*	.87295*	.86902*
		(1.434)	(5.564)	(-.058)	(11.48)	(7.701)	(7.964)	(6.205)	(5.567)
	CMDD	.96536	.98262	.97774	.97259	.87497*	.87823*	.88669*	.88079*
		(0.722)	(0.096)	(-.173)	(0.471)	(8.033)	(7.994)	(5.593)	(7.218)
CLR	FCFS	.97774	.97050	.97356	.98077	.87113*	.84569*	.90091*	.89408*
		(1.549)	(0.064)	(-.485)	(0.231)	(8.794)	(10.61)	(5.157)	(4.274)
	MODD	.98291	.97049	.95956*	.97809	.92949*	.86174*	.88328*	.84653*
		(0.201)	(-.348)	(0.077)	(0.353)	(3.585)	(8.776)	(5.824)	(8.118)
	CMDD	.98587	.98709	.97903	.98522	.88131*	.85988*	.84669*	.86267*
		(-.074)	(.3570)	(-.130)	(-.083)	(7.622)	(9.120)	(9.663)	(6.866)

* REJECT HYPOTHESIS THAT THIS CELL IS NORMAL

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION

- Fourth, if the residuals were normal they were examined to see if the variances were homogeneous.

The Shapiro-Wilks W statistic was compared for each transformation. The reciprocal transformation of the cost data had the fewest number of cells which were not normal. The variance of the reciprocal transformation were then tested for homogeneity and as discussed below were found to be homogeneous. For these reasons, the reciprocal transformation of the cost data was selected for further analysis. The means and standard deviations for the reciprocal cost data are shown in Table 5-17.

5.2.4.1 CHECK RESIDUALS OF RECIPROCAL DATA FOR NORMALITY

The normality of the reciprocal data was determined using a three step procedure. First a normal probability plot of the data for each treatment was examined. These plots, which were not reproduced here, appeared to be linear, suggesting that the data was normal (Neter, Wasserman and Kutner, 1985, p. 118).

Second, the Shapiro-Wilk W statistic was calculated for the data for each treatment. The Shapiro-Wilk W values for the residuals of the reciprocal cost data are shown in Table 5-18 The reciprocal of the total cost had only four cells with a W value low enough that the null hypothesis that they were normal was rejected at $\alpha = 0.05$.

Third, the kurtosis values for each treatment were calculated and shown in Table 5-18 Fifteen of the treatments had a negative kurtosis, but the number

of cells with a kurtosis greater than the absolute value of 1.0 was reduced to 14 cells. Both the Shapiro Wilk W and the kurtosis values indicated that the data were largely normal.

5.2.4.2 CHECK RECIPROCAL RESIDUALS FOR HETEROSCEDASCITY

Because the transformed data was normal, the Hartley test was used to inspect for heteroscedascity. The null hypothesis was that the variances were equal. The following procedure was used to conduct the Hartley test.

- First the treatments were arranged in ascending order of their variances.
- Second, starting with the first treatment, the ratio of the treatment variances to each other were calculated. These ratios are shown in Appendix E. All ratios were calculated by placing the largest variance in the numerator and the smallest in the denominator.
- Third, the ratios were compared to the critical value for the test. For $\alpha = 0.05$, this was 2.36.

The treatments, arranged in ascending order of their variance, are shown in Figure 5-1. Those treatments which did not have statistically different variances are underlined.

There were nineteen treatments with variances significantly different from the other fifty-three treatments. Fourteen of these nineteen treatments used the SMOOTH planning system. These fourteen included all the treatments which used the SMOOTH planning system and the CMS ORR rule. Treatments using

TABLE 5-17
MEAN AND STANDARD DEVIATION OF RECIPROCAL OF COST

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM							
		SMOOTH				UNCONTROLLED			
		ORDER REVIEW/RELEASE RULE				ORDER REVIEW/RELEASE RULE			
		IMM	MIL	CMS	MLC	IMM	MIL	CMS	MLC
QUE	FCFS	.02272 (.0017)	.02519 (.0014)	.01889 (.0007)	.02440 (.0011)	.01031 (.0023)	.01335 (.0027)	.013521 (.0018)	.01551 (.0026)
	MODD	.02373 (.0015)	.02671 (.0015)	.01896 (.0007)	.02387 (.0016)	.01222 (.0026)	.01397 (.0029)	.01359 (.0018)	.01581 (.0026)
	CMDD	.02131 (.0014)	.02437 (.0014)	.01884 (.0007)	.02785 (.0013)	.01098 (.0022)	.01287 (.0026)	.01354 (.0018)	.01398 (.0023)
JOB	FCFS	.02125 (.0015)	.02217 (.0011)	.01849 (.0007)	.02589 (.0012)	.00984 (.0022)	.01249 (00236)	.01332 (.0018)	.01533 (.0025)
	MODD	.02149 (.0014)	.02278 (.0017)	.01859 (.0007)	.02401 (.0012)	.01159 (.0028)	.01275 (.0025)	.01345 (.0017)	.01612 (.0028)
	CMDD	.02007 (00130)	.02143 (.0011)	.01854 (.0007)	.02313 (.0015)	.01073 (.0021)	.01207 (.0022)	.01334 (.0018)	.01512 (.0026)
CLR	FCFS	.02185 (.0015)	.02280 (.0011)	.01854 (.0006)	.02525 (.0013)	.00998 (.0022)	.01269 (.0024)	.01339 (.0017)	.01505 (.0023)
	MODD	.02179 (.0014)	.02376 (.0012)	.01867 (.0007)	.02404 (.0012)	.01145 (.0021)	.01302 (.0026)	.01347 (.0017)	.01596 (.0027)
	CMDD	.02056 (.0013)	.02201 (.0012)	.01861 (.0007)	.02387 (.0011)	.01079 (.0021)	.01222 (.0023)	.01325 (.0017)	.01598 (.0027)

* CELL ENTRY ARRANGED AS MEAN (STANDARD DEVIATION)
QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE
IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

TABLE 5-18
SHAPIRO-WILKS W AND THE KURTOSIS FOR RECIPROCAL OF COST*

"WHEN" Labor Assignment	Job Dispatch Rule	PLANNING SYSTEM/JOB SELECTION MECHANISM: SMOOTH — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC	UNCONTROLLED — ORDER REVIEW/RELEASE RULE: IMM	MIL	CMS	MLC
QUE	FCFS	.97051 (-.333)	.97265 (0.174)	.97658 (-.285)	.97915 (-.187)	.98267 (.639)	.98652 (.967)	.98820 (0.975)	.98374 (0.244)
	MODD	.97925 (-.377)	.98825 (-.125)	.98063 (-.211)	.98663 (0.266)	.98387 (0.565)	.98941 (.919)	.97949 (1.469)	.98692 (0.990)
	CMDD	.99122 (0.387)	0.9869 (0.053)	.98086 (-.076)	.97357 (-.291)	.98287 (0.857)	.98987 (1.261)	.97949 (1.931)	.98226 (0.425)
JOB	FCFS	0.9754 (0.568)	0.9780 (-.423)	.98993 (.671)	.98727 (-.106)	.99159 (0.732)	.97715 (1.293)	.98105 (0.779)	.98042 (0.249)
	MODD	.96965 (-.125)	.90874* (3.628)	.97155 (-.581)	.97609 (-.373)	.76991* (24.41)	.97611 (.711)	.96822 (1.653)	.97946 (0.327)
	CMDD	0.9691 (-.575)	.98233 (0.121)	.95523* (-.803)	.98661 (0.222)	.97975 (1.152)	.98364 (1.136)	.97806 (1.561)	.97008 (0.212)
CLR	FCFS	.97913 (-.343)	.98421 (0.286)	.97782 (0.259)	.97217 (-.122)	.98754 (1.002)	.98093 (1.184)	.97568 (0.931)	.98487 (0.801)
	MODD	.98445 (0.481)	.98038 (-.400)	.98406 (0.377)	.98121 (0.642)	.98347 (.856)	.98540 (0.935)	.97818 (1.41)	.97967 (0.419)
	CMDD	.97473 (-.398)	.98916 (1.084)	.97819 (-.162)	.98704 (0.125)	.98816 (1.035)	.98549 (0.939)	.95221* (2.386)	.98583 (0.778)

* CELL ENTRY ARRANGED AS SHAPIRO WILKS W AND (KURTOSIS)

QUE = RELEASE LABOR WHEN QUEUE EMPTY
JOB = RELEASE LABOR WHEN JOB FINISHED
CLR = USE JOB IF MACHINE CRITICAL OTHERWISE QUE
FCFS = FIRST COME FIRST SERVED
MODD = MODIFIED OPERATION DUE DATE

IMM = IMMEDIATE RELEASE OF JOBS
MIL = MODIFIED INFINITE LOADING
CMS = CRITICAL MACHINE SELECTION
MLC = MODIFIED LOAD CONVERSION
CMDD = CRITICAL MACHINE DUE DATE

the CMS ORR rule had the smallest variances. Three treatments using MIL and two treatments using MLC ORR rules were also included. The remaining five treatments had the largest variances. All of these five treatments used the UNCONTROLLED planning system.

Neter et al. (1985, pp.662-623) point out that the F test is robust to unequal variances and a slight departure from normality when two conditions are met. These conditions, both of which were met here, are that a fixed effects ANOVA model is used, and that the sample sizes for each treatment are equal. Given these conditions, they suggest that a smaller alpha value be used in the Hartley test. For an alpha = 0.01, the critical value is 2.77. At this alpha, sixty-ne treatments have equal variances as shown in Figure 5-2. Ten of the cells not nderlined in Figure 5-2 had the smallest variances of cells in the experiment. e eleventh treatment, 2122 (MIL/QUE/MODD/UNCONTROLLED), had the gest variance in the experiment.

While the ANOVA is robust to some heteroscedascity (Scheffe, 1959), e is still an issue of whether these eleven treatments without homogeneous nces invalidates the use of ANOVA. The issue was approached in two ways.

- First the effect on the ANOVA model of violating the assumption was considered.
- Second, the data was examined using non parametric statistics because homogeneity of variance is not a necessary assumption.

FIGURE 5-1
HOMOGENEITY OF VARIANCE: HARTLEY TEST

RECIPROCAL COST DATA: $\alpha = 0.05$: CELLS ARRANGED IN ASCENDING VARIANCE*

3311 3331 3211 3221 3111 3231 3131 3321 3121 2211 2231 4331 2311 4111 4324 2321 4221

2331 4211 4311 1331 1231 4131 1321 1221 1131 2131 2111 1211 1121 4231 1311 2121 4121

2221 1111 3222 3322 3332 3312 3232 3212 3132 3122 3112 1332 1232 1322 1212 2232 1312

4132 1132 2332 4312 1112 2212 2312 2222 4212 2132 1122 4122 2322 4112 4232 4332 2112

4322 1222 4222 2122

INDICATES THAT CELLS DO NOT HAVE SIGNIFICANTLY DIFFERENT VARIANCES

* ALL THE CELL RATIOS ARE SHOWN IN APPENDIX E

LEGEND: DIGIT #1 IS ORR RULE, DIGIT #2 IS WHEN RULE, DIGIT #3 IS DISPATCHING, DIGIT #4 IS THE PLANNING SYSTEM
ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
WHEN RULE (1=QUE, 2=JOB, 3=CLR)
DISPATCHING RULE (1=FCFS, 2=MODD, 3=CMDD)
PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

RECIPROCAL COST DATA: α = 0.01: CELLS ARRANGED IN ASCENDING VARIANCE*

3311 3331 3211 3221 3111 3231 3131 3321 3121 2211 2231 4331 2311 4111 4324 2321 4221

2331 4211 4311 1331 1231 4131 1321 1221 1131 2131 2111 1211 1121 4231 1311 2121 4121

2221 1111 3222 3322 3332 3312 3232 3212 3132 3122 3112 1332 1232 1322 1212 2232 1312

4132 1132 2332 4312 1112 2212 2312 2222 4212 2132 1122 4122 23224112 4232 4332 2112

4322 1222 4222 2122

INDICATES THAT CELLS DO NOT HAVE SIGNIFICANTLY DIFFERENT VARIANCES

* ALL THE CELL RATIOS ARE SHOWN IN TABLE A-1 IN APPENDIX A

LEGEND: DIGIT #1 IS ORR RULE, DIGIT #2 IS WHEN RULE, DIGIT #3 IS DISPATCHING, DIGIT #4 IS THE PLANNING SYSTEM
ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
WHEN RULE (1=QUE, 2=JOB, 3=CLR)
DISPATCHING RULE (1=FCFS, 2=MODD, 3=CMDD)
PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

These results were compared to the results from the parametric statistics.

5.2.4.2.1 Effect of Violating ANOVA Assumptions

Because unequal variances could potentially make the F test slightly more significant (Neter, Wasserman and Kutner, 1985, p.624), the pair-wise multiple comparisons shown in Figure 5-3 were run. These multiple comparisons showed that the eleven treatments whose variances were different from the other variances were not among the top performing treatments, so the increased significance did not seem to bias the results. This conclusion was tested further by comparing treatment 4231 (MLC/JOB/CMDD/SMOOTH) to treatment 2211 (MIL/JOB/FCFS/SMOOTH) using a t-test.

Treatment 2211 is the lowest cost treatment among those eleven treatments with the smallest variance. Notice in Figure 5-3 that treatment 4231 has the smallest significant difference from treatment 2211.

The Bonferroni pair-wise comparison uses the pooled variance in its test. This means that a significant difference could be due to the sub-optimality of a treatment or to the heterogeneity of the variances. A t-test of the differences between the treatments uses the actual variances of each mean. If the t-test verifies a significant difference between the means there is support for the hypothesis that the heterogeneity of the variance does not bias the results of the Bonferroni comparisons.

The null and alternative hypotheses of the t-test are:

H_o: $\mu_{4231} = \mu_{2211}$

H_A

The varia

s^2

s^2

The 95%

(0.

0.0

Beca

hypothesis

of ANOVA c

5.2.4.2.2 Us

The d

264-274). T

blocks were

was used to s

The AN

model's F val

parametric AN

other. It supp

results of the

The Kr

compare the

Figure 5-4. C

H_A: $\mu_{4231} > \mu_{2211}$

The variance of the difference between the two means is:

$$s^2_{4231-2211} = \{s^2_{4231}/n_{4231} + s^2_{2211}/n_{2211}\}$$

$$s^2_{4231-2211} = 0.000186$$

The 95% confidence interval for $\mu_{4231} - \mu_{2211}$ is:

$(0.02313-0.02217) \pm z_{.95}(0.000186) = [0.000654,-0.001266]$.

Because the interval does not include 0, H_o is rejected. This supports the hypothesis that the heteroscedascity of the treatments does not invalidate the use of ANOVA or Bonferroni multiple comparisons.

5.2.4.2.2 Use of Parametric and Non parametric Statistics

The data was also analyzed using the Friedman test (Conover, 1971, pp. 64-274). This test is a non parametric two way analysis of variance. The ocks were the common random number streams. There were 72 treatments. It is used to see if it supported the results from the ANOVA.

The ANOVA table for the Friedman test is shown in Table 5-19 The del's F value is 595.58 which is significant at 0.0001. This agrees with the ametric ANOVA result that the treatments are significantly different from each r. It supports the hypothesis that the heteroscedascity does not bias the ts of the parametric statistics.

The Kruskal-Wallis test (Conover, 1971, pp. 256-263) was used to are the treatments to each other. The results of this test are shown in 5-4. Comparing the Bonferroni multiple comparisons in Figure 5-3 and

the Kruska

best and w

metric tes

ments in t

FCFS SMC

(MIL QUE

However in

2111. The

date the use

Becau

ric statistics,

The an

1.

2.

3.

Each step will

the Kruskal-Wallis comparisons in Figure 5-4, it is clear that the rankings of the best and worst treatments are the same for both the parametric and non parametric tests. However, there is some slight change in the rankings of the treatments in the middle. For example in Figure 5-3, treatment 4311 (MLC/CLR/-FCFS/SMOOTH) is the fourth lowest cost treatment. It precedes treatment 2111 (MIL/QUE/FCFS/SMOOTH), while in Figure 5-4, treatment 2111 precedes 4311. However in both figures there is no significant difference between 4311 and 2111. The conclusion is that the heterogeneity is not severe enough to invalidate the use of parametric statistics.

Because the parametric statistics have more power than the non parametric statistics, they are used in the remainder of this research.

5.3 ANALYSIS OF RECIPROCAL COST DATA

The analysis of the reciprocal cost data was divided into three steps:

1. Fitting the reciprocal cost data to the full factorial fixed effects ANOVA model;
2. Conducting pair-wise multiple comparisons of reciprocal cost data;
3. Graphically and statistically examine the significant interactions identified by the ANOVA model.

ch step will be discussed in sequence below.

The

The overall

probability

Table 5-20.

effects, the b

significant.

generalize ab

ANOVA table.

The pla

data. The OR

"when" labor a

portions of the

The AN

that because o

considered sep

The rest

some possibilit

are not valid.

do not have th

means of the r

untransformed

5.3.1 FIT OF RECIPROCAL DATA TO ANOVA MODEL

The ANOVA table for the reciprocal of the cost is shown in Table 5-20 The overall ANOVA model was highly significant with a F value of 525.06 and a probability of 0.0001. The R^2 value was 92.70%. In examining the results in Table 5-20, it is important to note that all of the effects, including the main effects, the block effects, and more importantly all the interaction effects were significant. The significant higher order interactions made it impossible to generalize about the main effects. However there was some information in the ANOVA table.

The planning system accounts for the largest amount of variance in the data. The ORR rule accounts for another large portion of the variance. The "when" labor assignment rule and finally the job dispatch rule account for smaller portions of the variance.

The ANOVA table indicates that the main effects differ in importance and it because of the high order interactions the individual treatments must be considered separately.

The rest of the analysis was conducted on the reciprocal data. There is the possibility that the Bonferroni multiple comparisons of the reciprocal data not valid. Invalid comparisons can occur if the treatments being compared not have the same variance. This happens because the interval between the s of the reciprocal data is different than the intervals between the original nsformed data and because the Bonferroni test uses a pooled variance. So,

the interv

the interv

tion of th

sons in Fi

is probabl

The

analyze the

order of cos

were joined

wo major gr

entirely of tre

tally significa

group of trea

planning syst

In the

SMOOTH) wa

experiment.

planning syst

CMS ORR rule

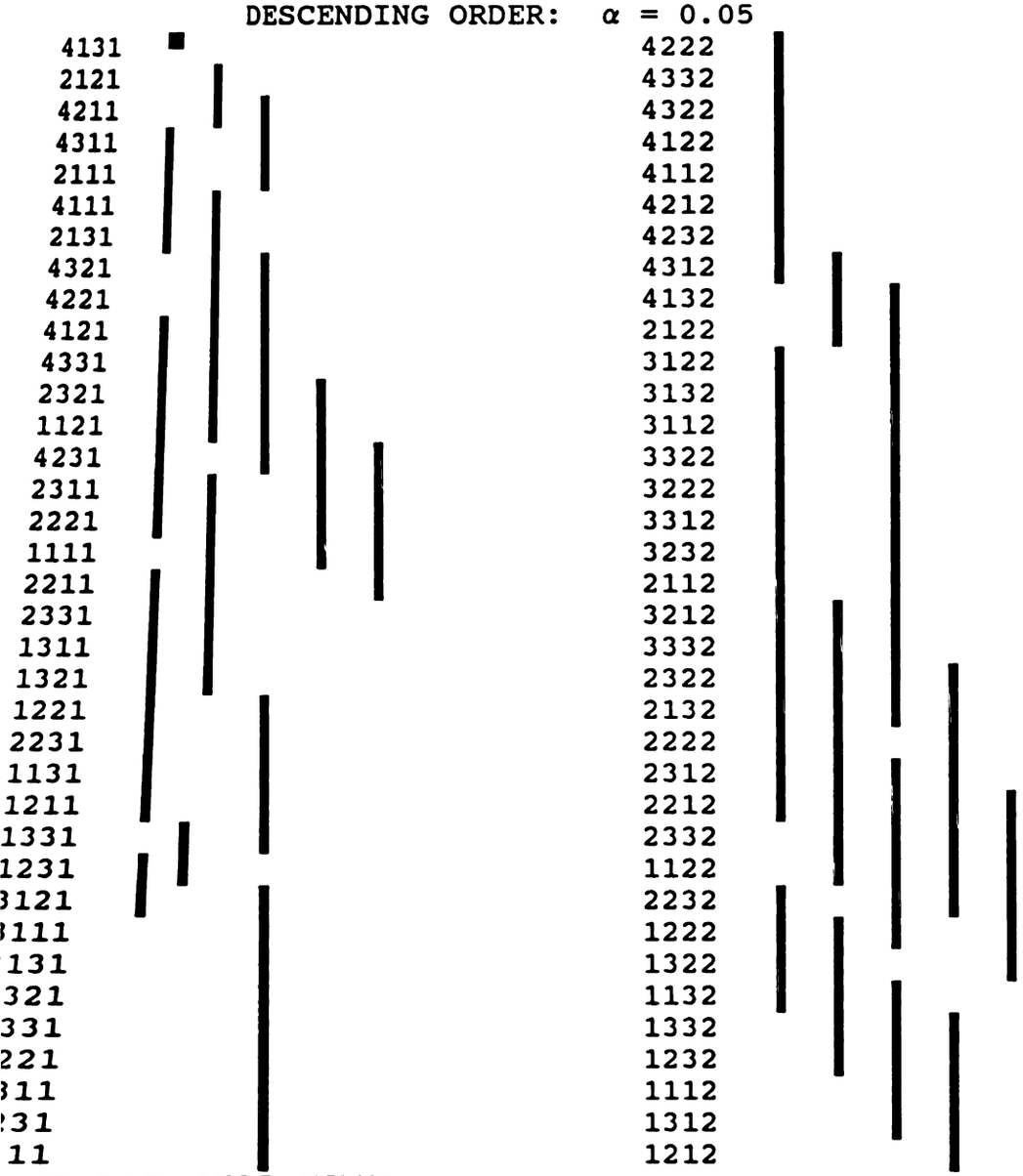
the interval between the points with the original data may test significant when the intervals with the reciprocal data do not test significant. However, examination of the Bonferroni comparisons in Figure 5-3 and the Kruskal-Wallis comparisons in Figure 5-4, shows that the rankings by both methods are the same, so it is probable that the comparisons in Figure 5-3 are valid.

5.3.2 PAIR-WISE MULTIPLE COMPARISONS

The pair-wise multiple comparisons shown in Figure 5-3 were used to analyze the reciprocal cost data. The treatments were arranged in ascending order of cost. Treatments which were not statistically different from each other were joined by a solid line on their right hand side. It was clear that there were two major groupings of treatments. The first group of treatments consisted entirely of treatments using the SMOOTH planning system. These had a statistically significant lower cost than the second group of treatments. The second group of treatments consisted entirely of treatments using the UNCONTROLLED planning system.

In the SMOOTH group, the treatment labeled 4131 (MLC/QUE/CMDD/-SMOOTH) was significantly lower cost than all the other treatments in the experiment. The other treatments using the MLC ORR rule and the SMOOTH planning system were also low cost performers. However, treatments using the MLC ORR rule were the worst performers in the SMOOTH group.

FIGURE 5-3
BONFERRONI COMPARISONS OF RECIPROCAL COST[1]



END FOR CELL MEANS
IT 1, DIGIT 2, DIGIT 3, DIGIT 4
IT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
IT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
IT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
IT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

ALL T TESTS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM
EDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-4
KRUSKAL-WALLIS TEST OF RECIPROCAL COST

GEND FOR CELL MEANS
GIT 1, DIGIT 2, DIGIT 3, DIGIT 4
IT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
IT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
IT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
IT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

TABLE 5-12
ANOVA TABLE: FRIEDMAN TEST

ALPHA = 0.05

SOURCE	D.F.	SUM OF SQUARES	[illegible]

TABLE 5-19

ANOVA TABLE: FRIEDMAN TEST

ALPHA = 0.05

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	2907924.2799	17105.43694	595.58	0.0001
ERROR	7029	201875.7200	28.72040		
CORRECTED TOTAL	7199	3109800.0000			

R-SQUARE	C.V.	ROOT MSE	GRAND MEAN
0.935084	14.68258	5.3591421	36.50000

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
TREATMENT	71	2907924.2799	40956.6800	1462.05	0.0001
SAMPLE NUMBER	99	0	0	0	1.0000

TABLE 5-20
ANOVA TABLE: RECIPROCAL COST DATA

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE
MODEL	170	0.18870501	
ERROR			

TABLE 5-20
ANOVA TABLE: RECIPROCAL COST DATA

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	0.18879503	0.00111056	525.06	0.0001
ERROR	7029	0.01486714	0.00000212		
CORRECTED TOTAL	7199	0.20366217			

R-SQUARE	C.V.	ROOT MSE	GRAND MEAN
0.927001	8.248745	0.0014534	0.01763109

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
ORR RULE	3	0.01893102	0.00631034	2983.45	0.0001
WHEN ASSIGNMENT	2	0.00117033	0.00058516	276.66	0.0001
DISPATCH RULE	2	0.00055202	0.00027601	130.49	0.0001
PLANNING SYSTEM	1	0.14536919	0.14536919	68728.75	0.0001
BLOCKING FACTOR	99	0.00946071	0.00009556	45.18	0.0001
ORR*WHEN	6	0.00075066	0.00012511	59.15	0.0001
ORR*DISPATCH	6	0.00059525	0.00009921	46.90	0.0001
ORR*PLANNING	3	0.00897519	0.00299173	1414.45	0.0001
WHEN*DISPATCH	4	0.00006200	0.00001550	7.33	0.0001
WHEN*PLAN	2	0.00056447	0.00028224	133.44	0.0001
DISPATCH*PLAN	2	0.00014795	0.00007398	34.98	0.0001
ORR*WHEN*DISPATCH	12	0.00035602	0.00002967	14.03	0.0001
ORR*WHEN*PLAN	6	0.00022091	0.00003682	17.41	0.0001
ORR*DISPATCH*PLAN	6	0.00060080	0.00010013	47.34	0.0001
WHEN*DISPATCH*PLAN	4	0.00022719	0.00005680	26.85	0.0001
ORR*WHEN*DISPATCH*PLAN	12	0.00081132	0.00006761	31.97	0.0001

In the UNCONTROLLED group, the treatments using the MLC ORR rule re the nine lowest cost treatments. However treatment 2122 (MIL\QUE\-DD\UNCONTROLLED) tied with two of the MLC rules. There was no tistically significant difference between the treatments using the CMS rule and eatments using the MIL rules.

To continue the analysis further, it was necessary to examine the four way teraction.

5.3.3 EXAMINATION OF THE FOUR WAY INTERACTION

As a first step in analyzing the four way interaction several different graphs of the treatment means (not reproduced in this study) were made. These graphs were examined and it was determined that the graph shown in Figure 5-5 provided the clearest representation of the interaction.

In Figure 5-5, the treatments were divided into the two groups (SMOOTH and UNCONTROLLED) identified using the multiple comparisons in Figure 5-3. Treatments using the SMOOTH planning system are represented with a circle, o. Treatments using the SMOOTH planning system are represented with a triangle, Δ. Treatments using the same planning system are connected with lines. The labels on the X axis show which ORR rule, which "when" labor assignment rule and which job dispatching rule are used in each treatment. The label for any treatment on the graph can be found by taking the label from the X axis and combining it with the appropriate label for the planning system.

For exa

number 1112

label came

represent t

signified th

portion sig

The third 1

dispatching

Figure

depended on t

ning system wa

to interpret the

rules from this

To contr

shown in Figur

interaction. Be

separate graph

Graphs

dispatching ru

5.3.3.1 FOUR

The trea

in Figure 5-6.

For example the first point on Figure 5-5 shown with a ▲ was treatment number 1112 (IMM\QUE\FCFS\UNCONTROLLED). The '111' portion of the label came from the X axis and the '2' portion of the label was added to it to represent the UNCONTROLLED planning system. The first 1 in the '111' portion signified that the treatment used the IMM ORR rule. The second 1 in the '111' portion signified that the treatment used the QUE "when" labor assignment rule. The third 1 in the '111' portion signified that the treatment used the FCFS job dispatching rule.

Figure 5-5 demonstrated graphically that the performance of the ORR rule depended on the planning system being used. It also illustrated that the planning system was the primary determinant of the system cost. But it was difficult to interpret the effect of the "when" labor assignment rules and job dispatching rules from this graph.

To continue the analysis, the appropriate multiple pair-wise comparisons shown in Figure 5-3 were combined with different graphs of the four way interaction. Because of the significant difference between each planning system, separate graphs were prepared for each type of planning system.

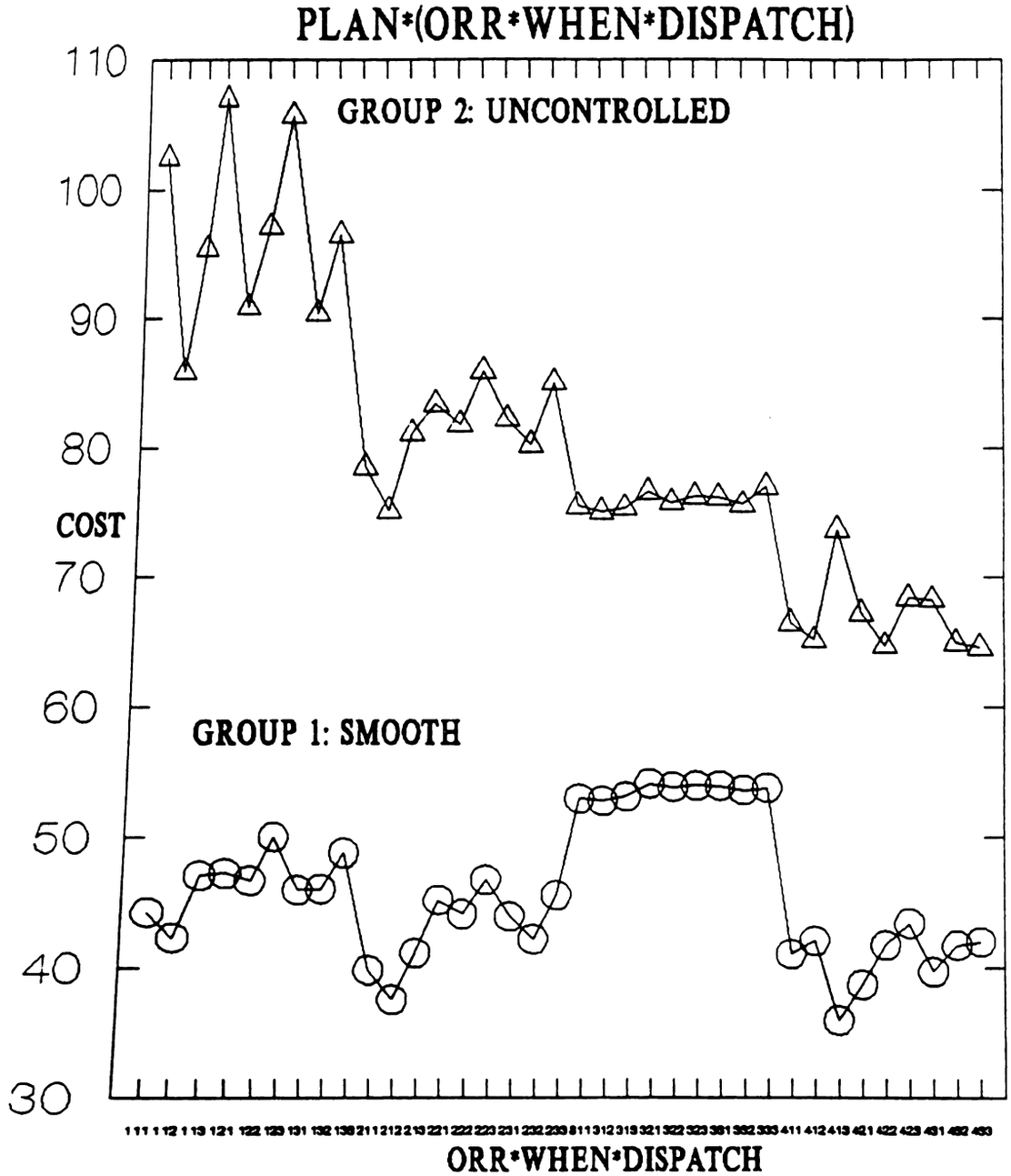
Graphs of the ORR rules, the "when" labor assignment rules and the job dispatching rules are explored separately.

1 FOUR WAY INTERACTION AND ORR RULES

The treatments which used the SMOOTH planning system were graphed in Figure 5-6. These treatments were organized by their ORR rule. The "when"

COST

FIGURE 5-5
FOUR WAY INTERACTION



EACH TREATMENT IS IDENTIFIED BY FOUR DIGITS

DIGIT 1 IS THE ORR RULE: 1=IMM,2=MIL,3=CMS,4=MLC

DIGIT 2 IS 'WHEN' RULE: 1=QUE,2=JOB,3=CLR

DIGIT 3 IS JOB DISPATCH RULE: 1=FCFS, 2=MODD, 3=CMDD

DIGIT 4 IS THE PLANNING SYSTEM: 1=SMOOTH, 2=UNCONTROLLED

labor assign

treatments

column (v

and job d

three of t

the same

treatments

SMOOTH)

Treat

treatment 212

(M.M.QUE, M

ments using th

of "when" labor

of "when" labor

MM ORR rule

Those tr

graphed in Fig.

the "when" labo

a given combin

using the MLC r

significantly lows

labor assignment rule and job dispatching rule were shown on the X axis. Those treatments which were not statistically different from other treatments in their column (which means other treatments using the same "when" labor assignment and job dispatching rules) were shown with a vertical line next to them. All but three of the MLC rules performed better than the other ORR rules combined with the same job dispatching and "when" labor assignment rules. Two of these treatments, 4111 (MLC/QUE/FCFS/SMOOTH) and 4321 (MLC/CLR/MODD/-SMOOTH) tied with the corresponding treatment using the MIL ORR rule.

Treatment 4121 (MLC/QUE/MODD/SMOOTH) performed worse than treatment 2121 (MIL/QUE/MODD/SMOOTH), tying with treatment 1121 (IMM/QUE/MODD/SMOOTH) for the second best rule in the column. Treatments using the CMS ORR rule were the worst performers for all combinations of "when" labor assignment and job dispatching rules. For a given combination of "when" labor assignment and job dispatching rules, the treatments using the IMM ORR rule were always third or tied for second.

Those treatments which used the UNCONTROLLED planning system were graphed in Figure 5-7. These treatments were organized by their ORR rule with the "when" labor assignment and job dispatching rules shown on the X axis. For a given combination of the "when" and job dispatching rules, only one treatment using the MLC rule, 4132 (MLC/QUE/CMDD/UNCONTROLLED), was not significantly lower in cost than its corresponding treatments. Only one treatment

using the CN

significantl

Al

label beg

sponding

5.3.3.2 F

To

treatments

those using

The treatmen

used shown or

In Figur

"when" rules fi

treatments wer

SMOOTH), 2:2

SMOOTH) and

difference in th

Figure 5-9.

The choi

performance for

using the CMS ORR rule, 3232 (CMS/JOB/CMDD/UNCONTROLLED), was significantly different from treatments using the MIL rule.

All the treatments which immediately released jobs to the shop floor (the label begins with the digit 1) gave a significantly higher cost than the corresponding treatments using ORR rules.

5.3.3.2 FOUR WAY INTERACTION AND "WHEN" LABOR ASSIGNMENT

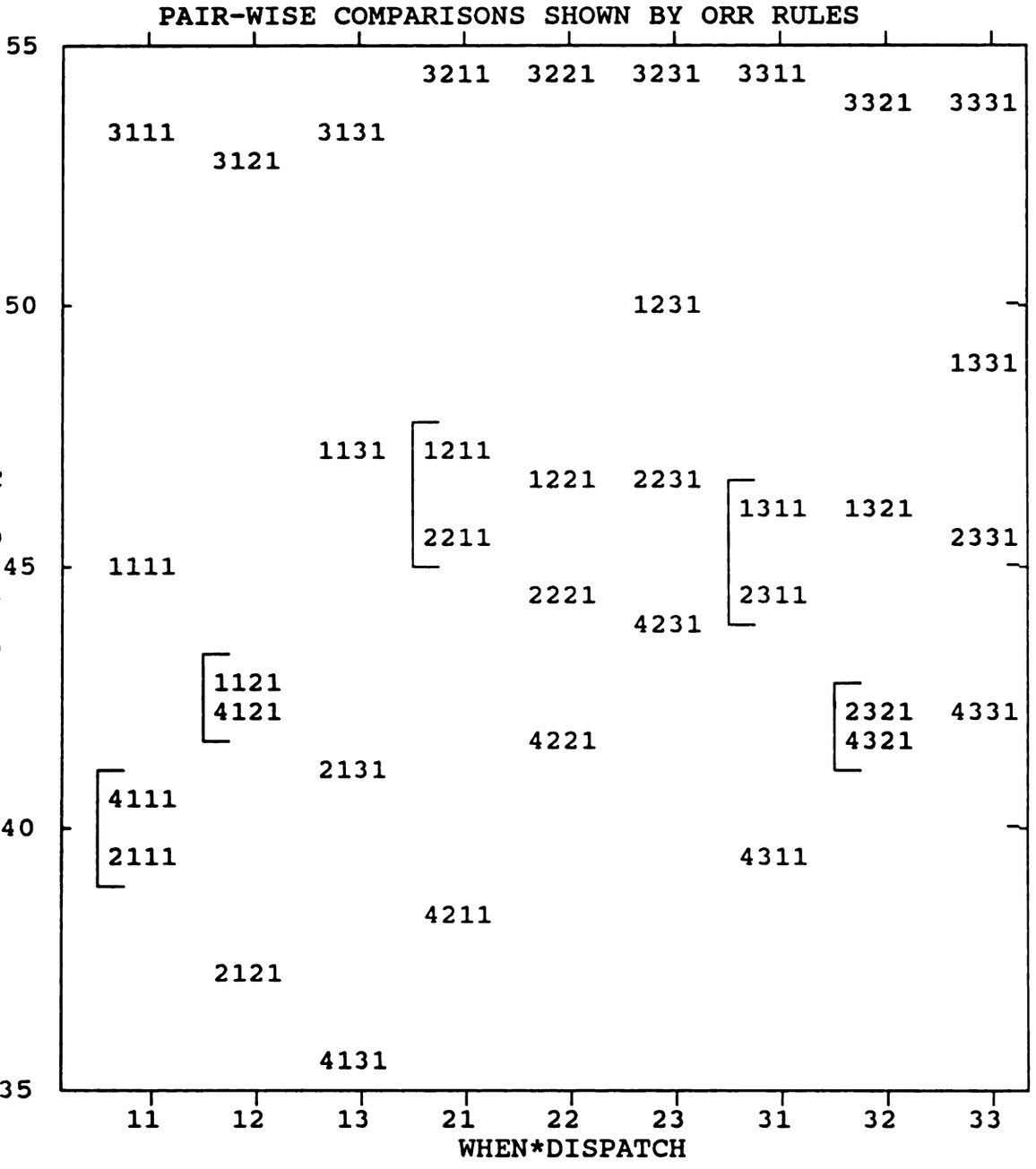
To examine the interaction of the "when" labor assignment rules the treatments using the SMOOTH planning system were graphed in Figure 5-8, and those using the UNCONTROLLED planning system were graphed in Figure 5-9. The treatments were graphed with the combined ORR and job dispatch rules used shown on the X axis.

In Figure 5-8, the QUE "when" rule was significantly better than the other "when" rules five times. No other "when" rule was better than QUE. The treatments were: 1121 (IMM/QUE/MODD/SMOOTH), 2111 (MIL/QUE/FCFS/SMOOTH), 2121 (MIL/QUE/MODD/SMOOTH), 2131 (MIL/QUE/CMDD/SMOOTH) and 4131 (MLC/QUE/CMDD/SMOOTH). There was no significant difference in the performance of the "when" labor assignment rules shown in Figure 5-9.

The choice of the "when" rule appears to have no impact on the system performance for treatments using the UNCONTROLLED planning system.

55
50
45
40
35
11
4111
2111
TREATMENT
DIGIT 2
DIGIT 4
CRR: 1=
DISPATCH
PLAN: 1=S

FIGURE 5-6
FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR: 1=IMM,2=MIL,3=CMS,4=MLC. WHEN: 1=QUE,2=JOB,3=CLR.
DISPATCH: 1=FCFS,2=MODD,3=CMDD
PLAN:1=SMOOTH,2=UNCONTROLLED

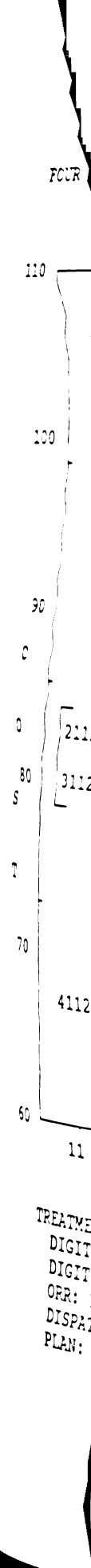
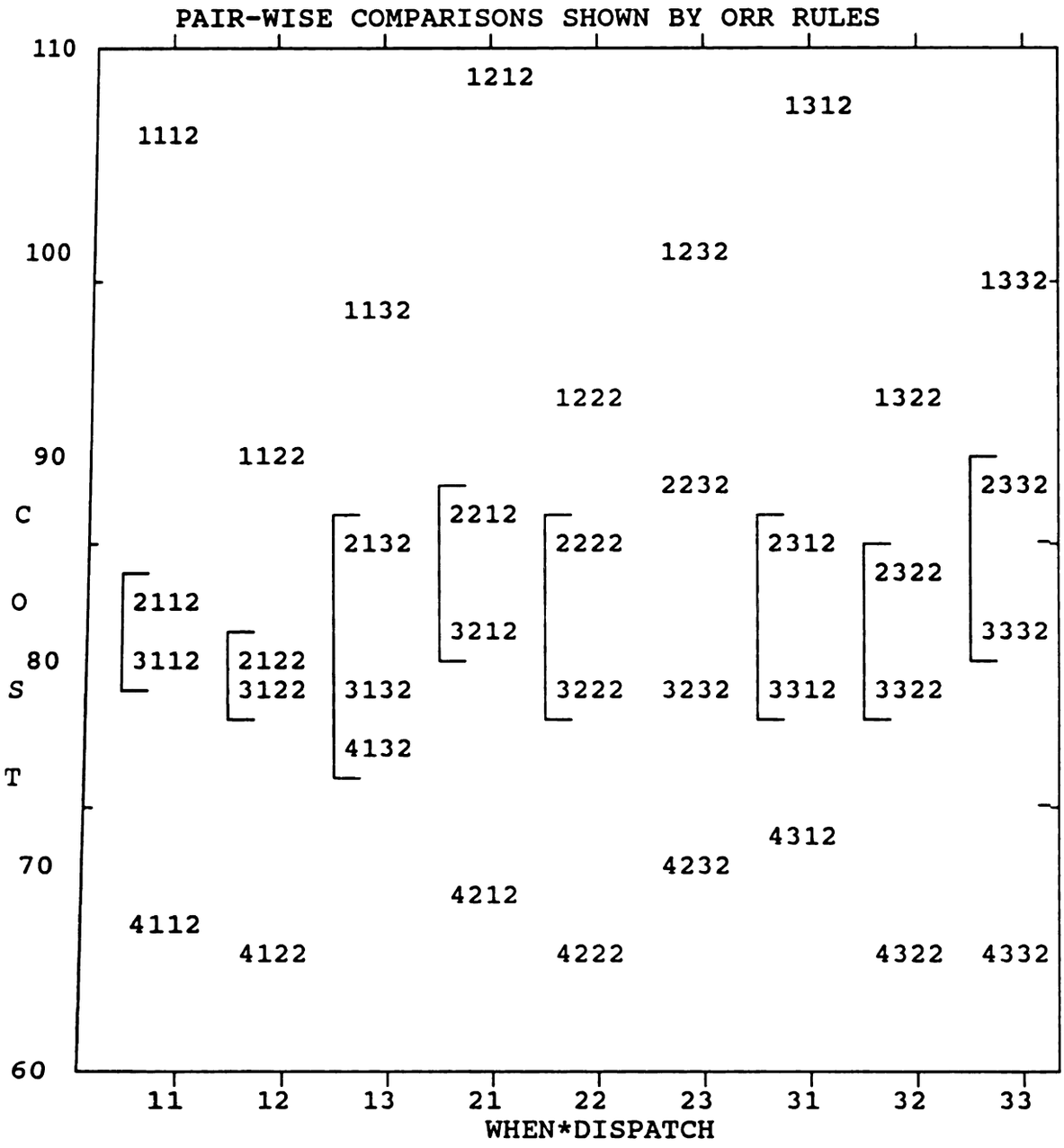

FOUR
110
100
90
80
70
60
11
TREATME
DIGIT
DIGIT
ORR:
DISPA
PLAN:

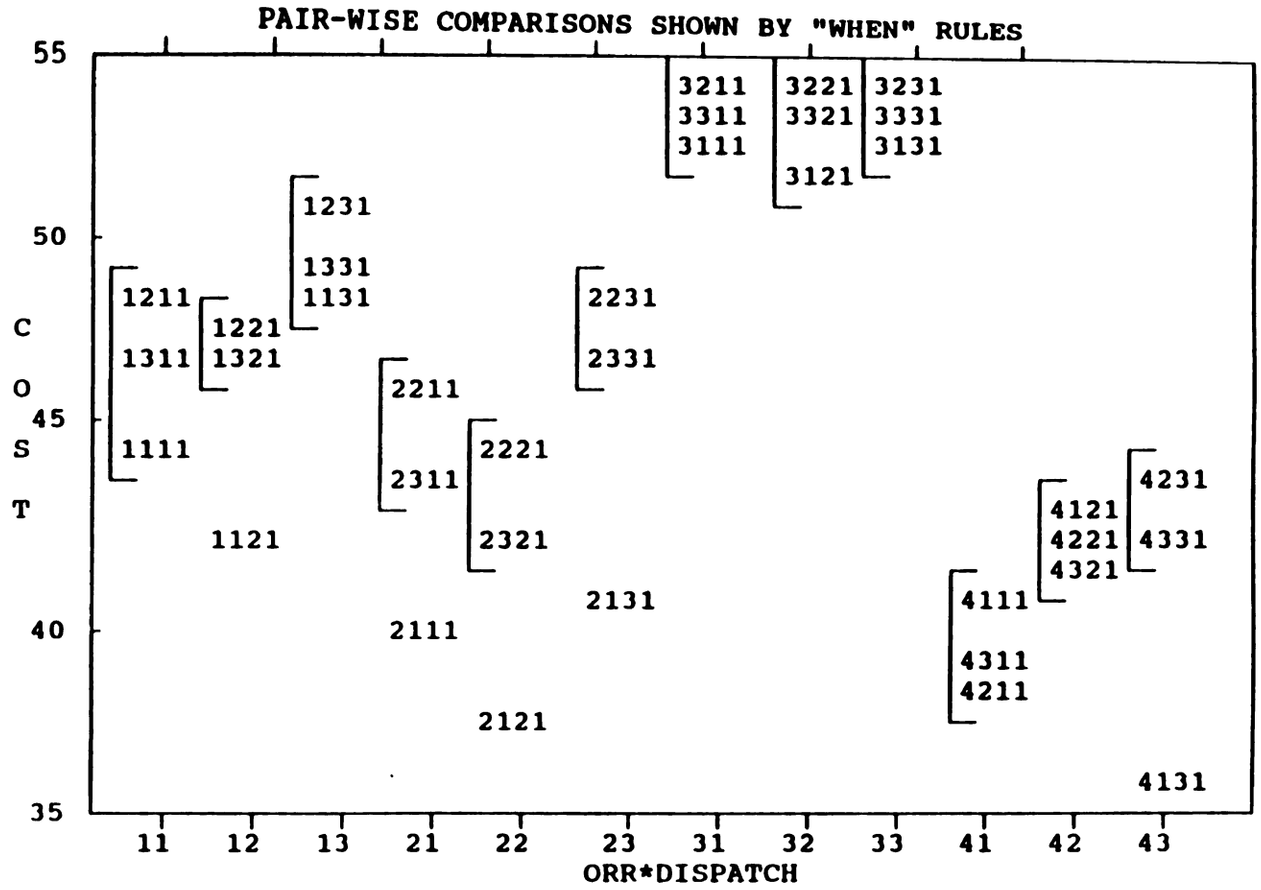
FIGURE 5-7
FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR: 1=IMM,2=MIL,3=CMS,4=MLC. WHEN: 1=QUE,2=JOB,3=CLR.
DISPATCH: 1=FCFS,2=MODD,3=CMDD.
PLAN: 1=SMOOTH,2=UNCONTROLLED

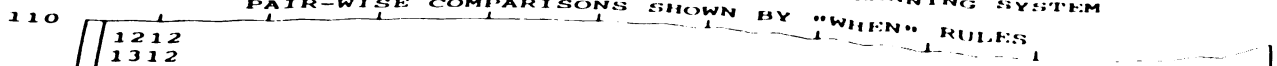
FIGURE 5-8
FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM

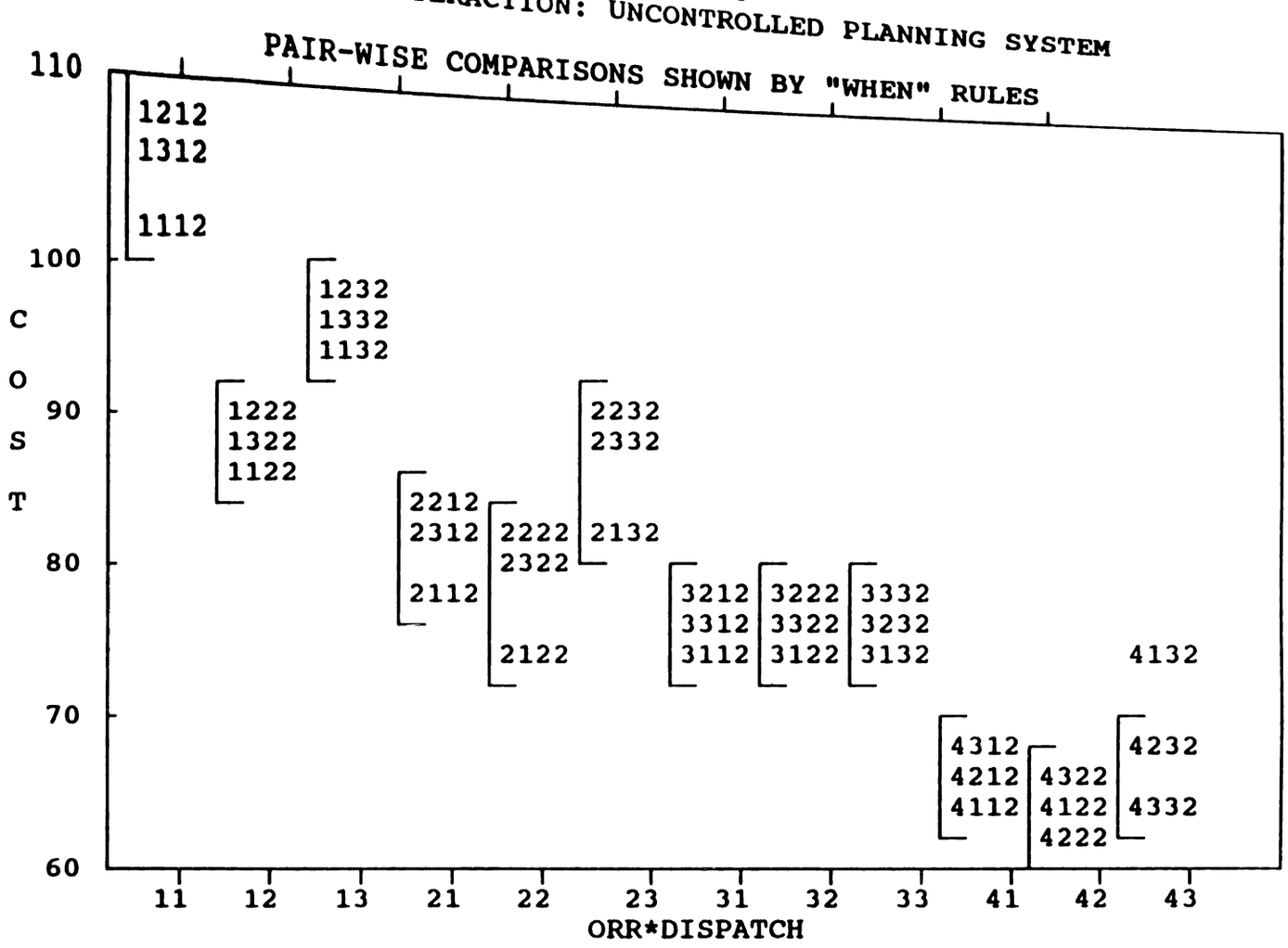
FIGURE 5-8
FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-4
FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM
PAIR-WISE COMPARISONS SHOWN BY "WHEN" RULES





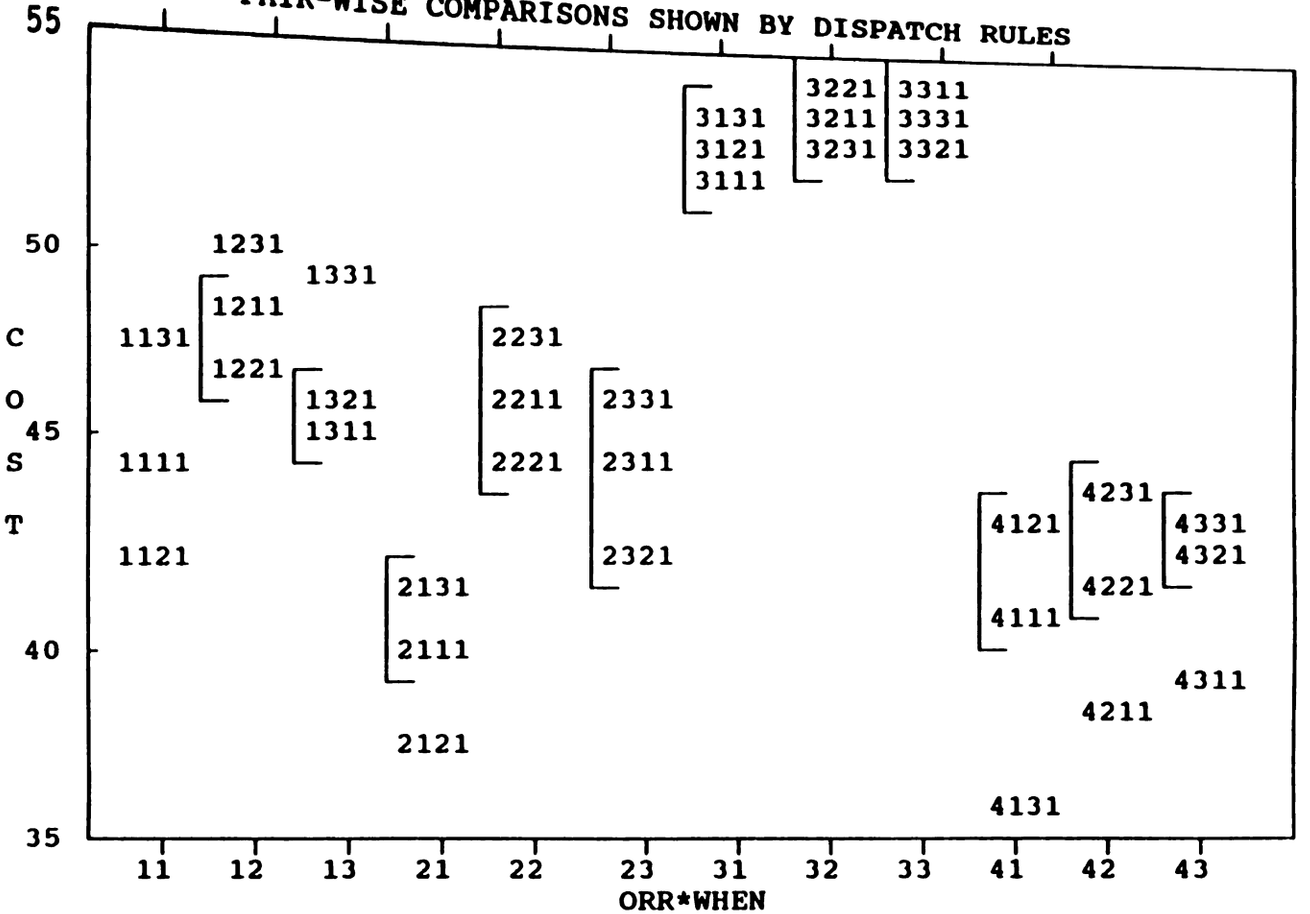
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-10
FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM
PAIR-WISE COMPARISONS SHOWN BY DISPATCH RULES

FIGURE 5-10

FOUR WAY INTERACTION: SMOOTH PLANNING SYSTEM

PAIR-WISE COMPARISONS SHOWN BY DISPATCH RULES



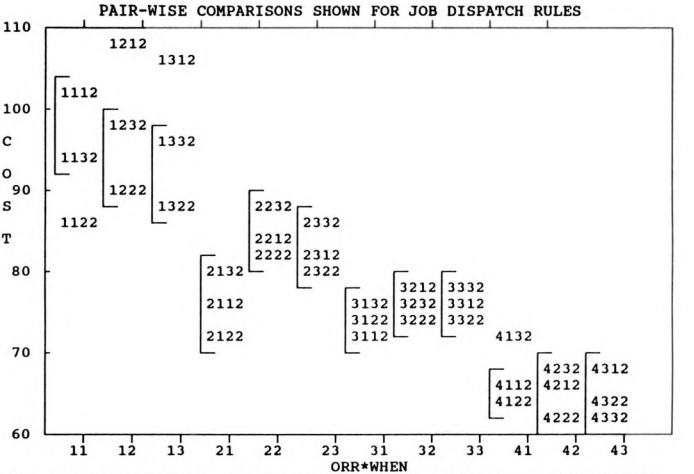
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-11

FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM

PAIR-WISE COMPARISONS SHOWN FOR JOB DISPATCH RULES

FIGURE 5-11
FOUR WAY INTERACTION: UNCONTROLLED PLANNING SYSTEM



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

5.3.3.3 FOU

The in

treatments u

the UNCON

graphed with

treatments sh

With t

which perfor

treatments us

(MM/QUE/M

treatments use

JOB/FCFS/SM

ment, 4131 (M

With the

rule had very li

(MM/QUE/MO

sponding treatm

best performing

other job dispat

5.3.3.3 FOUR WAY INTERACTION AND JOB DISPATCHING

The interaction of the job dispatching rule was examined by graphing treatments using the SMOOTH planning system in Figure 5-10 and those using the UNCONTROLLED planning system in Figure 5-11. The treatments were graphed with the combined ORR and "when" labor assignment rules used in the treatments shown on the X axis.

With the SMOOTH planning system, there were only five treatments which performed significantly better than the other treatments. Two of these treatments used the MODD job dispatch rule. They were treatments 1121 (IMM/QUE/MODD/SMOOTH) and 2121 (MIL/QUE/MODD/SMOOTH). Two treatments used the FCFS job dispatch rule. They were treatments 4211 (MLC/-JOB/FCFS/SMOOTH) and 4311 (MLC/CLR/FCFS/SMOOTH). The fifth treatment, 4131 (MLC/QUE/CMDD/SMOOTH), used the CMDD job dispatch rule.

With the UNCONTROLLED planning system, the choice of the job dispatch rule had very little impact on the system performance. One treatment, 1122 (IMM/QUE/MODD/UNCONTROLLED), was significantly better than corresponding treatments. However, even though the MODD rule was not always the best performing job dispatching rule, it was never significantly worse than the other job dispatching rules.

5.3.3.4 SUMMARY OF FOUR WAY INTERACTION

This analysis of the four way interaction established several points:

- First, the SMOOTH planning system always performed better than UNCONTROLLED planning system.
- Second the choice of the ORR rules was more important when the UNCONTROLLED planning system was used than when the SMOOTH planning system was used. The MLC rule was the best ORR rule or tied for the best for most treatment combinations examined, but not all.
- Third, the choice of the "when" labor assignment rule did not seem to be important with either planning system. If there was a difference in the performance of the "when" labor assignment rule, the QUE rule was the best performing rule with the one exception of treatment 4332 (MLC/CLR/CMDD/UNCONTROLLED) performing better than the corresponding treatment using the QUE rule.
- Fourth, the choice of the job dispatch rule did not seem to have an important impact on the performance of the shop. There was no clear reason to select one job dispatch rule over the other if the planning system was SMOOTH. Even though CMDD was the job dispatch rule in the best performing treatment (4131), the FCFS and MODD dispatch rules performed as well or better than CMDD in subsequent treatments. When the planning system was UNCON-

The fo

mances of th

planning syst

differently for

The ne

As expl

four different

week, and the

hours tardy per

per week.

Before e

fitted to an AN

were conducted

effects, full facto

random number

Tables 5-19, 5-2

labor transfers r

ANOVA of the t

TROLLED, the MODD job dispatching rule never performed significantly worse than the other job dispatching rules.

The four way interaction appears to be caused by the different performances of the ORR, "when" labor assignment and job dispatching rules with each planning system. In addition, the "when" and job dispatching rules performed differently for each ORR rule they were used with.

The next step in the analysis was to investigate the a priori hypotheses.

5.4 INVESTIGATION OF A PRIORI HYPOTHESES

As explained in chapter 4, each a priori hypothesis is investigated using four different performance measures. These measures are mean total cost per week, and the three components of total cost. These components are the mean hours tardy per week, total inventory per week and the number of labor transfers per week.

Before exploring the a priori hypotheses, the component cost data was fitted to an ANOVA model and the multiple comparisons of the treatment means were conducted. The reciprocal of each cost component was fitted to a fixed effects, full factorial ANOVA model. The model was blocked on the common random number streams as discussed earlier. The ANOVA tables are shown in Tables 5-19, 5-20 and 5-21 for the mean hours tardy, total inventory and mean labor transfers respectively. The results for these ANOVA models and for the ANOVA of the total cost are summarized in Table 5-24.

For a

intera ions

one th ee wa

transfe s. Be

multipl com

5-14 fc the

l ecaus

mance i easu

sons. T e or

analyzec dire

arbitrary Ra

should b acce

All the a priori

alternativ e hyp

For all of these cost components, the main effects and higher order interactions are significant, with two exceptions. The four way interaction and one three way interaction were not significant for the mean number of labor transfers. Because the ANOVA models were significant, Bonferroni pair-wise multiple comparisons were calculated. The results are shown in Figures 5-12 to 5-14 for the mean hours tardy, total inventory and labor transfers respectively.

Because the higher order interactions are significant for all four performance measures, the a priori hypotheses are investigated using multiple comparisons. The original linear contrasts for each hypothesis are shown but not analyzed directly because with the higher order interactions, the weights are arbitrary. Rather the multiple comparisons are used to determine whether H_o should be accepted or rejected. Each hypothesis will now be examined in turn. All the a priori hypotheses are formally stated in chapter four, so only the alternative hypothesis are restated here.

TABLE 5-21

TABLE 5-21
ANOVA TABLE: RECIPROCAL HOURS TARDY

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	11.64273772	0.06848669	564.55	0.0001
ERROR	7029	0.85269507	0.00012131		
CORRECTED TOTAL	7199	12.495432979			

R-SQUARE	C.V.	ROOT MSE	GRAND MEAN
0.931759	16.06076	0.01101413	0.06857787

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
ORR RULE	3	3.31777872	1.10592624	9116.45	0.0001
WHEN ASSIGNMENT	2	0.00207361	0.00103680	8.55	0.0002
DISPATCH RULE	2	0.00941089	0.00470545	38.79	0.0001
PLANNING SYSTEM	1	6.05244497	6.05244497	49891.97	0.0001
BLOCKING FACTOR	99	0.31076557	0.00313905	25.88	0.0001
ORR*WHEN	6	0.00498941	0.00083157	6.85	0.0001
ORR*DISPATCH	6	0.05504470	0.00917412	75.62	0.0001
ORR*PLANNING	3	1.18713629	0.39571210	3261.96	0.0001
WHEN*DISPATCH	4	0.05958109	0.01489527	122.79	0.0001
WHEN*PLAN	2	0.00568293	0.00284146	23.42	0.0001
DISPATCH*PLAN	2	0.01426412	0.00713206	58.79	0.0001
ORR*WHEN*DISPATCH	12	0.19399116	0.01616593	133.26	0.0001
ORR*WHEN*PLAN	6	0.01412997	0.00235500	19.41	0.0001
ORR*DISPATCH*PLAN	6	0.07674404	0.01279067	105.44	0.0001
WHEN*DISPATCH*PLAN	4	0.08298888	0.02074722	171.03	0.0001
ORR*WHEN*DISPATCH*PLAN	12	0.25571139	0.02130928	175.66	0.0001

TABLE 5-22
ANOVA TABLE: RECIPROCAL TOTAL INVENTORY

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	0.00883054	0.00005194	2202.53	0.0001
ERROR	7029	0.00016577	0.00000002		

TABLE 5-22
ANOVA TABLE: RECIPROCAL TOTAL INVENTORY

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	0.00883054	0.00005194	2202.53	0.0001
ERROR	7029	0.00016577	0.00000002		
CORRECTED TOTAL	7199	0.00899631			

R-SQUARE	C.V.	ROOT MSE	GRAND MEAN
0.981573	1.579726	0.00015357	0.00972134

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
ORR RULE	3	0.00570383	0.00190128	80617.46	0.0001
WHEN ASSIGNMENT	2	0.00014500	0.00007250	3074.07	0.0001
DISPATCH RULE	2	0.00008419	0.00004210	1784.94	0.0001
PLANNING SYSTEM	1	0.00056884	0.00056884	24119.62	0.0001
BLOCKING FACTOR	99	0.00004259	0.00000043	18.24	0.0001
ORR*WHEN	6	0.00043095	0.00007183	3045.52	0.0001
ORR*DISPATCH	6	0.00026268	0.00004378	1856.32	0.0001
ORR*PLANNING	3	0.00023880	0.00007960	3375.21	0.0001
WHEN*DISPATCH	4	0.00006473	0.00001618	686.20	0.0001
WHEN*PLAN	2	0.00001955	0.00000978	414.50	0.0001
DISPATCH*PLAN	2	0.00001402	0.00000701	297.15	0.0001
ORR*WHEN*DISPATCH	12	0.00019270	0.00001606	680.90	0.0001
ORR*WHEN*PLAN	6	0.00005972	0.00000995	422.01	0.0001
ORR*DISPATCH*PLAN	6	0.00005046	0.00000841	356.61	0.0001
WHEN*DISPATCH*PLAN	4	0.00023872	0.00005968	2530.54	0.0001
ORR*WHEN*DISPATCH*PLAN	12	0.00071378	0.00005948	2522.11	0.0001

TABLE 5-23
ANOVA TABLE: RECIPROCAL LABOR TRANSFERS

TABLE 5-23
ANOVA TABLE: RECIPROCAL LABOR TRANSFERS

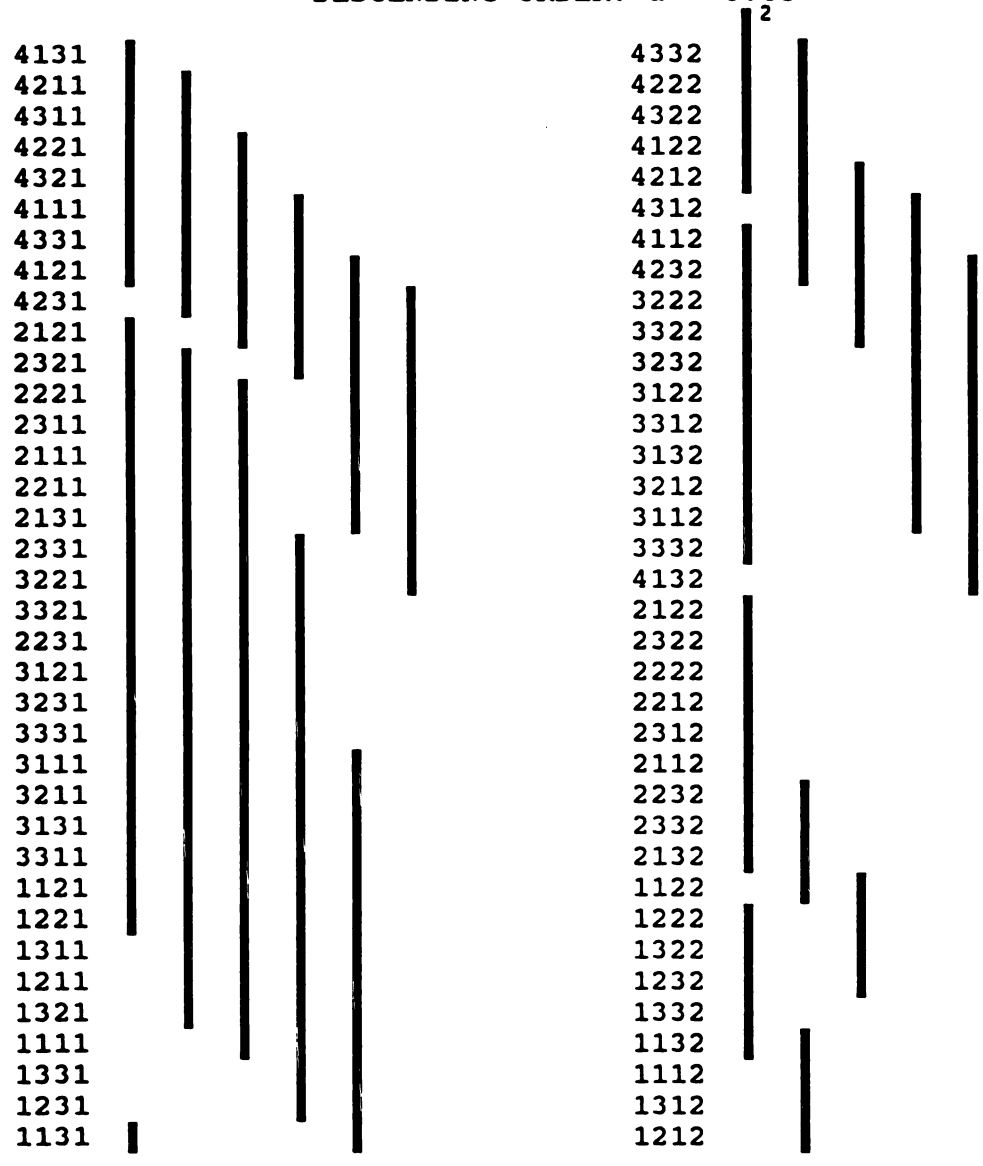
SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	170	0.00883054	0.00005194	2202.53	0.0001
ERROR	7029	0.00016577	0.00000002		
CORRECTED TOTAL	7199	0.00899631			

R-SQUARE	C.V.	ROOT MSE	GRAND MEAN
0.993669	5.677660	0.00066242	0.01166711

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
ORR RULE	3	0.33079979	0.11026660	99999.99	0.0001
WHEN ASSIGNMENT	2	0.06866030	0.03483002	79375.78	0.0001
DISPATCH RULE	2	0.00060060	0.00030030	684.37	0.0001
PLANNING SYSTEM	1	0.00185760	0.00185760	4233.37	0.0001
BLOCKING FACTOR	99	0.00049808	0.00000503	11.47	0.0001
ORR*WHEN	6	0.07512083	0.01252014	28532.74	0.0001
ORR*DISPATCH	6	0.00103207	0.00017201	392.01	0.0001
ORR*PLANNING	3	0.00243878	0.00081293	1852.74	0.0001
WHEN*DISPATCH	4	0.00002043	0.00000511	11.64	0.0001
WHEN*PLAN	2	0.00084652	0.00042326	964.58	0.0001
DISPATCH*PLAN	2	0.00002404	0.00001202	27.39	0.0001
ORR*WHEN*DISPATCH	12	0.00011988	0.00000999	22.77	0.0001
ORR*WHEN*PLAN	6	0.00103354	0.00017226	392.56	0.0001
ORR*DISPATCH*PLAN	6	0.00003137	0.00000523	11.92	0.0001
WHEN*DISPATCH*PLAN	4	0.00000318	0.00000079	1.81	0.1236*
ORR*WHEN*DISPATCH*PLAN	12	0.00000367	0.00000031	0.70	0.7561*

* DENOTES THAT EFFECT IS INSIGNIFICANT AT $\alpha = 0.05$

FIGURE 5-12
COMPARISONS: RECIPROCAL HOURS TARDY[1]
DESCENDING ORDER: $\alpha = 0.05$



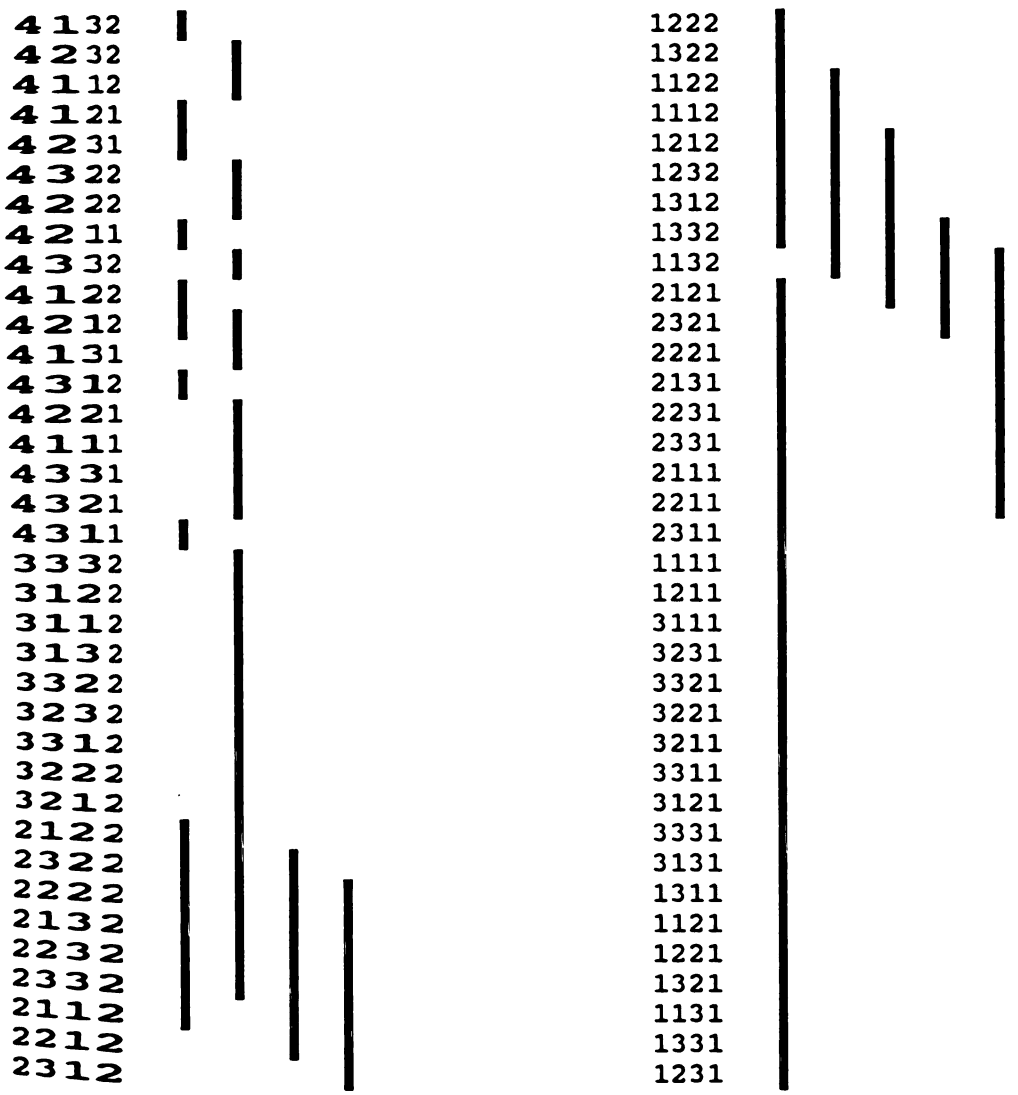
LEGEND DIGITS 1,2,3 & 4:
DIGIT 1=ORR (1=IMM, 2=MIL,3=CMS,4=MLC)
DIGIT 2=WHEN(1=QUE,2=JOB,3=CLR)
DIGIT 3=DISPATCH(1=FCFS,2=MODD,3=CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

---

[1]. ALL T TESTS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03. $\alpha = 0.05$

[2]. Indicates not significant difference from prior column.

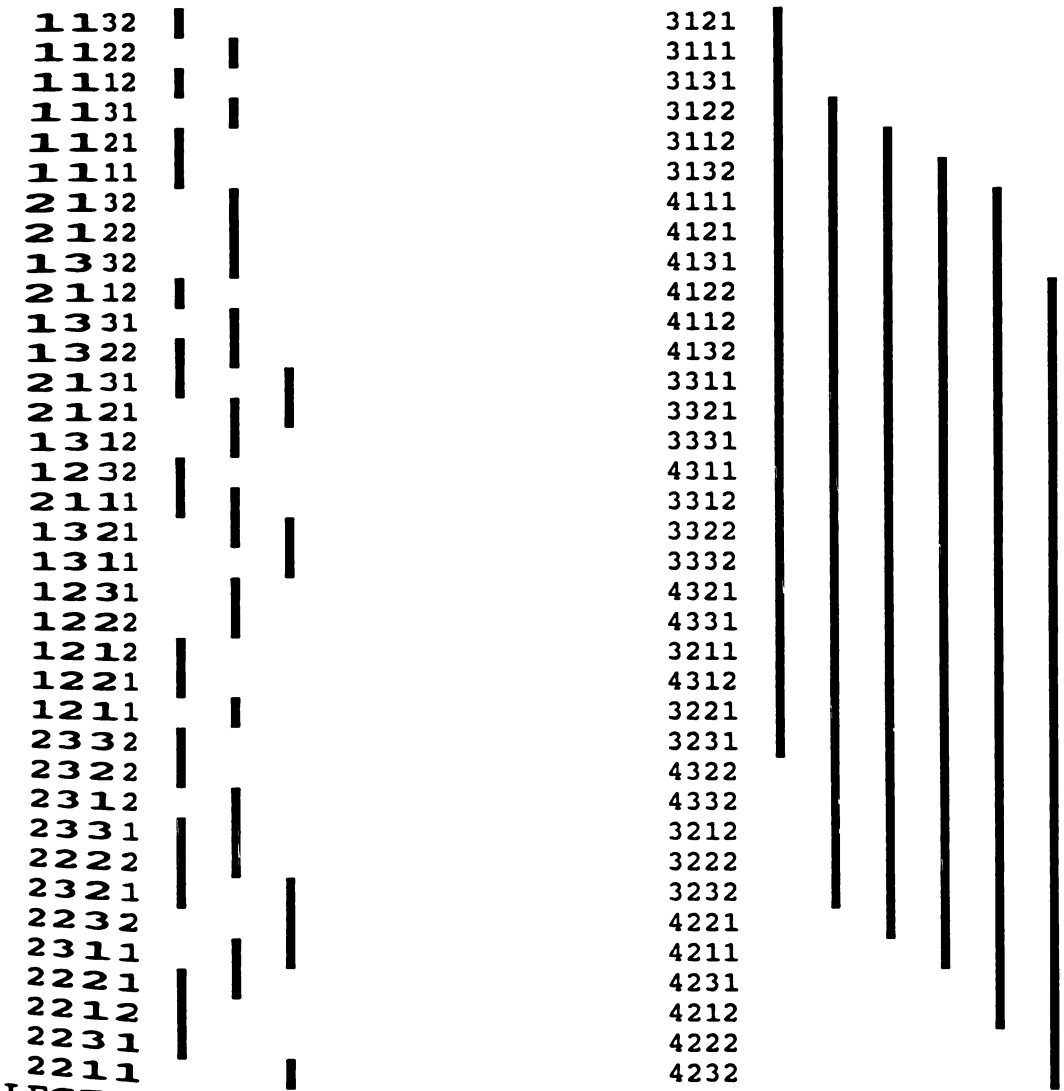
FIGURE 5-13
BONFERRONI COMPARISONS: TOTAL INVENTORY RECIPROCAL[1]
ALPHA = 0.05: DESCENDING ORDER



LEGEND DIGITS 1,2,3,4:
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

[1]. ALL T TESTS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-14
BONFERRONI COMPARISONS: RECIPROCAL LABOR TRANSFERS[1]
DESCENDING ORDER: $\alpha = 0.05$



LEGEND DIGITS 1,2,3 & 4:
DIGIT 1=ORR RULE (1=IMM, 2=MIL,3=CMS,4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

[1]. ALL T TESTS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03. $\alpha = 0.05$

TABLE 5-24
SUMMARY OF ANOVA FOR TOTAL COST AND COMPONENTS

CELL CONTENTS ARE PROBABILITY THAT F VALUE IS GREATER THAN CRITICAL F

	PERFORMANCE MEASURES			
SOURCE	RECIPROCAL MEAN TOTAL COST	RECIPROCAL MEAN HOURS TARDY/WEEK	RECIPROCAL TOTAL INVENTORY	RECIPROCAL MEAN LABOR TRANSFERS
MODEL	0.0001	0.0001	0.0001	0.0001
ORR RULE	0.0001	0.0002	0.0001	0.0001
WHEN ASSIGNMENT	0.0001	0.0001	0.0001	0.0001
DISPATCH RULE	0.0001	0.0001	0.0001	0.0001
PLANNING SYSTEM	0.0001	0.0001	0.0001	0.0001
BLOCKING FACTOR	0.0001	0.0001	0.0001	0.0001
ORR*WHEN	0.0001	0.0001	0.0001	0.0001
ORR*DISPATCH	0.0001	0.0001	0.0001	0.0001
ORR*PLANNING	0.0001	0.0001	0.0001	0.0001
WHEN*DISPATCH	0.0001	0.0001	0.0001	0.0001
WHEN*PLAN	0.0001	0.0001	0.0001	0.0001
DISPATCH*PLAN	0.0001	0.0001	0.0001	0.0001
ORR*WHEN*DISPATCH	0.0001	0.0001	0.0001	0.0001
ORR*WHEN*PLAN	0.0001	0.0001	0.0001	0.0001
ORR*DISPATCH*PLAN	0.0001	0.0001	0.0001	0.0001
WHEN*DISPATCH*PLAN	0.0001	0.0001	0.0001	0.1236*
ORR*WHEN*DISPATCH*PLAN	0.0001	0.0001	0.0001	0.7561*

* DENOTES INSIGNIFICANT EFFECT AT $\alpha = 0.05$

The

planning sy

planning sy

The

H_A that the

TROLLED p

(Group 1) p

planning sys

The

Figure 5-12

system prod

system.

The

the mean n

There is no

less invento

The a

rules improv

5.4.1 HYPOTHESIS 1

The first alternative hypothesis was treatments using the SMOOTH planning system perform better than treatments using the UNCONTROLLED planning system.

The multiple comparisons in Figure 5-3 allow us to support the alternative H_A that the SMOOTH planning system does perform better than the UNCONTROLLED planning system. All treatments with the SMOOTH planning system (Group 1) performed better than any treatment with the UNCONTROLLED planning system (Group 2).

The multiple comparisons of the mean hours tardy per week shown in Figure 5-12 allow us to support the alternative H_A that the SMOOTH planning system produces fewer tardy hours per week than the UNCONTROLLED planning system.

The multiple comparisons of the mean total inventory (Figure 5-13) and the mean number of labor transfers (Figure 5-14) do not allow us to reject H_o. There is no support for the hypothesis that the SMOOTH planning system has less inventory than the UNCONTROLLED planning system.

5.4.2 HYPOTHESIS 2

The alternative hypothesis 2 is that the use of LABOR information in ORR rules improves shop performance when using the SMOOTH planning system.

This

them to the

graphed in

in the same

rules) a ver

that CMS pe

MIL rules pe

most MLC an

tally.

There

information

mance.

The m

Figure 5-12 a

between trea

To ana

comparisons

treatment usi

same vertical

treatments us

OPR rule.

This hypothesis averages the CMS and MLC rules together and compares them to the MIL rule. All 27 treatments contained in the hypothesis were graphed in Figure 5-15. If there is no significant difference between treatments in the same vertical column (meaning they have the same "when" and dispatch rules) a vertical line is drawn between the treatments. This figure demonstrates that CMS performs poorly compared to MLC and MIL, and that some MLC and MIL rules perform the same. The multiple comparisons in Figure 5-3 show that most MLC and MIL treatments cannot be distinguished from each other horizontally.

There was no support for the alternative hypothesis that the use of LABOR information with a SMOOTH planning system significantly improved performance.

The multiple comparisons of the mean hours tardy per week shown in Figure 5-12 also do not allow us to reject H_o. There is no significant difference between treatments using MIL and those using CMS.

To analyze the effect of the ORR rule on total inventory, the multiple comparisons of total inventory are organized in Figure 5-16, so that each treatment using the same "when" and job dispatching rule is graphed on the same vertical axis. This graph does not allow H_o to be rejected because all the treatments using the CMS ORR rule are tied with the treatments using the MIL ORR rule.

comp

figure

ORR

syster

them

graph

better

rules

suppo

perfor

reject

does i

Figure

rules.

reatme

To analyze the effect of the ORR rule on total labor transfers, the multiple comparisons of total labor transfers are graphed in Figure 5-17. Based on this figure, H_0 is accepted because MIL performs better than both CMS and MLC.

5.4.3 HYPOTHESIS 3

The alternative hypothesis 3 is that the use of LABOR information in the ORR rule improves shop performance when using the UNCONTROLLED planning system.

This hypothesis averages the CMS and MLC rules together and compares them to the MIL rule. The pair-wise comparisons of these treatments are graphed in Figure 5-18. Treatments using the MLC rules generally perform better than treatments using the MIL ORR rules. Treatments using the CMS rules perform worse than treatments using the MIL ORR rules. There is no support for the alternative hypothesis that the use of labor information improves performance of the ORR rule.

The multiple comparisons of the mean hours tardy per week allow us to reject H_0 and support the alternative hypothesis that the use of labor information does improve performance.

The multiple comparisons of the total inventory level are graphed in Figure 5-19. They are organized to compare treatments using different ORR rules. There are some significant differences between the CMS and the MIL treatments, but not enough to reject H_0 and support the alternative hypothesis.

Figure 5

number

native h

labor tra

T

MLC OR

shop by

Th

where it

alternati

improves

Th

H_0 to be

Th

allow H_0

The

do not all

The multiple comparisons for the total labor transfers are graphed in Figure 5-20. The treatments using the CMS and MLC ORR rules have the largest number of labor transfers. H_0 was accepted. There is no support for the alternative hypothesis that the use of LABOR information reduces the number of labor transfers.

5.4.4 HYPOTHESIS 4

The alternative hypothesis 4 was that finite loading of the shop with the MLC ORR rule improves shop performance compared to infinite loading of the shop by the CMS ORR rule when using the SMOOTH planning system.

The treatments in the hypothesis (MLC vs. CMS) are shown in Figure 5-21 where it is evident from the multiple comparisons that H_0 is rejected. The alternative hypothesis, that finite loading of the shop using the MLC ORR rule improves performance is supported.

The multiple comparisons of the mean hours tardy in Figure 5-12 allow H_0 to be rejected. Finite loading reduces the number of hours of tardy work.

The multiple comparisons of the total inventory shown in Figure 5-16 allow H_0 to be rejected.

The multiple comparisons of the total labor transfers shown in Figure 5-17 do not allow H_0 to be rejected.

Th

MLC ORI

the CMS

Th

Figure 5-

and that

performar

UNCONT

different t

The

for the me

from treatr

The

total inver

The

difference b

number of

The a

stop perform

5.4.5 HYPOTHESIS 5

The alternative hypothesis 5 was that finite loading of the shop with the MLC ORR rule improves performance compared to infinite loading of the shop by the CMS rule when using the UNCONTROLLED planning system.

The treatments involved in this contrast (MLC vs. CMS) are graphed in Figure 5-22 where it is evident from the multiple comparisons that H_o is rejected and that the alternative hypothesis, that finite loading of the shop improves performance is supported. Treatment 4132 (MLC/QUE/CMDD-/UNCONTROLLED) is the single exception to this because it is not significantly different than treatment 3132 (CMS/QUE/CMDD/UNCONTROLLED).

The multiple comparisons in Figure 5-12 do not allow H_o to be rejected for the mean hours tardy per week. Four treatments using MLC are not different from treatments using CMS.

The multiple comparisons in Figure 5-19 allow H_o to be rejected for the total inventory level.

The multiple comparisons in Figure 5-20 support H_o that there is no difference between the MLC and CMS rules when the performance measure is the number of labor transfers.

5.4.6 HYPOTHESIS 6

The alternative hypothesis 6 is that the use of ORR rules in SFC improves shop performance compared to SFC systems which do not use ORR mechanisms.

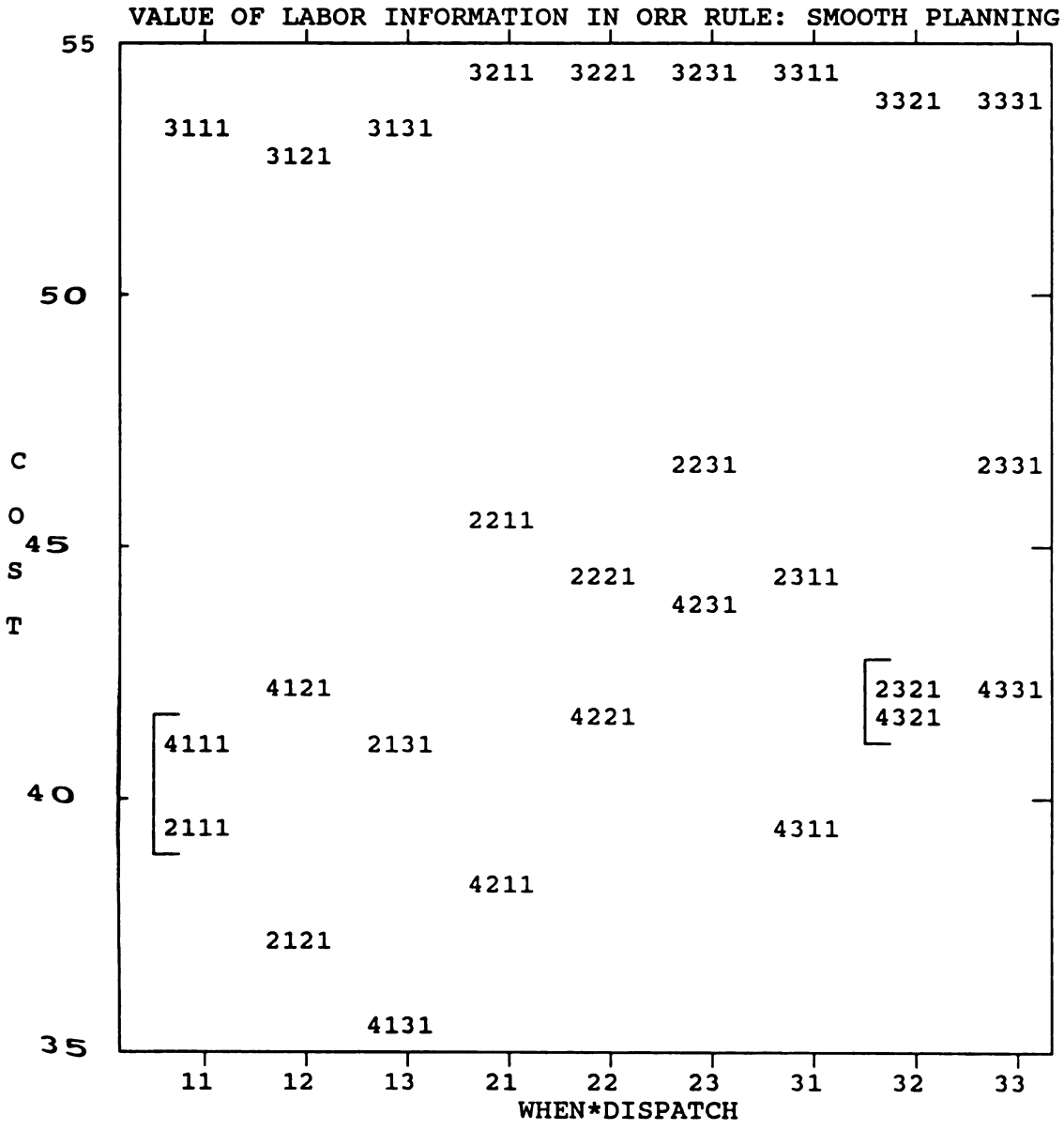
This hypothesis involves all 72 treatments of the experiment. It is graphed in Figure 5-23. Examination of the treatments which use the SMOOTH planning system show that in many cases IMM performs better than the corresponding treatment using an ORR rule. The primary example of this is that all treatments using the CMS ORR rule and the SMOOTH planning system perform worse than the corresponding treatments using the IMM ORR rule and the SMOOTH planning system. H_0 is accepted, because it is clear that some treatments with ORR rules perform worse than treatments which do not use ORR rules.

The multiple comparisons in Figure 5-12 of the mean hours tardy per week allow H_0 to be rejected. The use of the MLC, CMS and MIL ORR rules does not improve system performance for a given planning system.

Likewise, the multiple comparisons in Figure 5-13 of the total inventory allow H_0 to be rejected, because the use of ORR rules never improves system performance for a given planning system. However, the multiple comparisons of the number of labor transfers in Figure 5-14 allows H_0 to be accepted. Treatments using either the MLC and the CMS ORR rules result in more labor transfers than the immediate release of jobs to the shop.

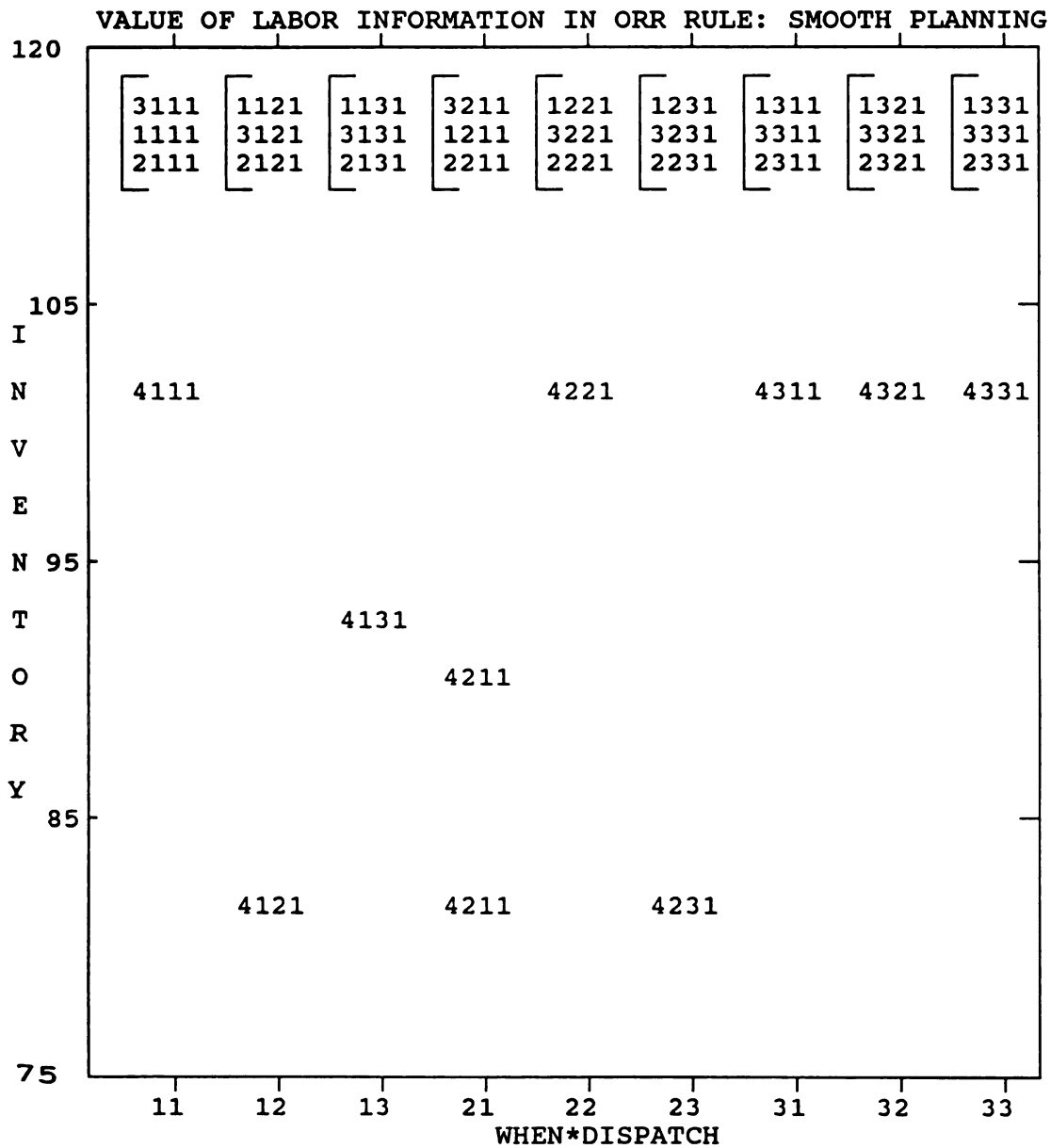
5
50
45
40
35

FIGURE 5-15
HYPOTHESIS 2: TOTAL COST



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR:1=IMM,2=MIL,3=CMS,4=MLC. WHEN:1=QUE,2=JOB,3=CLR.
DISPATCH:1=FCFS,2=MODD,3=CMDD.
PLAN:1=SMOOTH,2=UNCONTROLLED.

FIGURE 5-16
HYPOTHESIS 2: TOTAL INVENTORY



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR:1=IMM,2=MIL,3=CMS,4=MLC. WHEN:1=QUE,2=JOB,3=CLR
DISPATCH: 1=FCFS,2=MODD,3=CMDD.
PLAN: 1=SMOOTH,2=UNCONTROLLED.

185

155

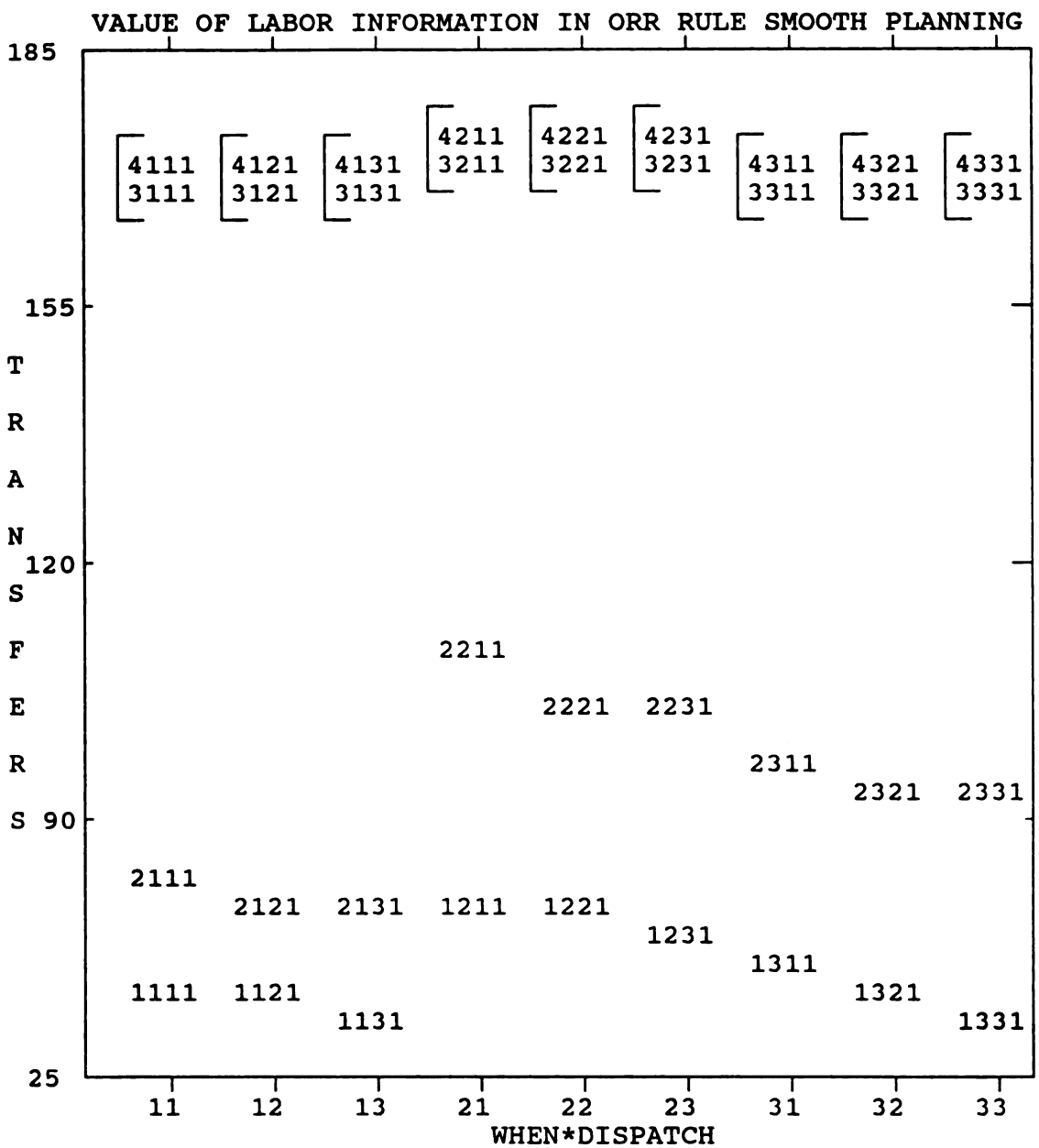
T
R
A
N
S
F
E
R

120

90

25

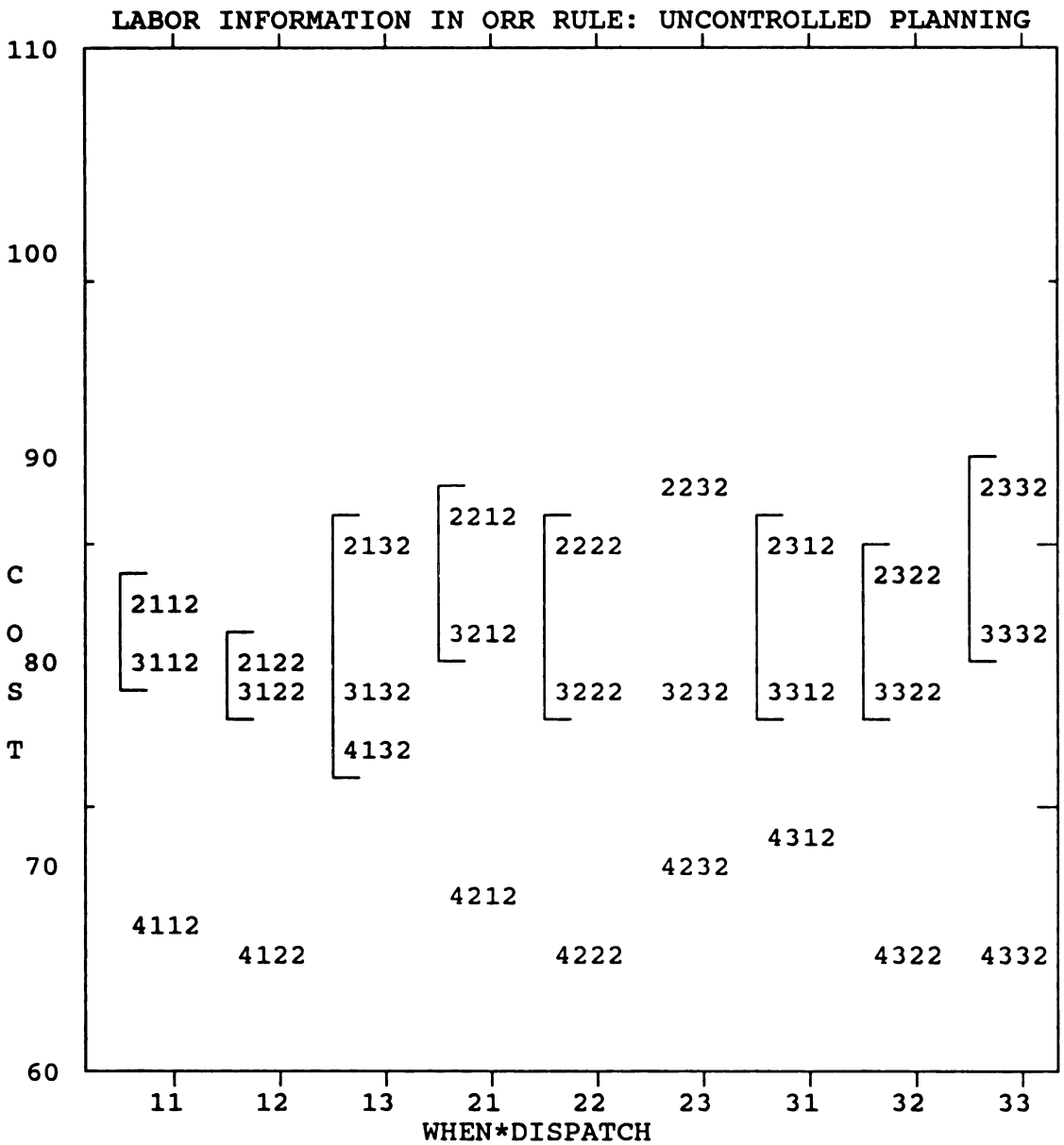
FIGURE 5-17
HYPOTHESIS 2: LABOR TRANSFERS



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR:1=IMM,2=MIL,3=CMS,4=MLC. WHEN:1=QUE,2=JOB,3=CLR
DISPATCH: 1=FCFS,2=MODD,3=CMDD.
PLAN: 1=SMOOTH,2=UNCONTROLLED.

110
100
90
80
70
60

FIGURE 5-18
HYPOTHESIS 3: TOTAL COST



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR: 1=IMM,2=MIL,3=CMS,4=MLC. WHEN:1=QUE,2=JOB,3=CLR.
DISPATCH: 1=FCFS,2=MODD,3=CMDD.
PLAN: 1=SMOOTH,2=UNCONTROLLED.

1

1

9

C

D

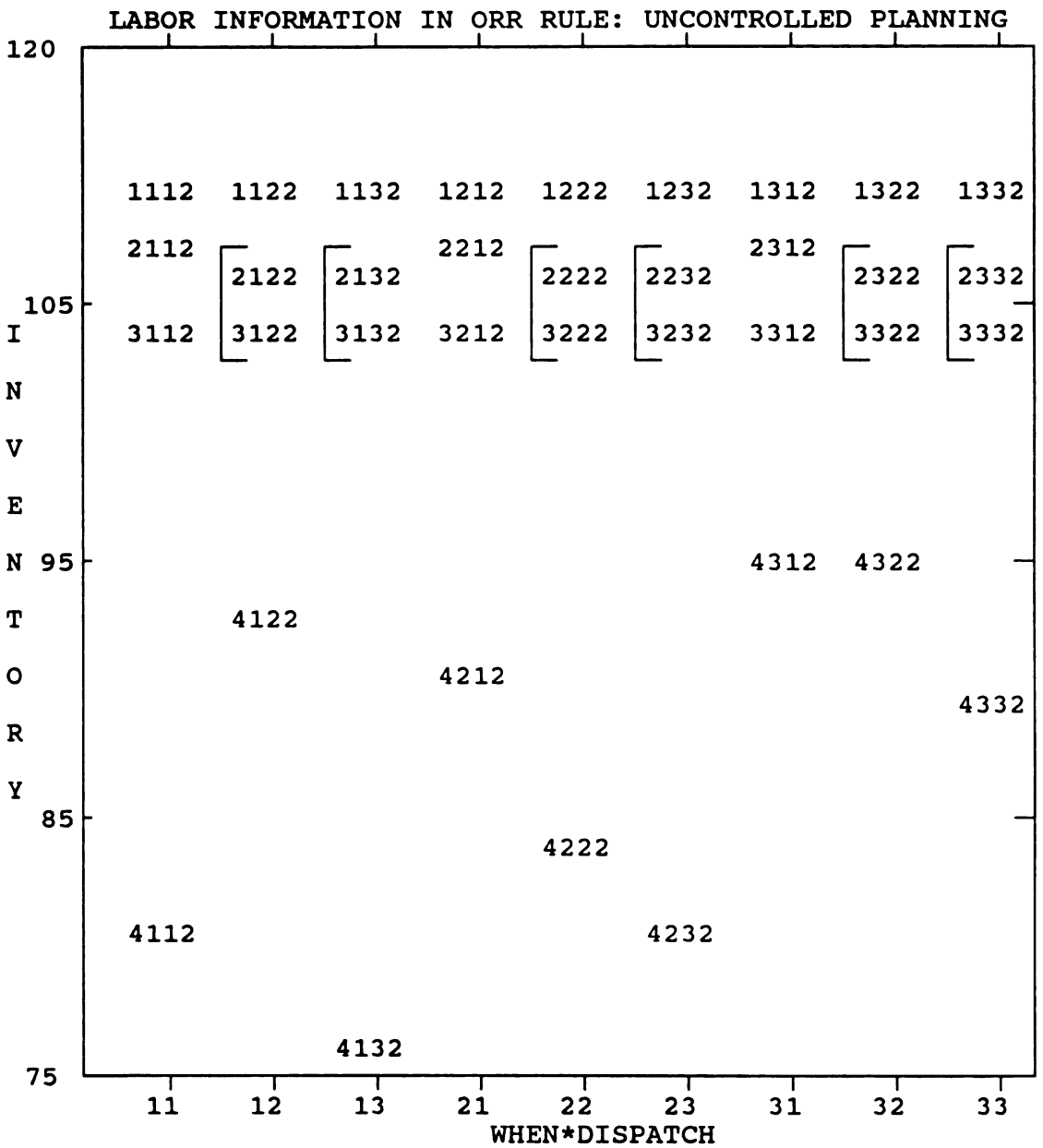
S

7

70

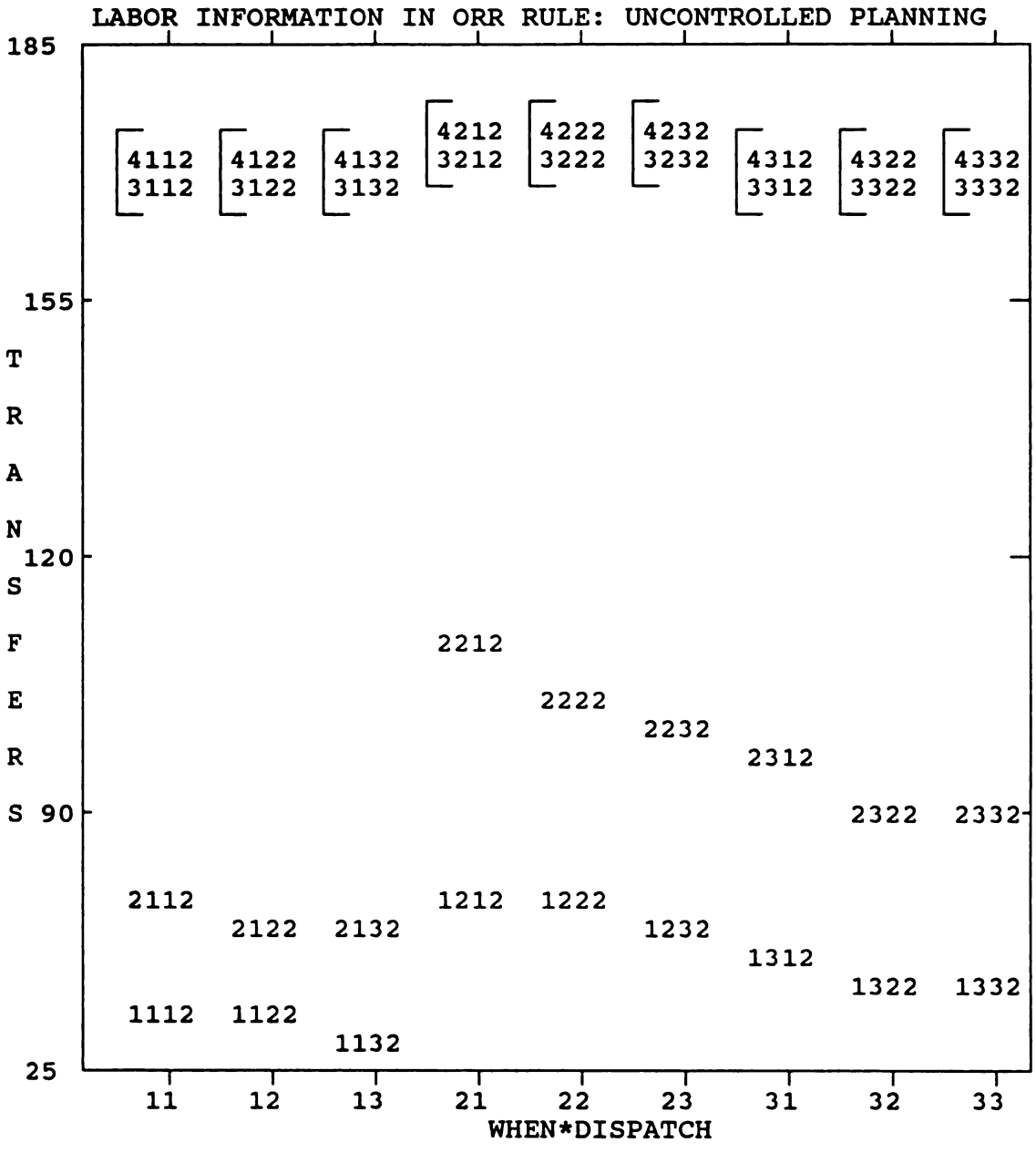
60

FIGURE 5-19
HYPOTHESIS 3: TOTAL INVENTORY



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR:1=IMM,2=MIL,3=CMS,4=MLC. WHEN:1=QUE,2=JOB,3=CLR
DISPATCH: 1=FCFS,2=MODD,3=CMDD.
PLAN: 1=SMOOTH,2=UNCONTROLLED.

FIGURE 5-20
HYPOTHESIS 3: LABOR TRANSFERS



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR:1=IMM,2=MIL,3=CMS,4=MLC. WHEN:1=QUE,2=JOB,3=CLR
DISPATCH: 1=FCFS,2=MODD,3=CMDD.
PLAN: 1=SMOOTH,2=UNCONTROLLED.

110

105

100

95

85

80

75

70

65

C

O 60

S

55

T

50

45

40

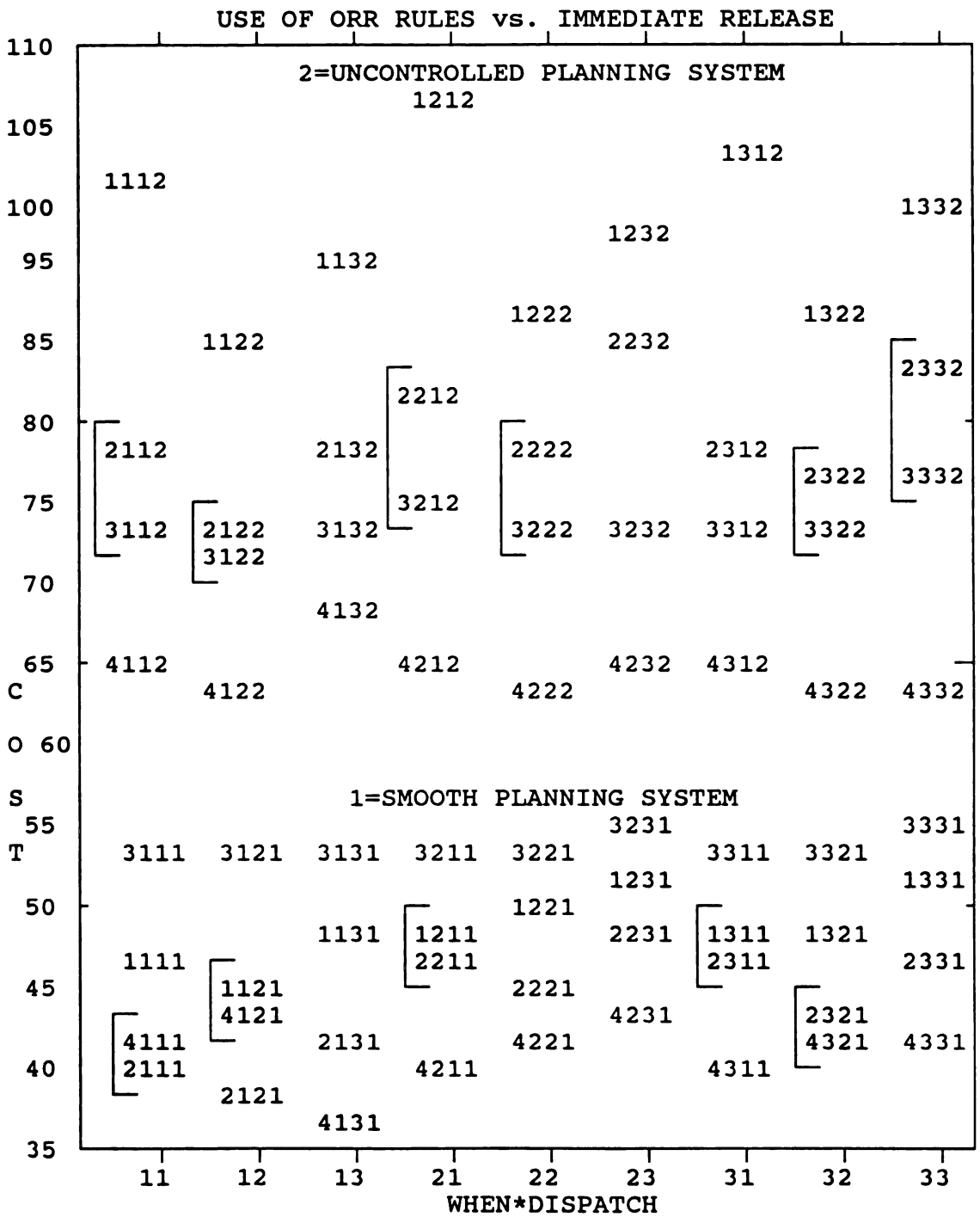
35

TRE

CR

DI

FIGURE 5-23
HYPOTHESIS 6: TOTAL COST



TREATMENTS USE 4 DIGIT CODE (ORR, WHEN, DISPATCH, PLAN)
ORR: 1=IMM,2=MIL,3=CMS,4=MLC. WHEN: 1=QUE,2=JOB,3=CLR.
DISPATCH: 1=FCFS,2=MODD,3=CMDD.

disp

syst

trea

Wh

POC

rem

of jo

alter

job

in P

is ac

treat

Figu

rules.

showe

rules.

5.4.7 HYPOTHESIS 7

The alternative hypothesis 7 is that the use of POOL information in job dispatching rules improves shop performance when using the SMOOTH planning system.

The treatments in this contrast are graphed in Figure 5-25, so that treatments using different job dispatching rules can be compared vertically. While the best performing treatment, 4131 (MLC/QUE/CMDD/SMOOTH) used POOL information in the CMDD dispatch rule, pair-wise comparisons of the remaining treatments did not show any significant difference between the types of job dispatch rules used. H_0 was accepted. There was no support for the alternative hypothesis that POOL information improved the performance of the job dispatch rule.

The multiple comparisons of the mean hours tardy per week were graphed in Figure 5-25, they were organized to compare their job dispatch rules. The H_0 is accepted because there is no significant performance difference between the treatments using the different job dispatching rules.

The multiple comparisons of the total inventory per week were graphed in Figure 5-26. They were organized to allow comparison of their job dispatching rules. H_0 was accepted because only treatments using the MLC ORR rule showed a significant difference between the performance of the job dispatching rules.

grap

rules

diffe

the n

dispa

plann

compa

job di

hypoth

rule.

job dis

vertica

not sig

signifi

treatme

The multiple comparisons of the number of labor transfers per week were graphed in Figure 5-27. They were organized to compare their job dispatching rules. The comparisons allow H_0 to be accepted. There were some significant differences with treatments using IMM and the MIL ORR rules, but the rest of the treatments have no significant differences.

5.4.8 HYPOTHESIS 8

The alternative hypothesis 8 is that the use of POOL information in job dispatching rules improves shop performance when using the UNCONTROLLED planning system.

The treatments in this contrast are graphed in Figure 5-28. The multiple comparisons do not show a significant difference between the performance of the job dispatch rules, so H_0 is accepted. There is no support for the alternative hypothesis that POOL information improves the performance of the job dispatch rule.

The treatments are organized in Figures 5-29, 5-30 and 5-31 so that the job dispatching rules can be compared to their corresponding treatments on the vertical axis. The null hypothesis, H_0, denoting that the job dispatching rules do not significantly influence the hours tardy per week (Figure 5-29) is accepted.

The null hypothesis, H_0, indicating that the job dispatching rules do not significantly influence total inventory (Figure 5-30) is accepted. However, if the treatment use the MLC ORR rule, the job dispatch rule does have an impact on

performance. CMDD is the best job dispatching rule twice and MODD was the best rule once.

The null hypothesis that the job dispatching rule does not influence the number of labor transfers (Figure 5-31) is accepted because treatments using CMS and MLC ORR rules have no significant difference caused by the job dispatching rules. However, there are some significant differences when the IMM or MIL ORR rules are used. MODD is the best rule twice and CMDD is the best rule twice.

5.4.9 HYPOTHESIS 9

The alternative hypothesis 9 is that the use of POOL information in the "when" labor assignment rules improves shop performance when using the SMOOTH planning system.

The treatments in this contrast are shown in Figure 5-32. There are no treatments using the CLR "when" rule which perform significantly better than corresponding treatments using other job dispatching rules.

The multiple comparisons of the mean tardy hours per week, total inventory per week and the number of labor transfers per week are graphed in Figures 5-33, 5-34 and 5-35 respectively. Each graph is organized so that treatments using different "when" labor assignment rules are compared vertically. The null hypothesis, H_o, that "when" rules do not influence the hours tardy per week (Figure 5-33) is accepted. The null hypothesis, H_o, that the "when" rule

d

n

c

n

th

w

"w

cy

mu

"w

inv

fig

"wh

37)

sig

trea

by si

does not significantly influence total inventory (Figure 5-34) is accepted. The null hypothesis, H_o, that "when" rules do not significantly influence the number of labor transfers (Figure 5-35) is accepted, because there are no differences with treatments using CMS and the MLC ORR rules. However, with treatments using the IMM and MIL ORR rules, there are significant differences and the QUE "when" rule is always the best rule.

5.4.10 Hypothesis 10

The alternative hypothesis 10 is that the use of POOL information in the "when" labor assignment rules improves shop performance when using the UNCONTROLLED planning system.

The treatments in this hypothesis are graphed in Figure 5-36. The multiple comparisons show no advantage to using LABOR information in the "when" rule. The null hypothesis, H_o, is accepted.

The multiple comparisons of the mean hours tardy per week, the total inventory per week, and the number of labor transfers per week are graphed in Figures 5-37, 5-38 and 5-39 respectively. The null hypothesis, H_o, that the "when" rule has no statistically significant impact on the hours tardy (Figure 5-37) is accepted. The null hypothesis, H_o, that the "when" rule has no statistically significant impact on the total inventory (Figure 5-38) is accepted. However, for treatments where treatments use MLC as the ORR rule, there are some statistically significant differences between the "when" rules. The QUE rule is the best rule

twice and the worst rule once. The null hypothesis, H_o, that the "when" labor assignment rule has no statistically significant impact on labor transfers (Figure 5-39) is accepted because all treatments using CMS and MLC ORR rules are tied. However for treatments using IMM or MIL ORR rules, the QUE rule always performs the best.

5.4.11 SUMMARY OF HYPOTHESES EVALUATION

The analysis of the a priori hypotheses is summarized in Table 5-25. Each of these hypotheses is addressed in a separate section below.

5.4.11.1 SUMMARY OF HYPOTHESIS 1

There was strong support for the alternative hypothesis that the SMOOTH planning system performed better than the UNCONTROLLED planning system when the performance measure was either total cost or mean hours tardy per week. When the performance measure was either total inventory or the mean number of labor transfers, there was no difference in performance of either planning system.

5.4.11.2 SUMMARY OF HYPOTHESES 2 AND 3

Hypotheses 2 and 3 investigated whether the use of LABOR information in the ORR rule would improve performance of the ORR rule with the SMOOTH and UNCONTROLLED planning systems respectively. When the performance measure was mean hours tardy per week, this was supported if the SMOOTH planning system was used. There was no support for the alternative hypothesis

with any other performance measure. However, there was a strong indication that treatments using the MLC ORR rule perform better than treatments using other ORR rules for all the performance measures except the number of labor transfers.

5.4.11.3 SUMMARY OF HYPOTHESES 4 AND 5

Hypotheses 4 and 5 that the time phasing of the work load by the ORR rule (finite loading of the shop) would improve performance with the SMOOTH and UNCONTROLLED planning systems respectively was supported when the performance measure was total cost, tardy hours or total inventory. When the performance measure was the mean hours tardy per week, the alternative was supported only for treatments using the SMOOTH planning system. The null hypothesis, H_o, that finite loading does not improve performance was supported when the UNCONTROLLED planning system was used. The null hypothesis, H_o, that there was no significant difference in the number of labor transfers between the treatments using MLC and those without MLC was accepted.

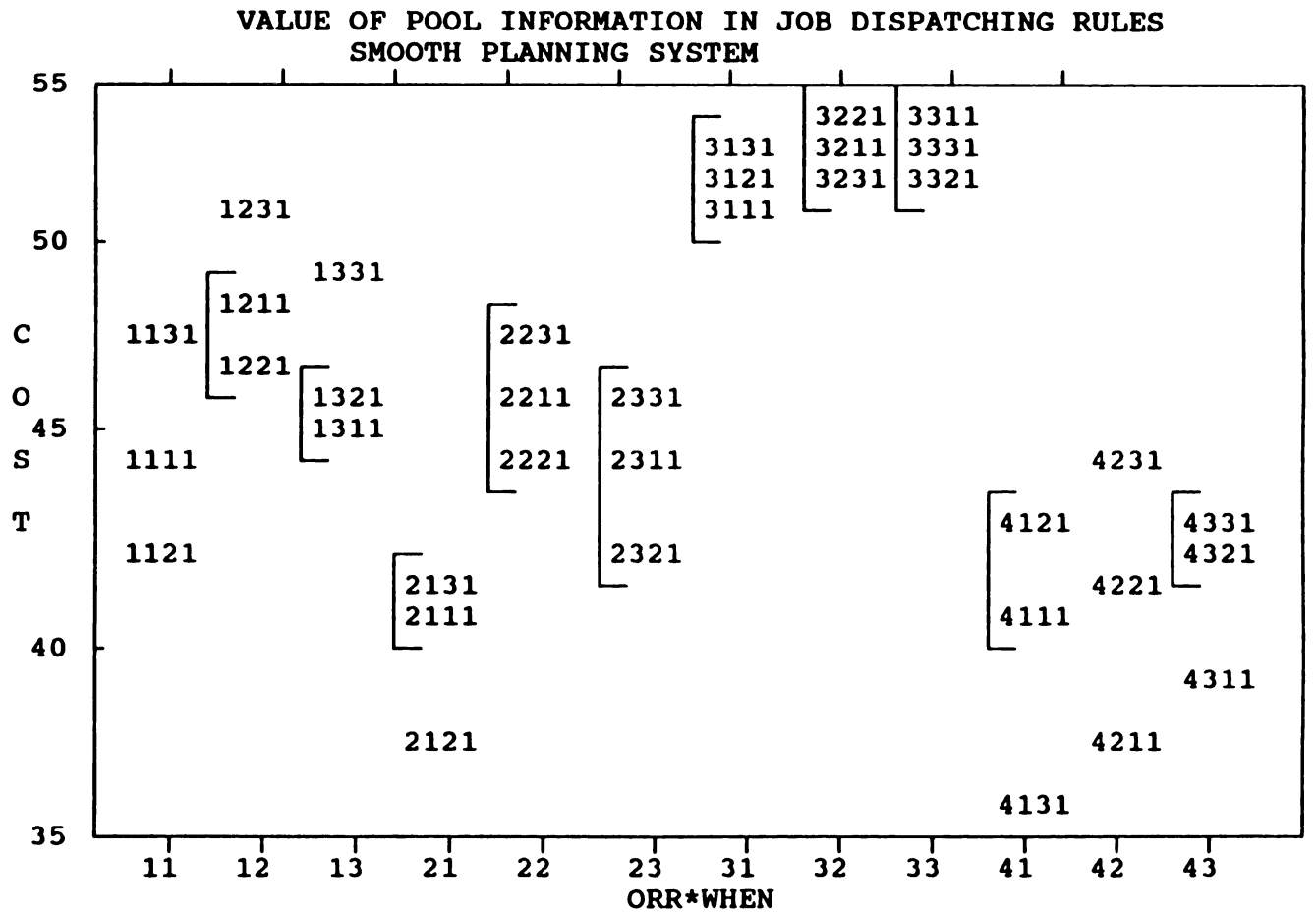
5.4.11.4 SUMMARY OF HYPOTHESIS 6

Hypothesis 6 investigated whether ORR rules could result in deterioration of shop performance. When the performance measure was either total cost or the number of labor transfers, H_o was accepted. Sometimes the use of the ORR rule caused system performance to deteriorate. When the performance measure was either the mean hours tardy or the total inventory, H_o was rejected. There

HYDROGRAPH

FIGURE 5-24

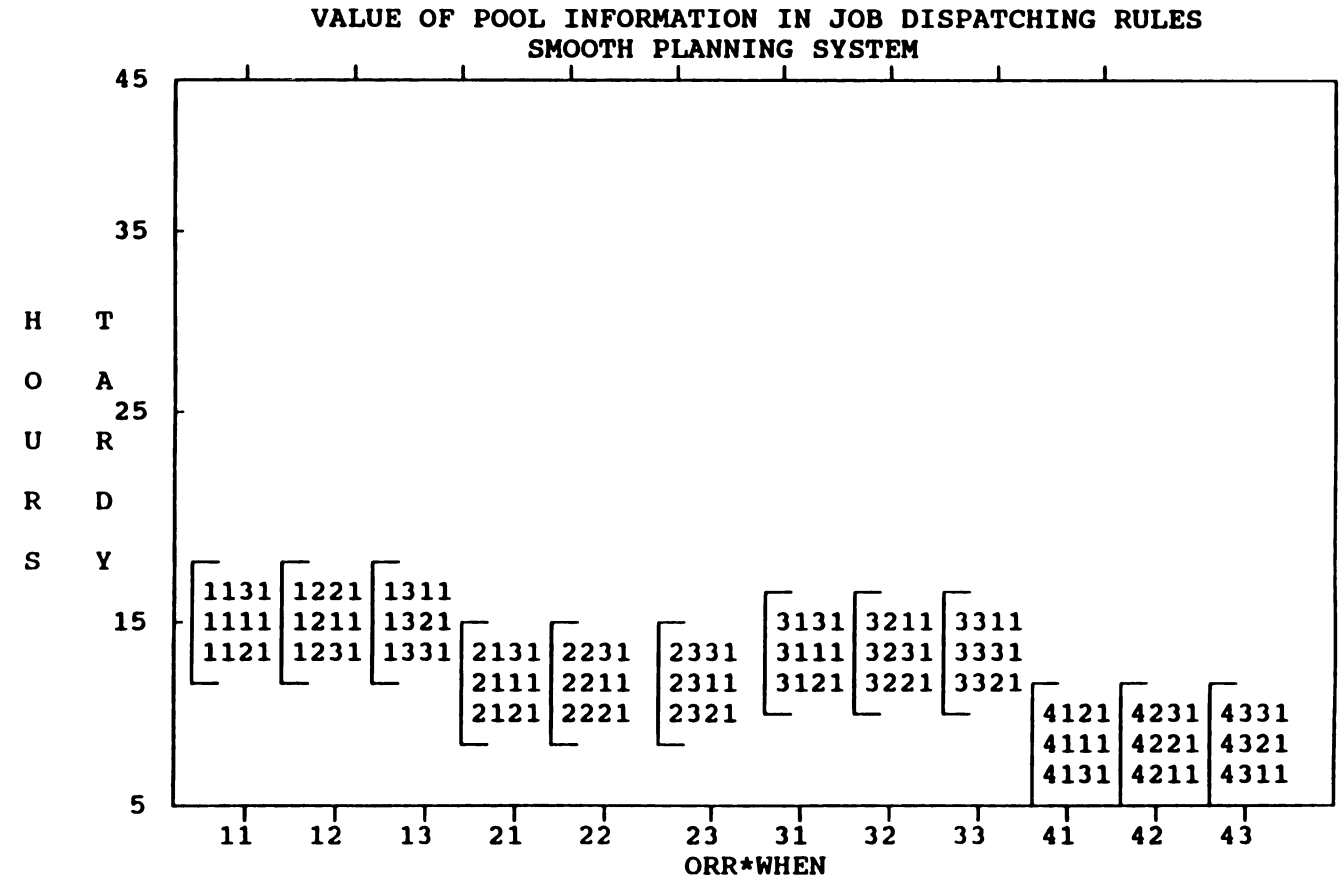
FIGURE 5-24
HYPOTHESIS 7: TOTAL COST



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-25
HYPOTHESIS 7: HOURS TARDY/WEEK

FIGURE 5-25
HYPOTHESIS 7: HOURS TARDY/WEEK

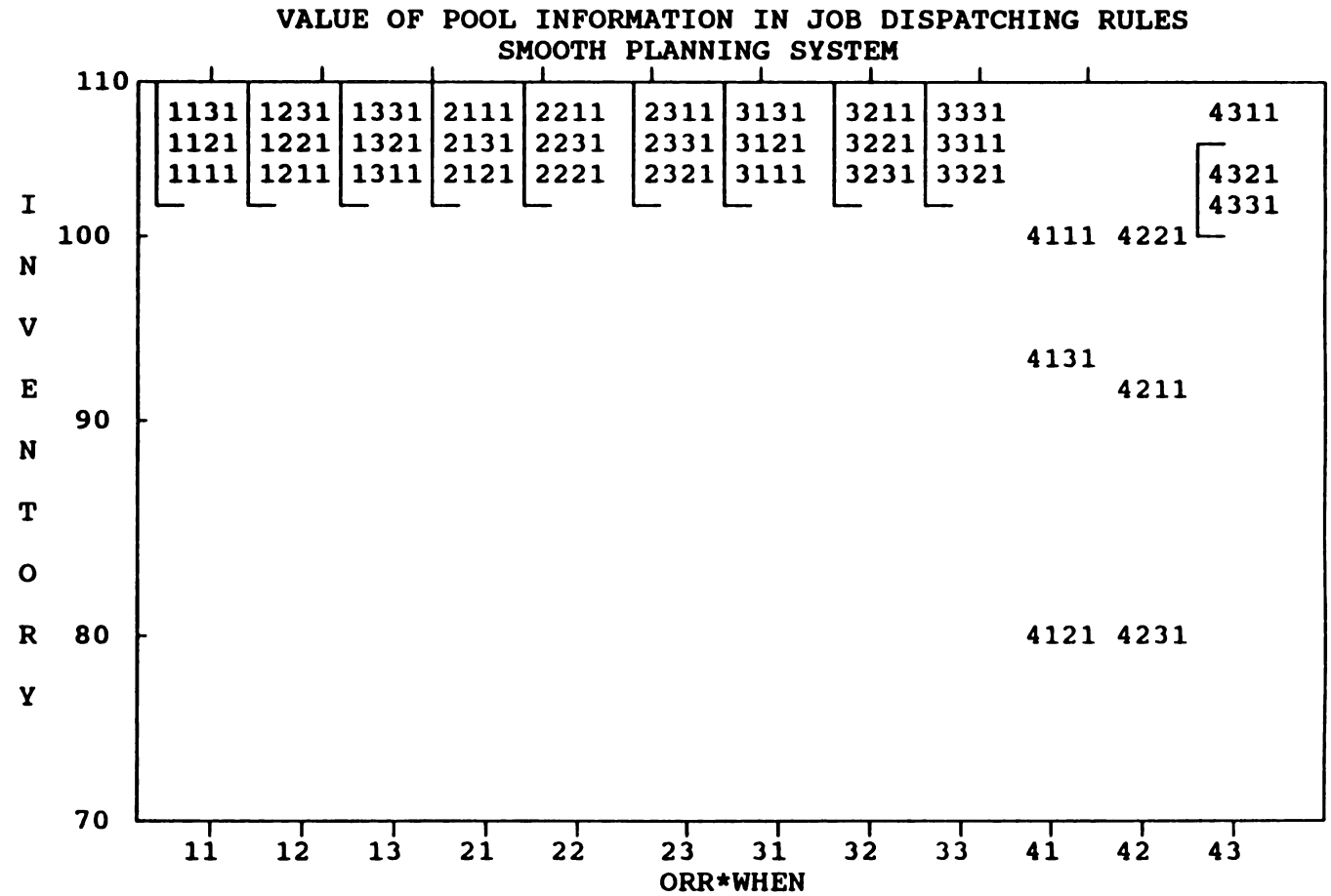


TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-26
HYPOTHESIS 7: TOTAL INVENTORY

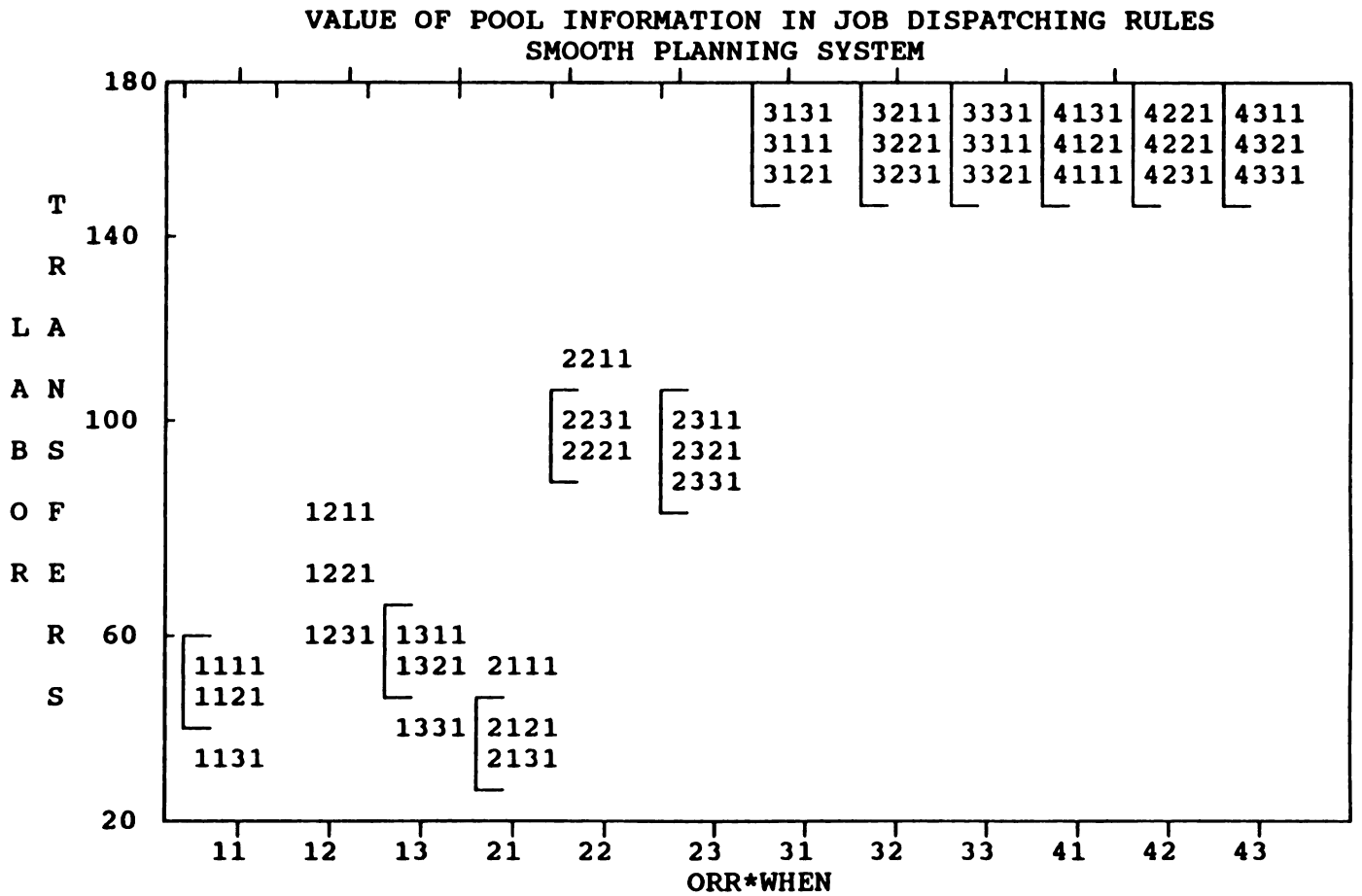
VALUE OF POOL INFORMATION IN JOB DISPATCHING

FIGURE 5-26
HYPOTHESIS 7: TOTAL INVENTORY



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

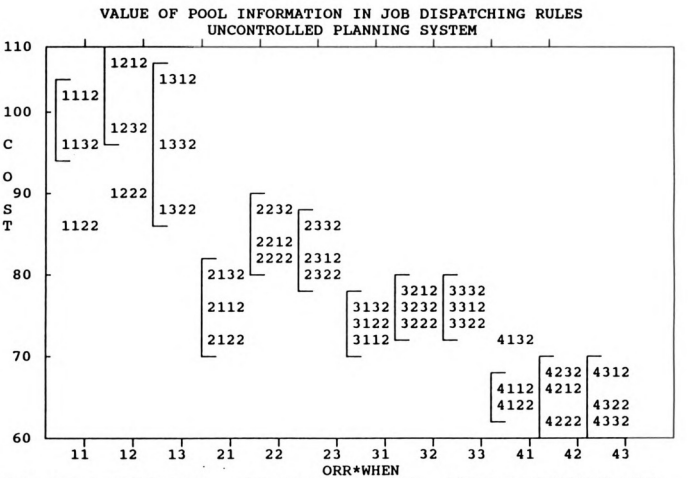
FIGURE 5-27
HYPOTHESIS 7: LABOR TRANSFERS



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

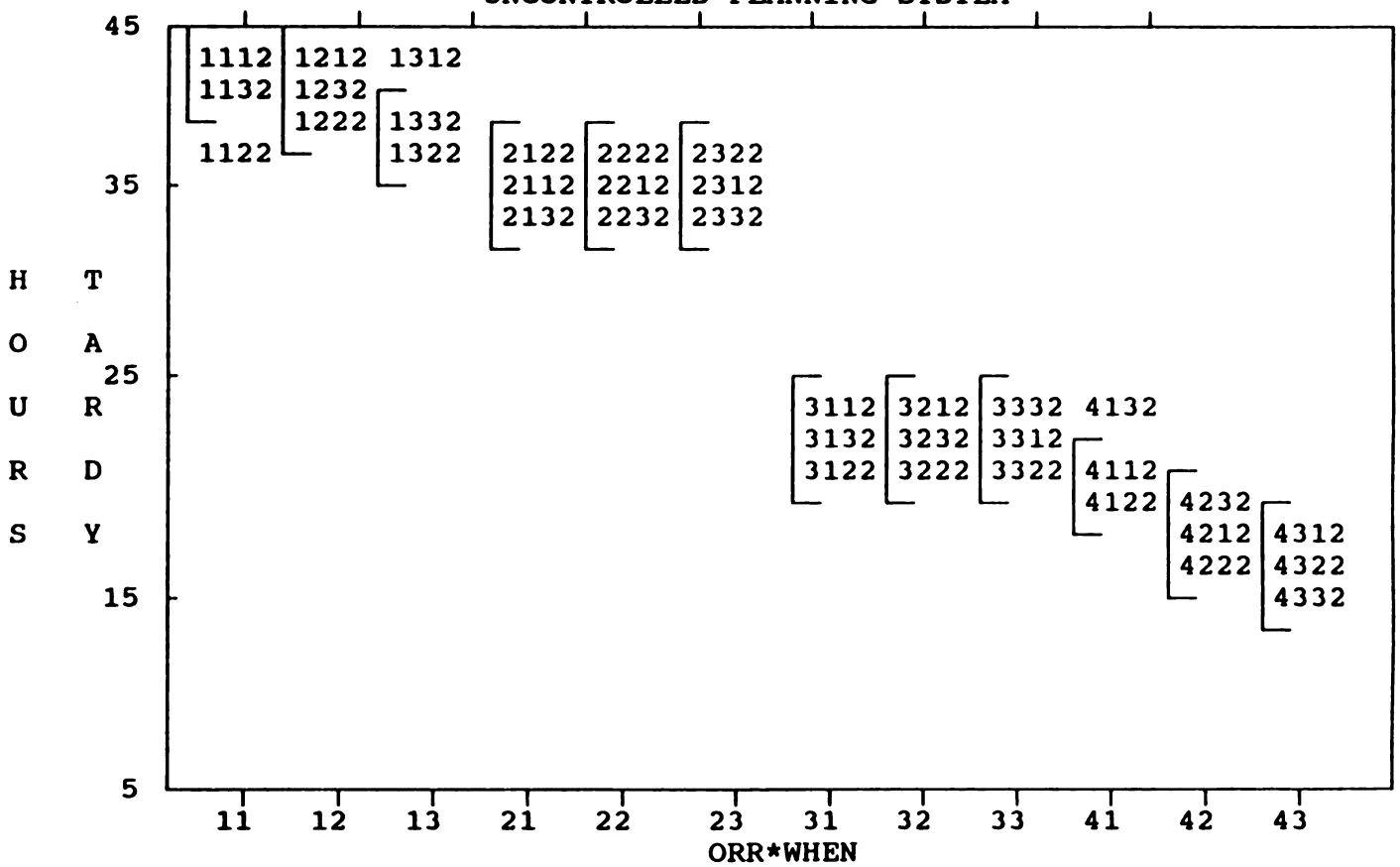
FIGURE 5-28
HYPOTHESIS 8: TOTAL COST

FIGURE 5-28
HYPOTHESIS 8: TOTAL COST



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

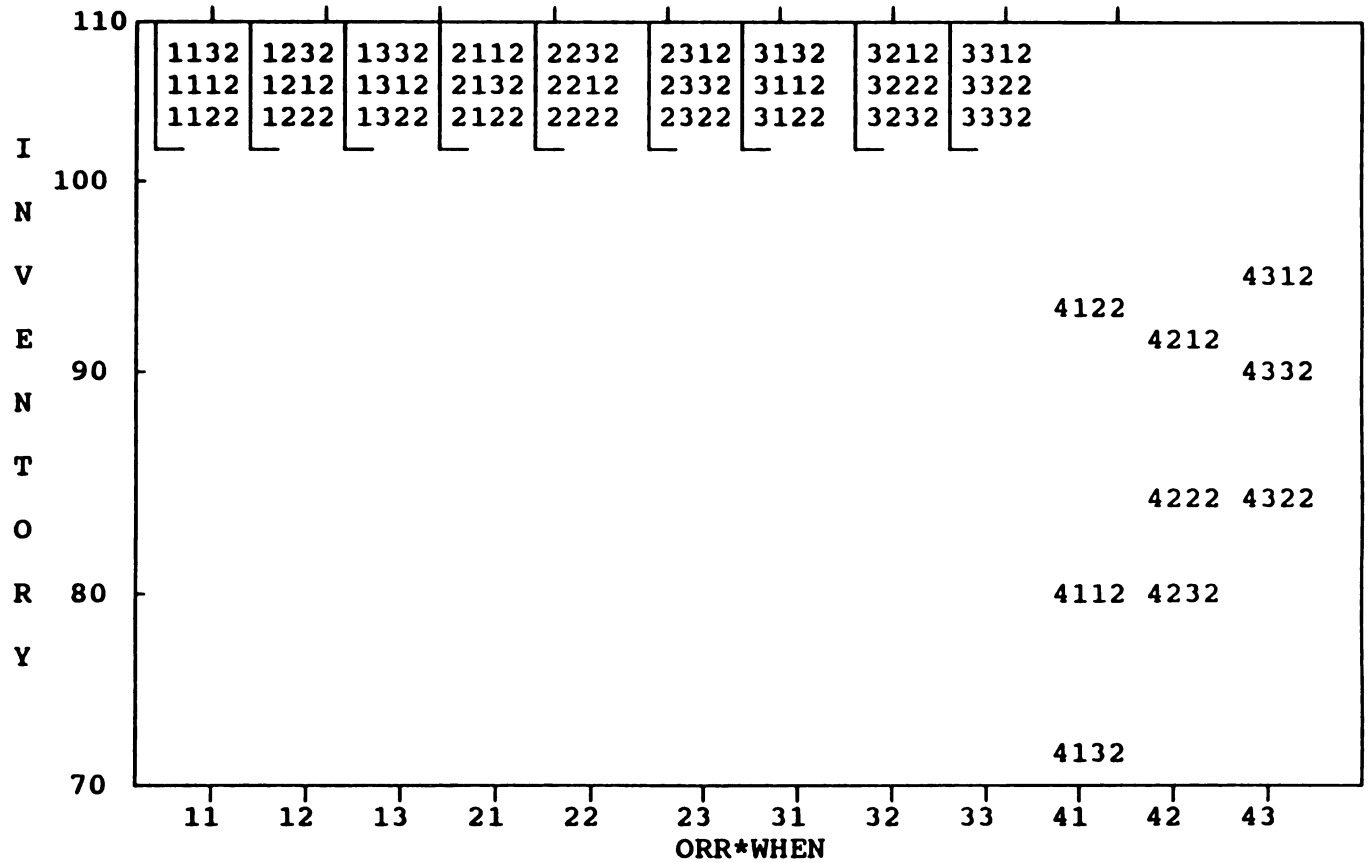
FIGURE 5-29
HYPOTHESIS 8: HOURS TARDY/WEEK
VALUE OF POOL INFORMATION IN JOB DISPATCHING RULES
UNCONTROLLED PLANNING SYSTEM



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-10 Hydrothermal system.

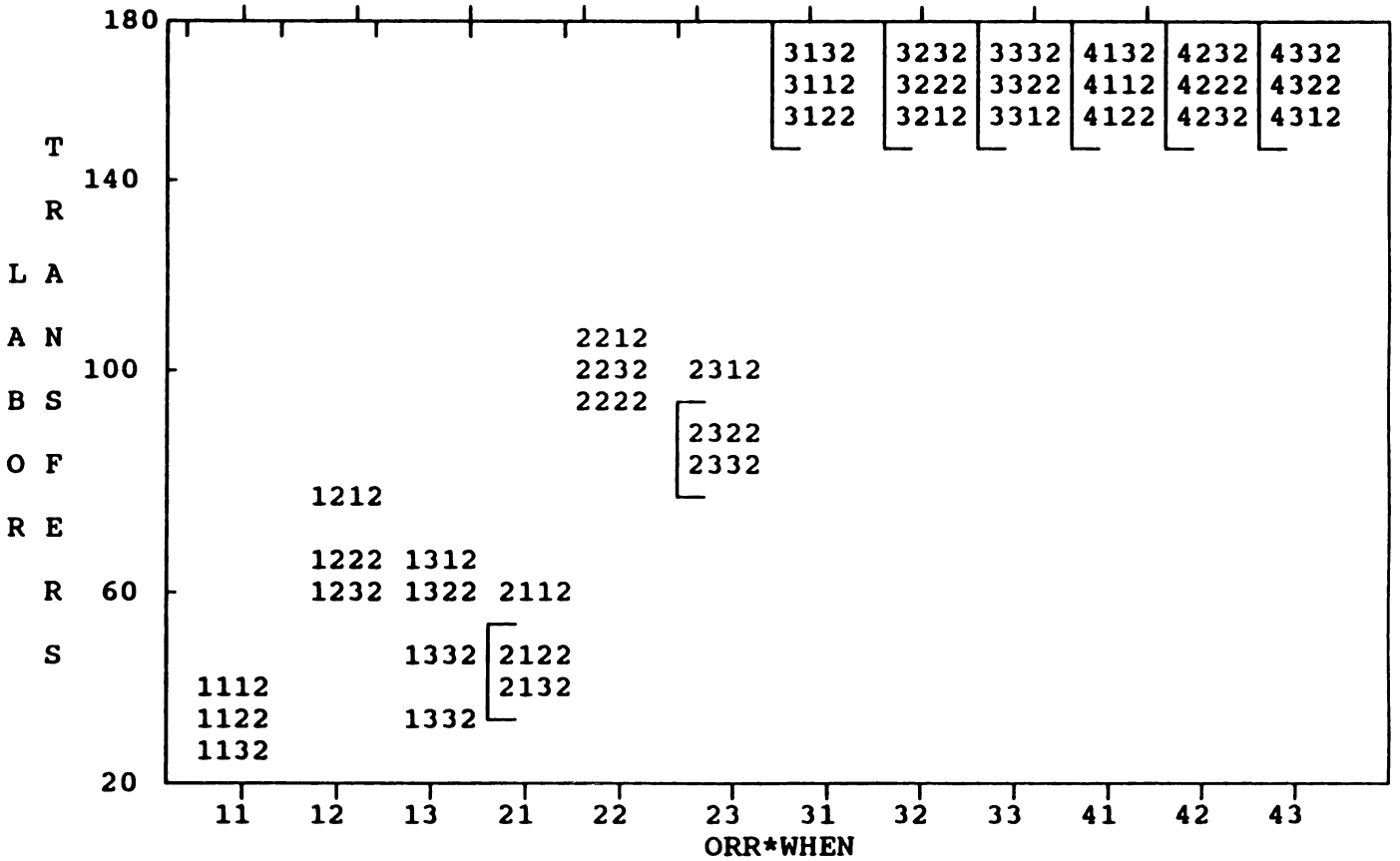
FIGURE 5-30
HYPOTHESIS 8: TOTAL INVENTORY
VALUE OF POOL INFORMATION IN JOB DISPATCHING RULES
UNCONTROLLED PLANNING SYSTEM



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

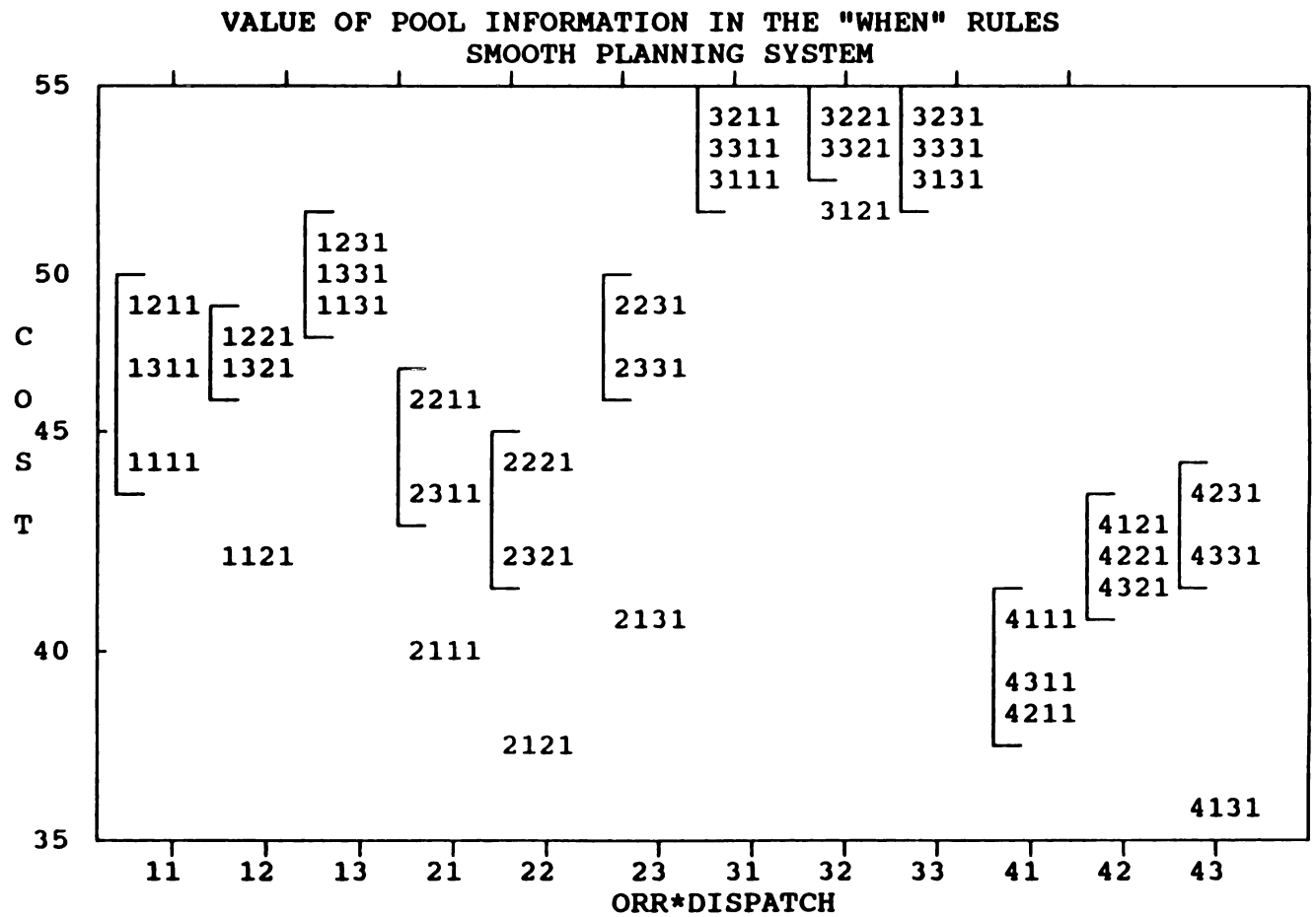
FIGURE 5-11

FIGURE 5-31
HYPOTHESIS 8: LABOR TRANSFERS
VALUE OF POOL INFORMATION IN JOB DISPATCHING RULES
UNCONTROLLED PLANNING SYSTEM



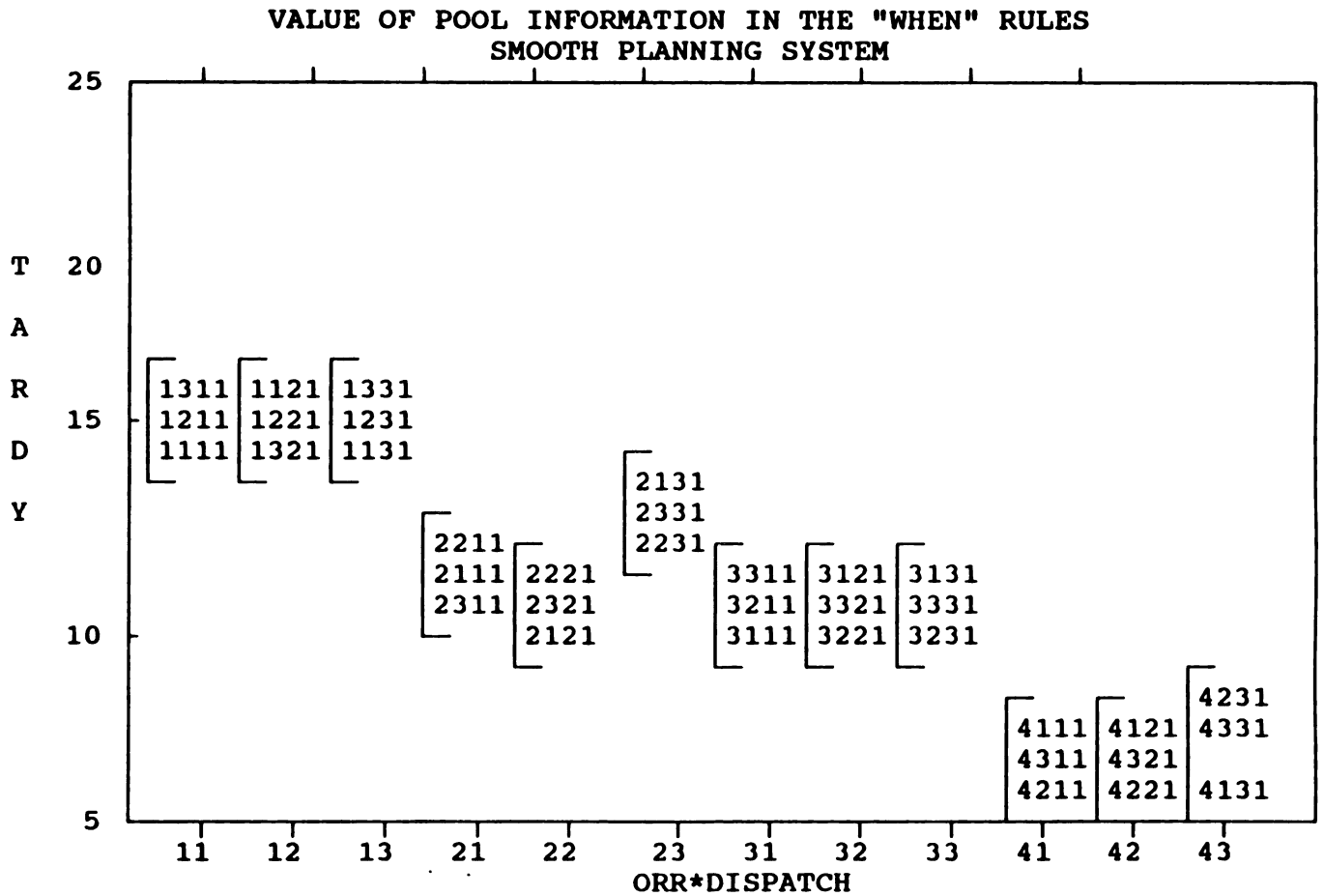
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-32
HYPOTHESIS 9: TOTAL COST



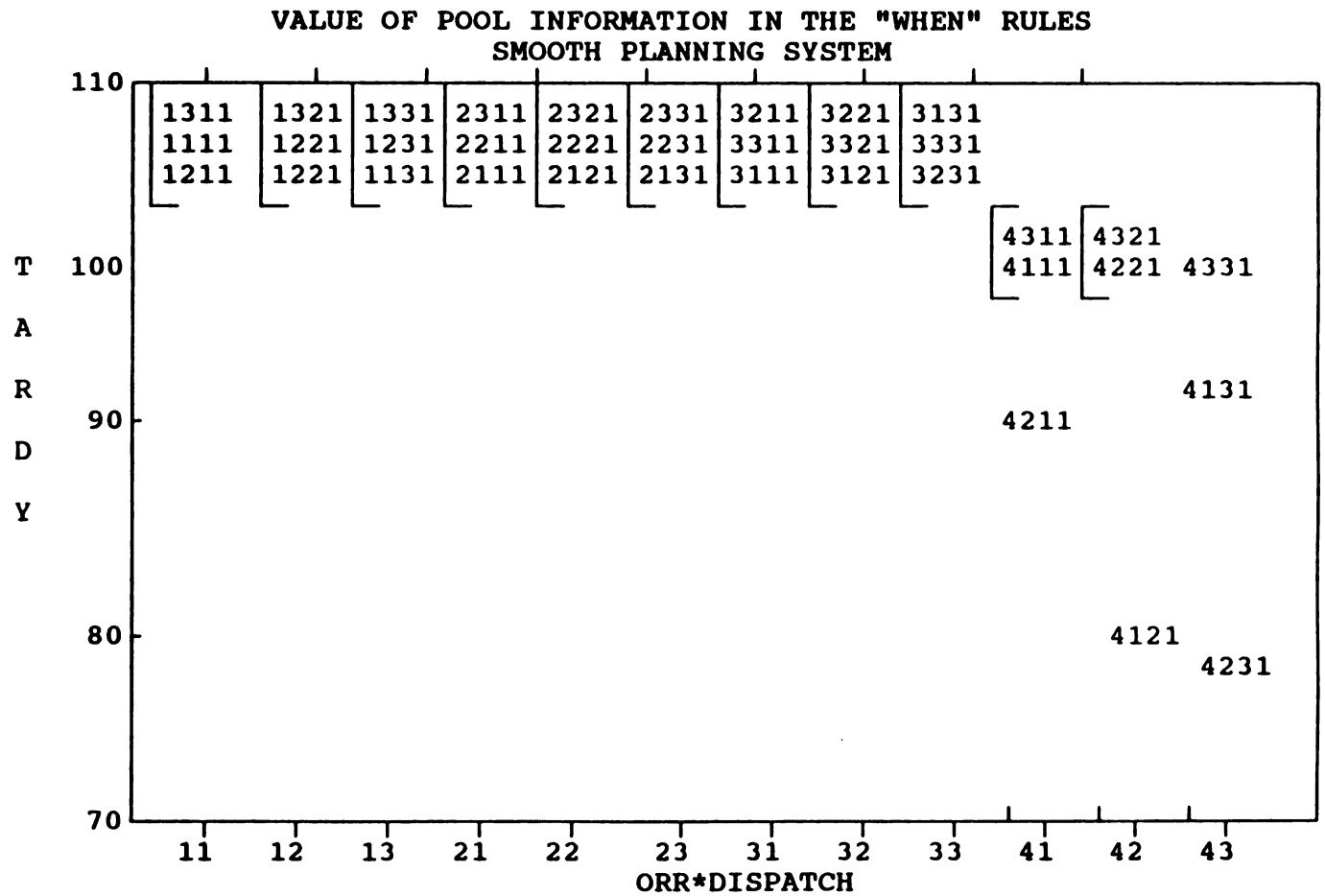
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-33
HYPOTHESIS 9: HOURS TARDY PER WEEK



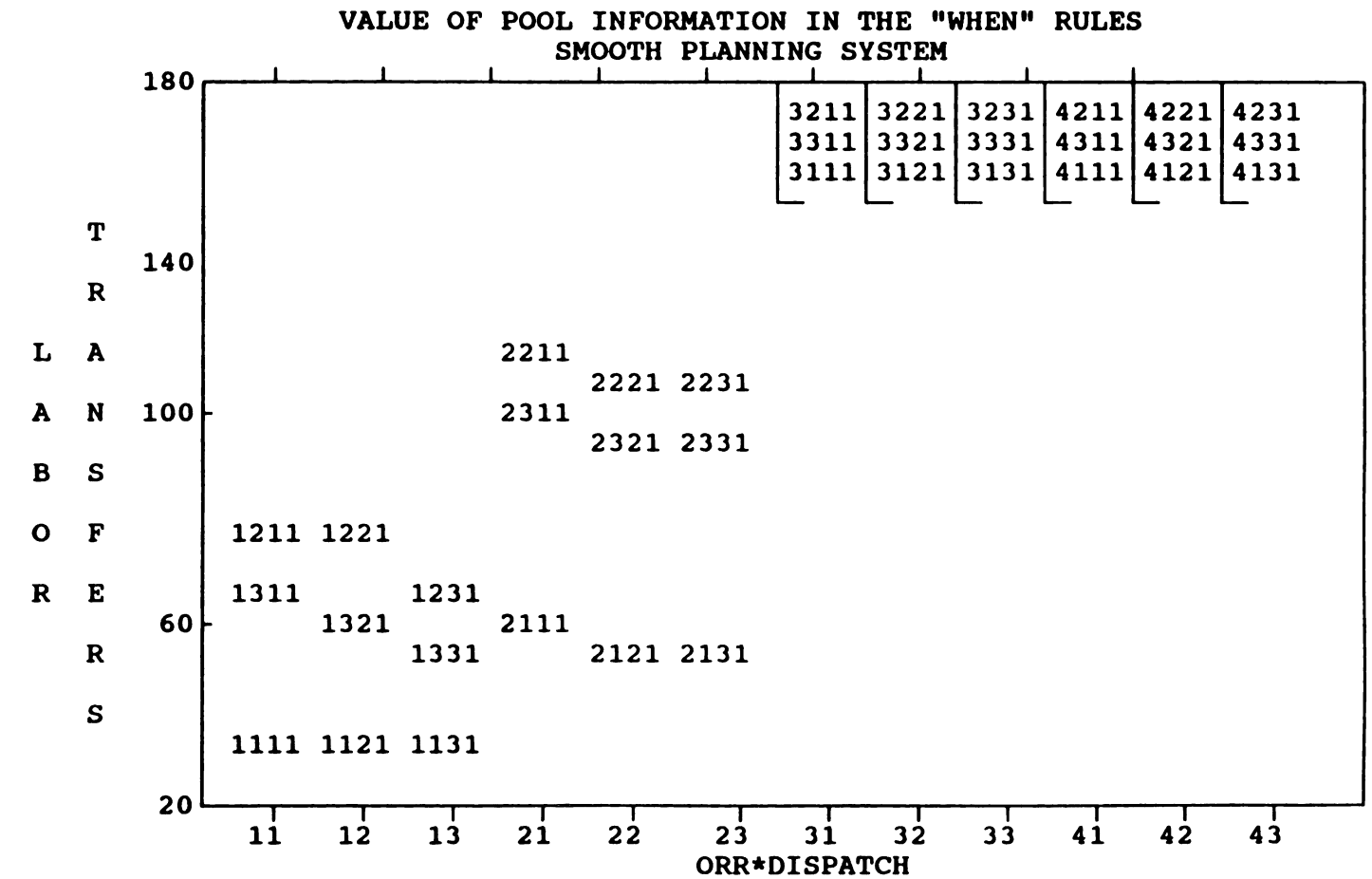
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-34
HYPOTHESIS 9: TOTAL INVENTORY



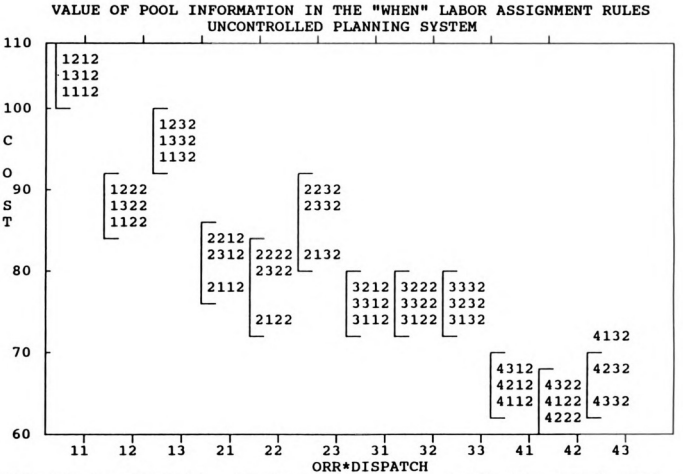
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-35
HYPOTHESIS 9: LABOR TRANSFERS



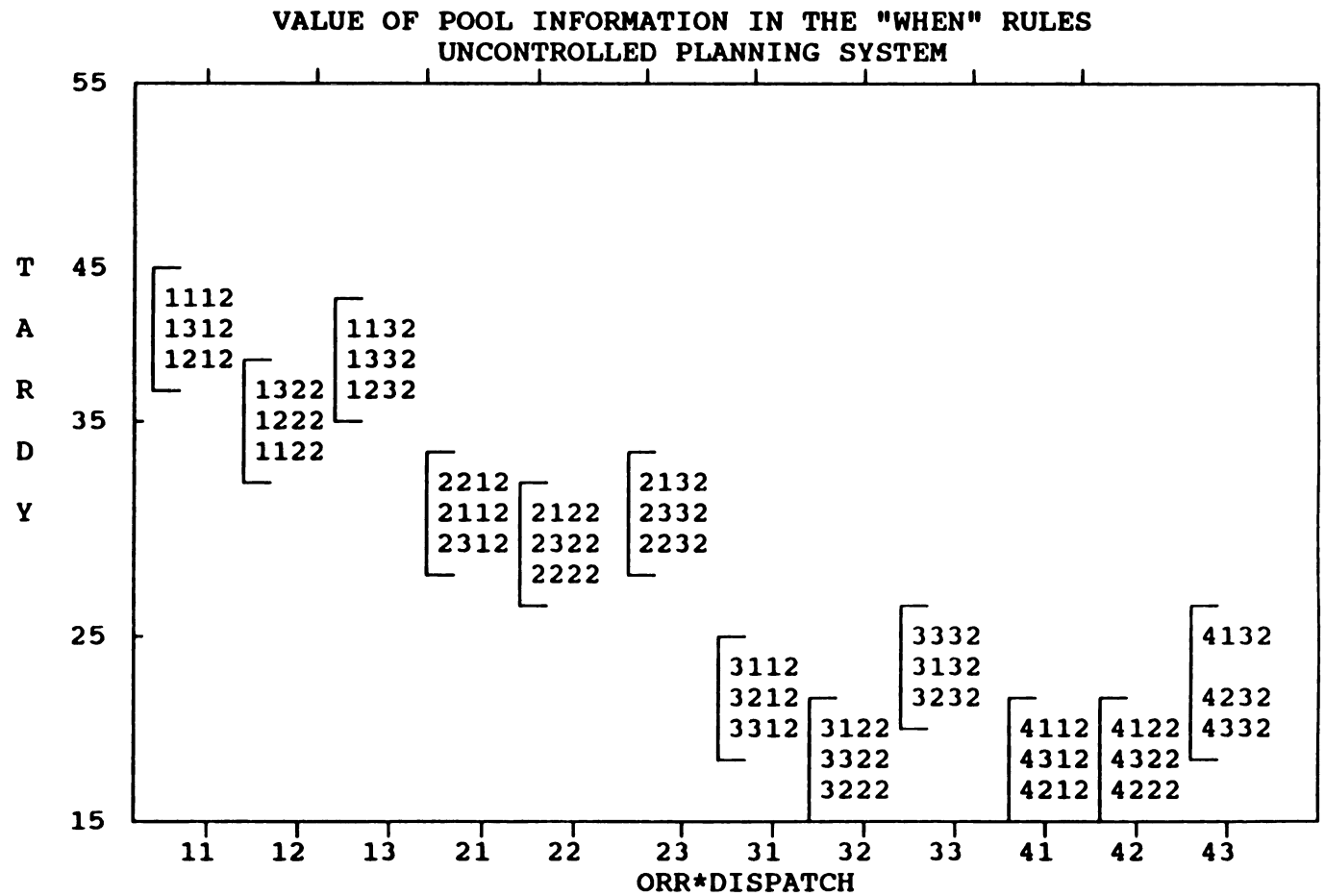
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-36
HYPOTHESIS 10: TOTAL COST



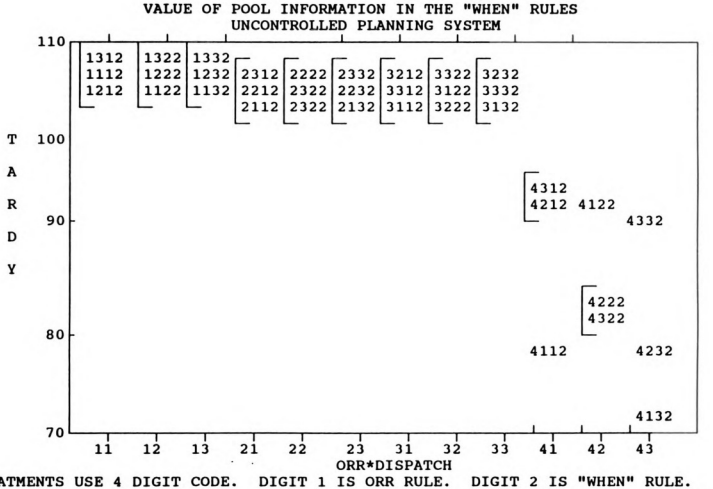
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-37
HYPOTHESIS 10: HOURS TARDY PER WEEK



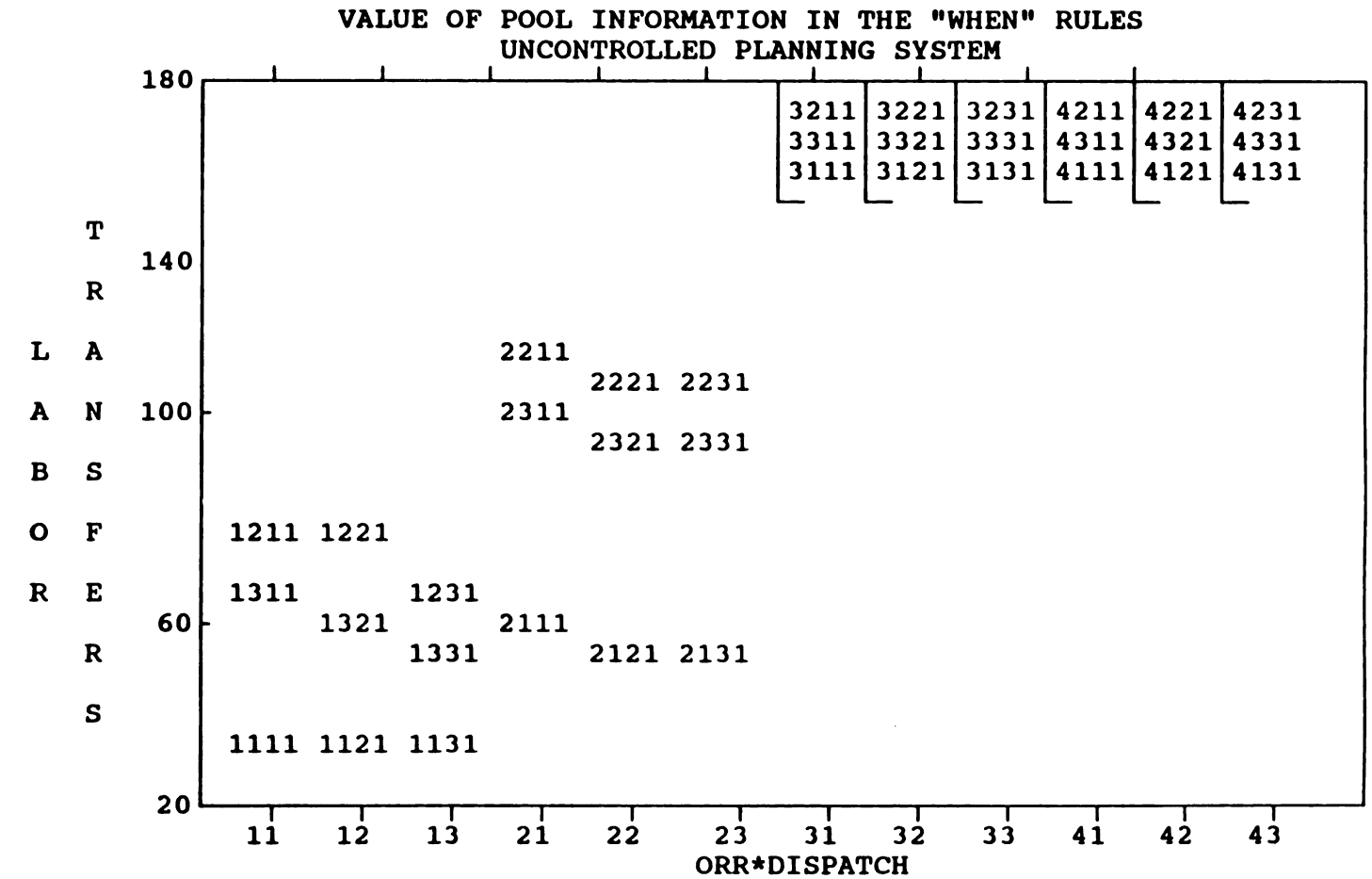
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-38
HYPOTHESIS 10: TOTAL INVENTORY



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-39
HYPOTHESIS 10: LABOR TRANSFERS



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

TABLE 5-25
SUMMARY OF A PRIORI HYPOTHESES

	NULL HYPOTHESIS ACCEPTED OR REJECTED BASED ON MULTIPLE COMPARISONS PERFORMANCE MEASURES			
	RECIPROCAL MEAN TOTAL COST	RECIPROCAL MEAN HOURS TARDY/WEEK	RECIPROCAL TOTAL INVENTORY	RECIPROCAL MEAN LABOR TRANSFERS
HYPOTHESIS 1	REJECT	REJECT	ACCEPT	ACCEPT
HYPOTHESIS 2	ACCEPT	ACCEPT	ACCEPT	ACCEPT
HYPOTHESIS 3	ACCEPT	ACCEPT	ACCEPT	ACCEPT
HYPOTHESIS 4	REJECT	REJECT	REJECT	ACCEPT
HYPOTHESIS 5	REJECT	REJECT	REJECT	ACCEPT
HYPOTHESIS 6	ACCEPT	REJECT	REJECT	ACCEPT
HYPOTHESIS 7	ACCEPT	ACCEPT	ACCEPT	ACCEPT
HYPOTHESIS 8	ACCEPT	ACCEPT	ACCEPT	ACCEPT
HYPOTHESIS 9	ACCEPT	ACCEPT	ACCEPT	ACCEPT
HYPOTHESIS 10	ACCEPT	ACCEPT	ACCEPT	ACCEPT

was support for the alternative hypothesis that the use of ORR rule improved performance.

5.4.11.5 SUMMARY OF HYPOTHESES 7 AND 8

Hypotheses 7 and 8 that the use of POOL information in the job dispatch rule improved performance with the SMOOTH or UNCONTROLLED planning systems respectively was not supported for any performance measure. For the total labor transfers the use of POOL information in the job dispatching rule did make a significant difference when there was no ORR rule.

5.4.11.6 SUMMARY OF HYPOTHESES 9 AND 10

Hypotheses 9 and 10 that the use of POOL information in the "when" labor assignment rule improved performance with the SMOOTH or UNCONTROLLED planning systems respectively were not supported for any performance measure. For labor transfers when the CMS or MLC rules were used, there was never a difference in the "when" rule. For labor transfers the use of the QUE rule with the IMM and MIL ORR rules improved performance.

5.5 POST HOC ANALYSIS

The findings in this section were the result of post hoc analysis to identify patterns or trends in the data requiring further investigation. Post hoc analysis does not provide as much construct validity as a priori analysis, because the a priori hypotheses are formulated in advance before being suggested by the data

(Kerlinger, 1986, p.23). However, post hoc findings can form a strong basis for future research.

The post hoc analysis examines the significant findings of the a priori analysis. These are that:

- Treatments using the MLC ORR rules had significantly lower costs, lower tardiness and lower inventory than treatments using the CMS ORR rules.
- The SMOOTH planning system performed better than the UNCONTROLLED planning system for the total cost and mean hours tardy performance measures. However, UNCONTROLLED performed better than SMOOTH for the total inventory and labor transfer performance measures.
- CMS ORR rules had higher costs than immediate release of jobs with the SMOOTH planning system, but performed better than immediate release of jobs when the performance measures were mean hours tardy or total inventory.
- The importance of the job dispatching rules is reduced when ORR rules are used.
- The importance of the "when" labor assignment rule is reduced when ORR rules are used.

The purpose of the post hoc analysis is to gain more insight into each of these findings. Each finding is examined separately below. The important results of this post hoc analysis are summarized in Table 5-26.

The post hoc analysis was conducted with multiple pair-wise comparisons of the total cost, the component cost measures, and eight of the secondary performance measures listed earlier. The multiple comparisons of these performance measures are shown in Figures 5-40 to 5-47.

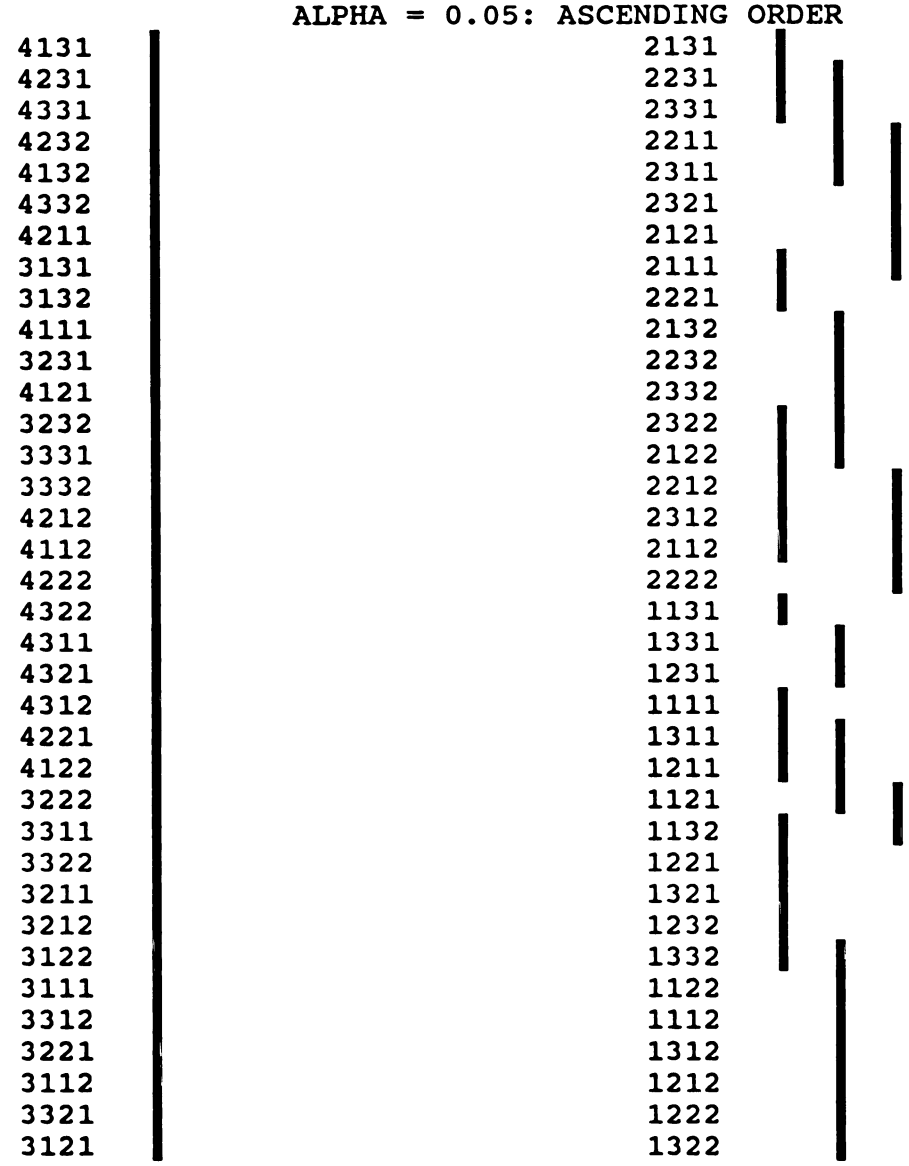
5.5.1 CMS AND MLC ORR RULES

The left column of Table 5-26 lists all performance measures that were tested. The factors which were analyzed are shown across the top. The results of each test are shown in the column below each factor. As shown, the treatments using the CMS rule never performed better than treatments using the MLC rule. At best, treatments using CMS were equal to treatments using MLC.

The MLC rule and the CMS rule share the same trigger mechanism, but have different job selection mechanisms. There were two post hoc hypotheses. They were:

- If both the CMS and MLC ORR rules performed equally well on a performance measure, it could be attributed to their trigger mechanism.
- If treatments using MLC performed better than the treatments using CMS, it could be attributed to the job selection mechanism.

FIGURE 5-40
BONFERRONI COMPARISONS OF FLOW TIME[1]
ALPHA = 0.05: ASCENDING ORDER

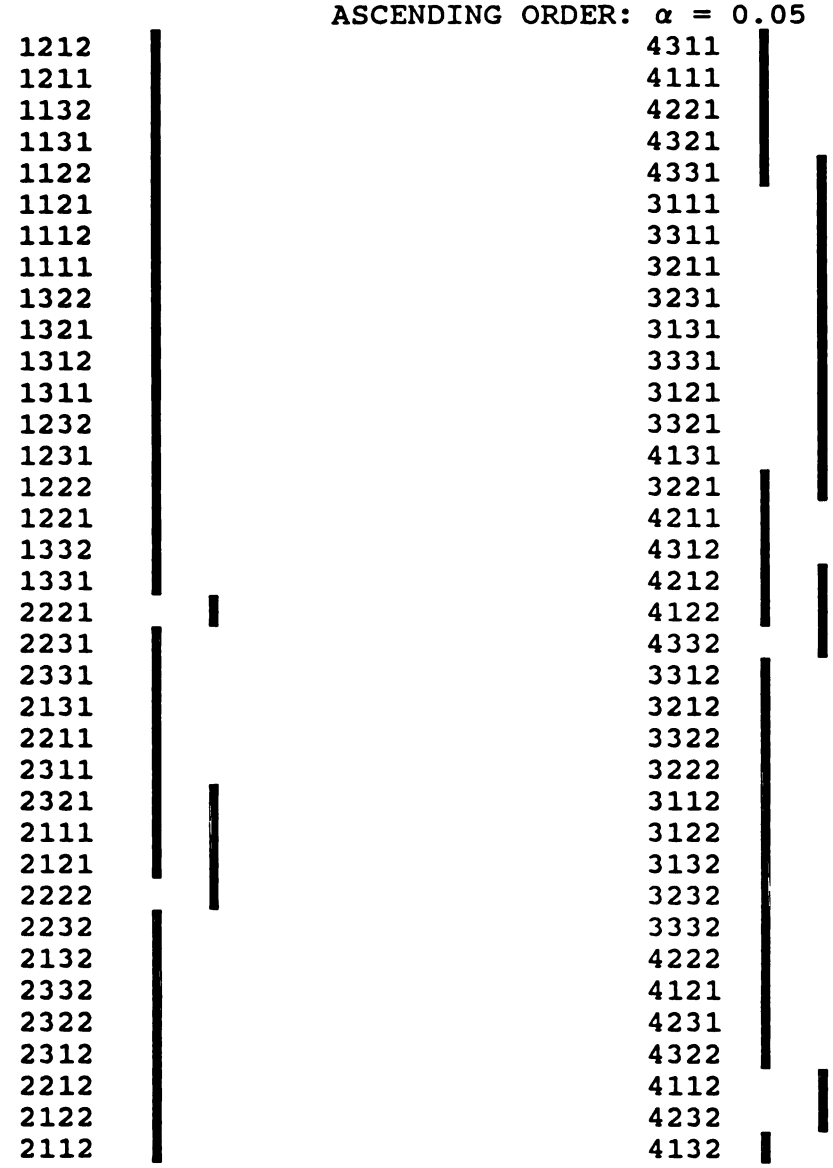


LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

---

[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-41
BONFERRONI COMPARISONS OF TIME IN POOL[1]
ASCENDING ORDER: $\alpha = 0.05$

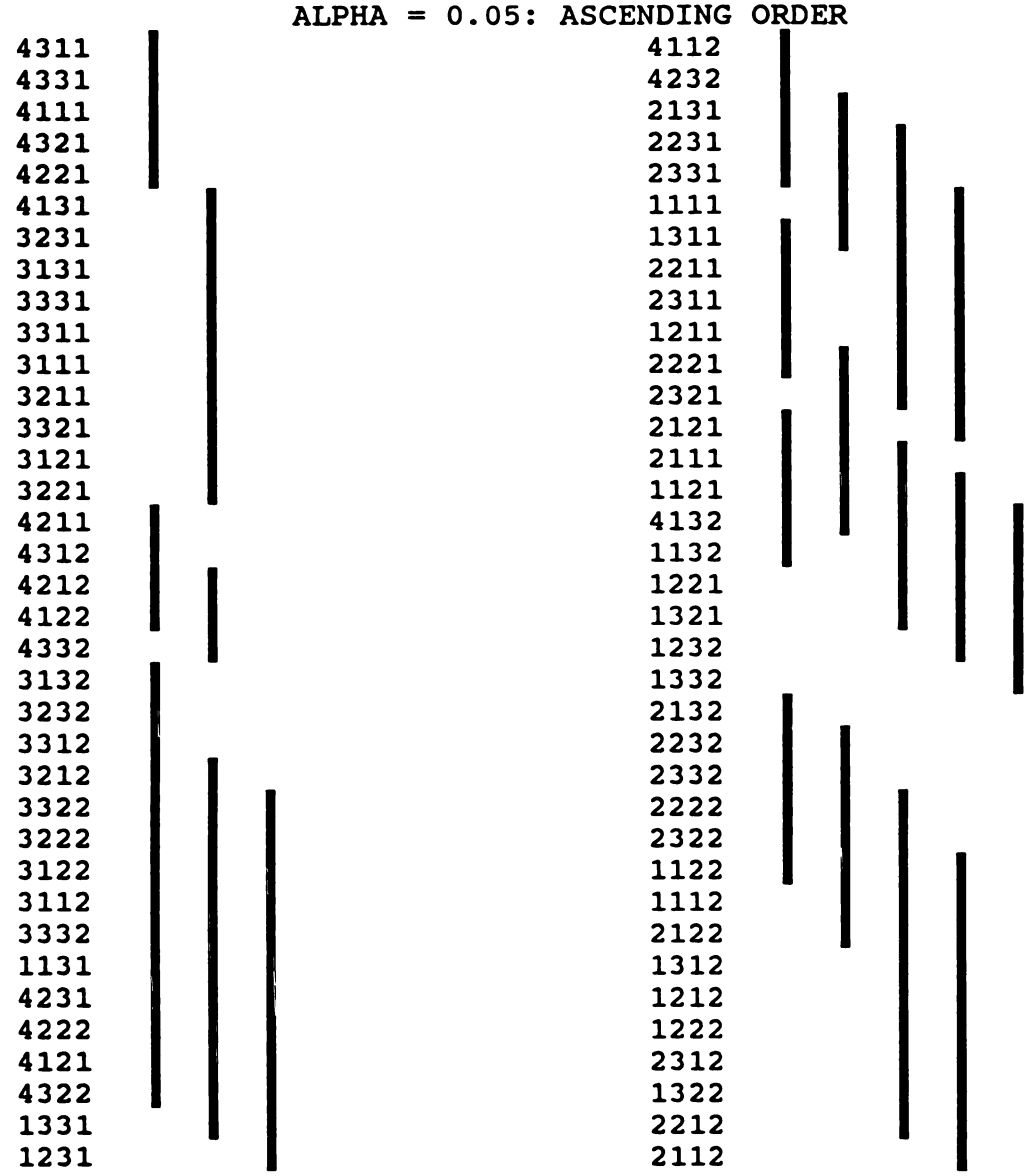


LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED

---

[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-42
BONFERRONI COMPARISONS OF TIME IN SYSTEM[1]
ALPHA = 0.05: ASCENDING ORDER



LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

---

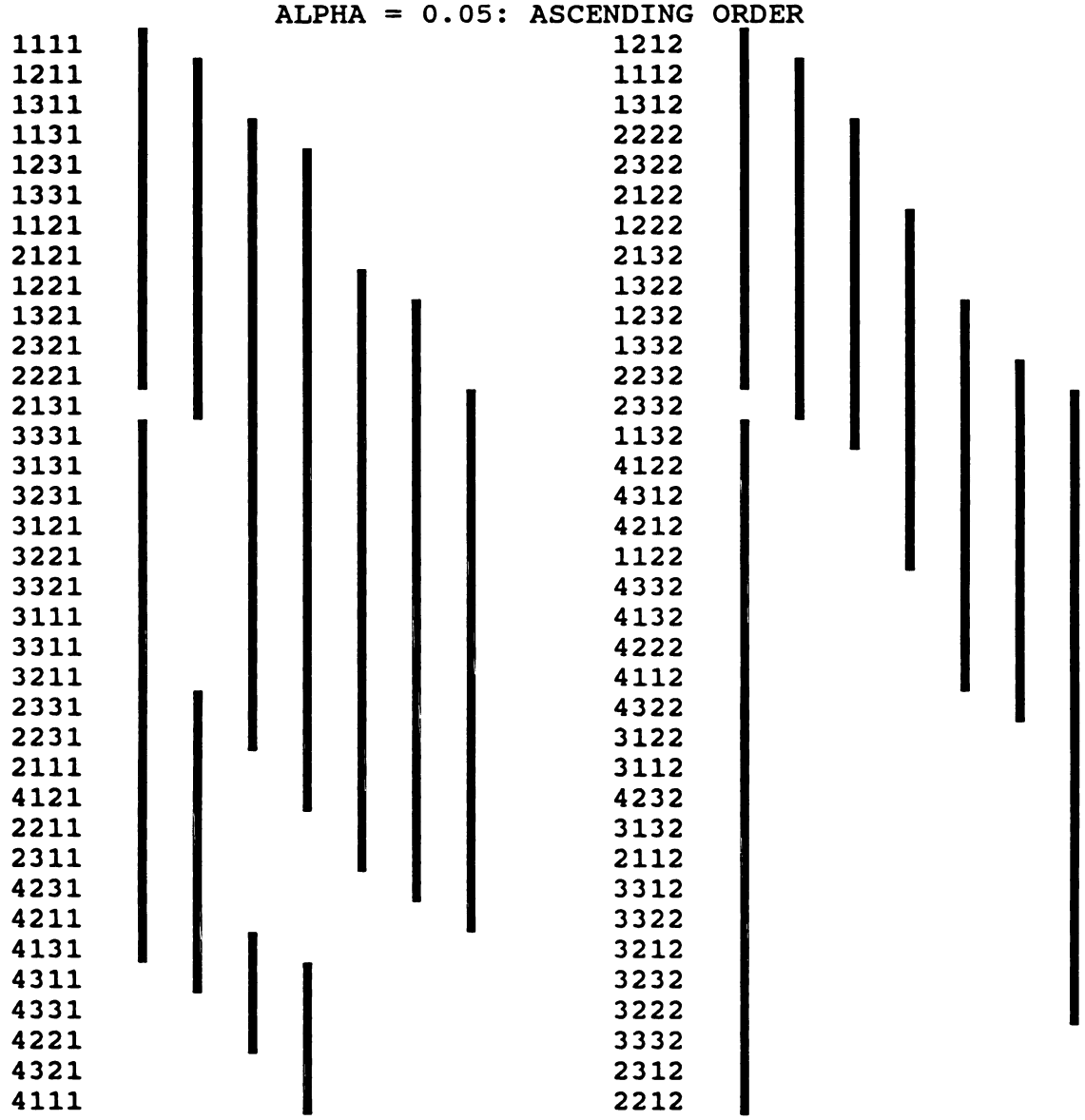
[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

111
121
131
113
123
133
112
212
122
132
232
222
213
333
313
323
312
322
332
311
331
321
233
223
211
412
221
231
423
421
413
431
433
422
432
411

LEGEN
DIGIT
DIGIT
DIGIT
DIGIT
DIGIT
DIGIT

1. CO
PROCED

FIGURE 5-43
BONFERRONI COMPARISONS OF HOURS TARDY PER JOB[1]
ALPHA = 0.05: ASCENDING ORDER



LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

---

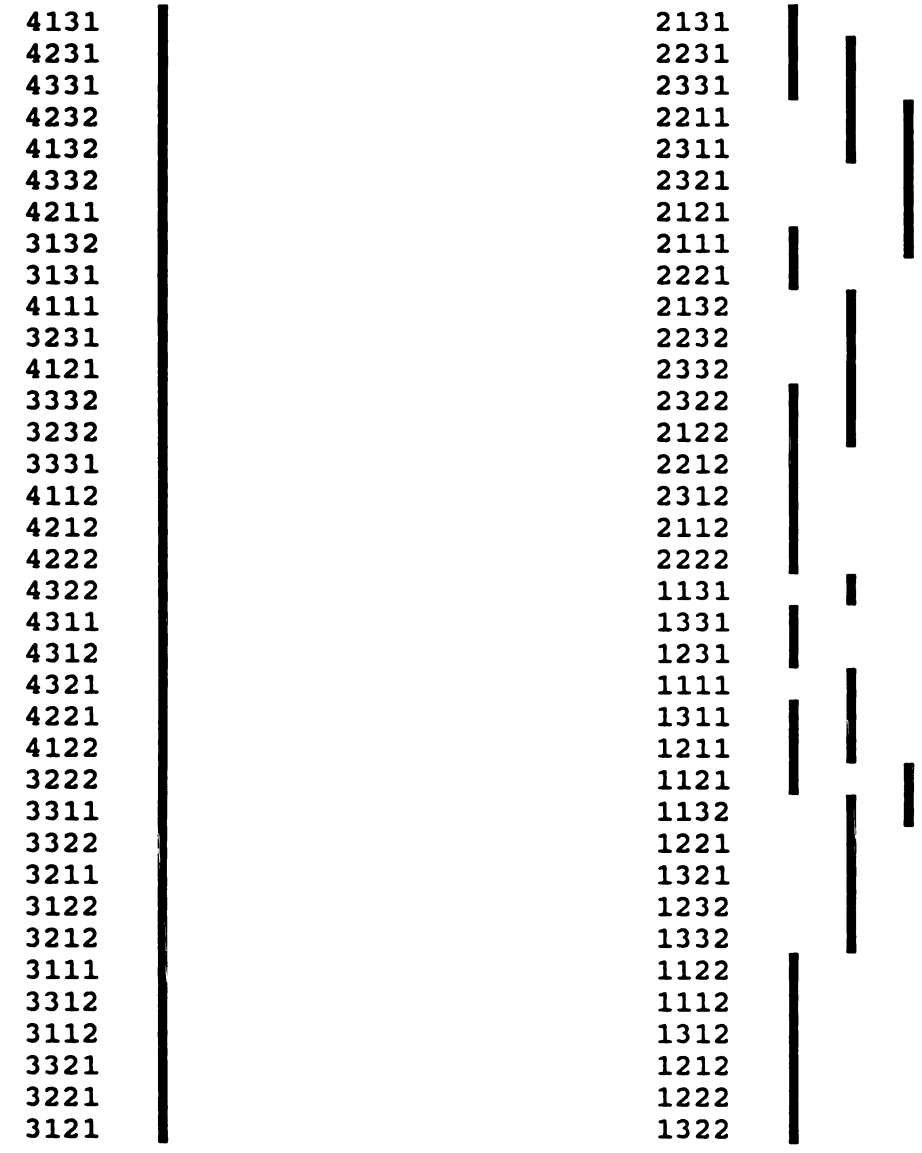
[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

4131
4231
4331
4232
4132
4332
4211
3132
3131
4111
3231
4121
3332
3232
3331
4112
4212
4222
4322
4311
4312
4321
4221
4122
3222
3311
3322
3211
3122
3212
3111
3312
3112
3321
3221
3121

LEGEND
DIGIT
DIGIT
DIGIT
DIGIT
DIGIT
DIGIT

1. CO
PROCEDU

FIGURE 5-44
BONFERRONI COMPARISONS OF WAIT TIME PER JOB[1]
ASCENDING ORDER: $\alpha = 0.05$

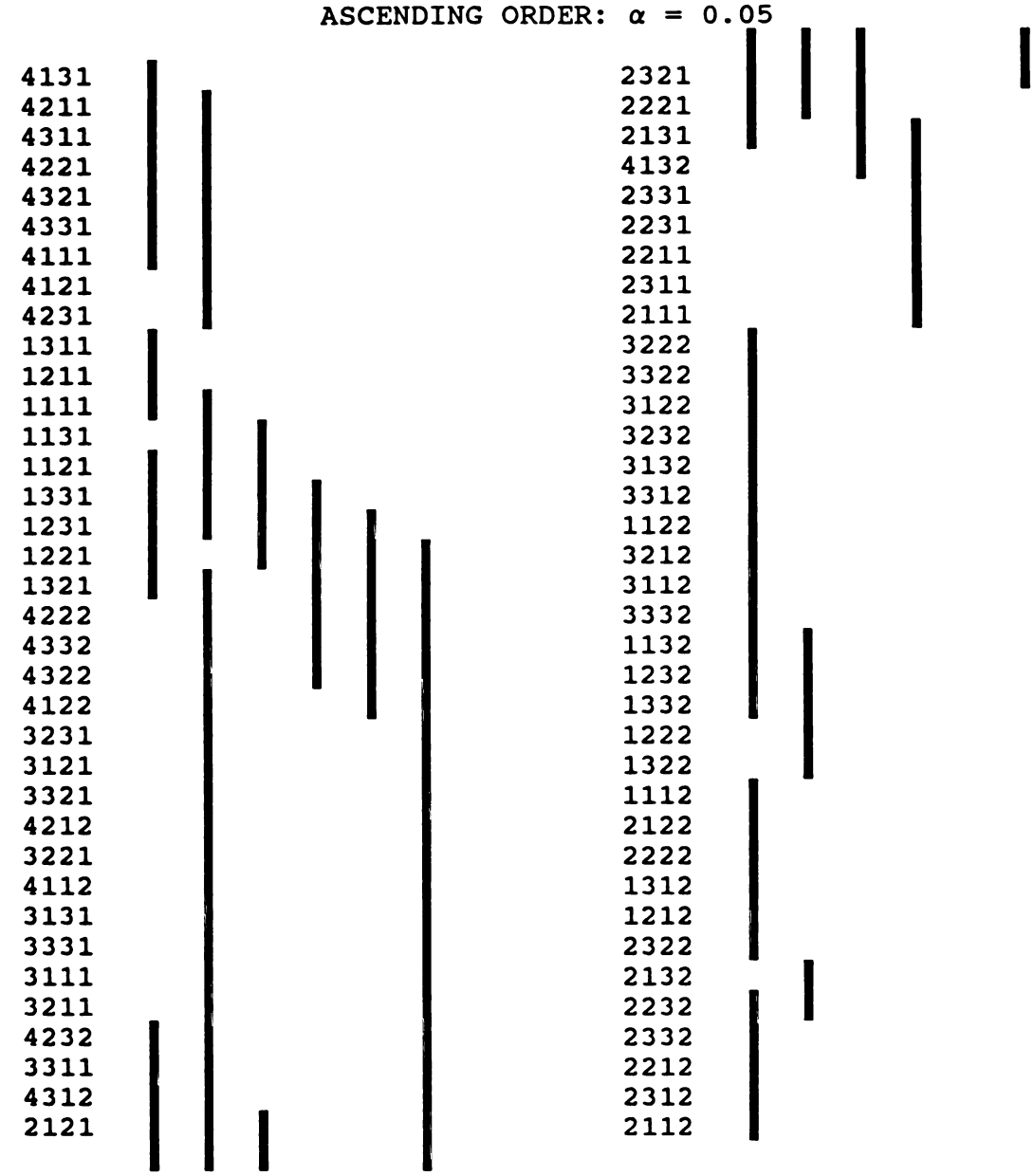


LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

---

[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-45
BONFERRONI COMPARISONS OF PERCENT OF TARDY JOBS[1]
ASCENDING ORDER: $\alpha = 0.05$

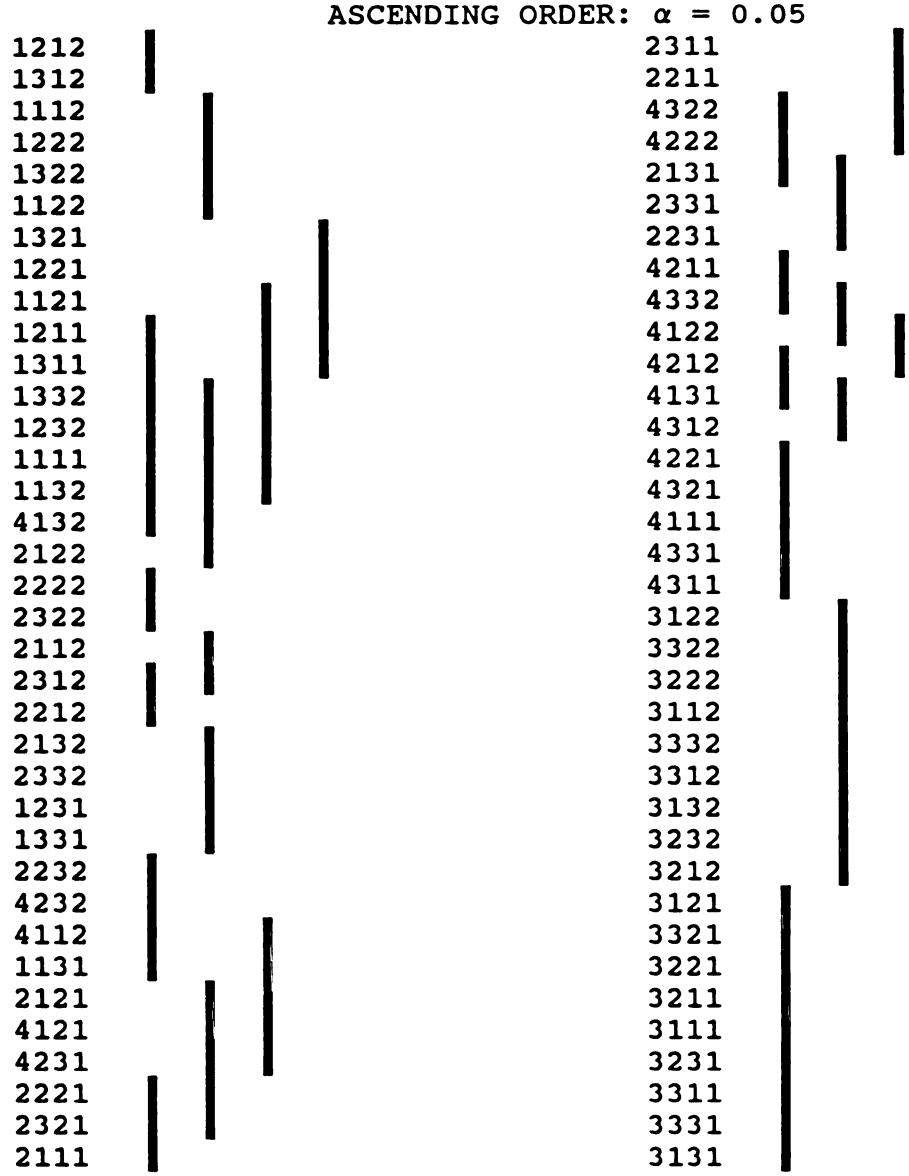


LEGEND FOR CELL MEANS
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

---

[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-46
BONFERRONI COMPARISONS OF FINISHED GOODS INVENTORY[1]
ASCENDING ORDER: $\alpha = 0.05$

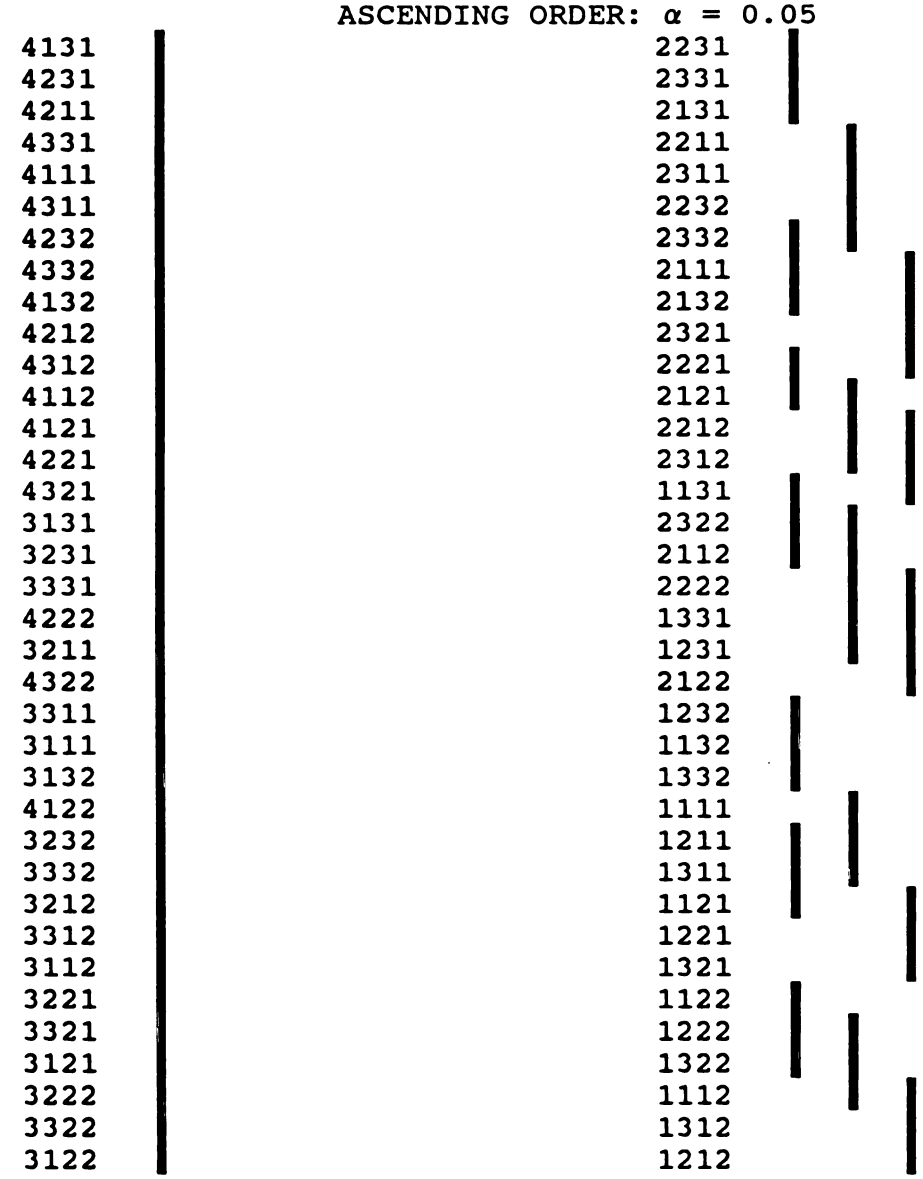


LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH,2=UNCONTROLLED)

---

[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

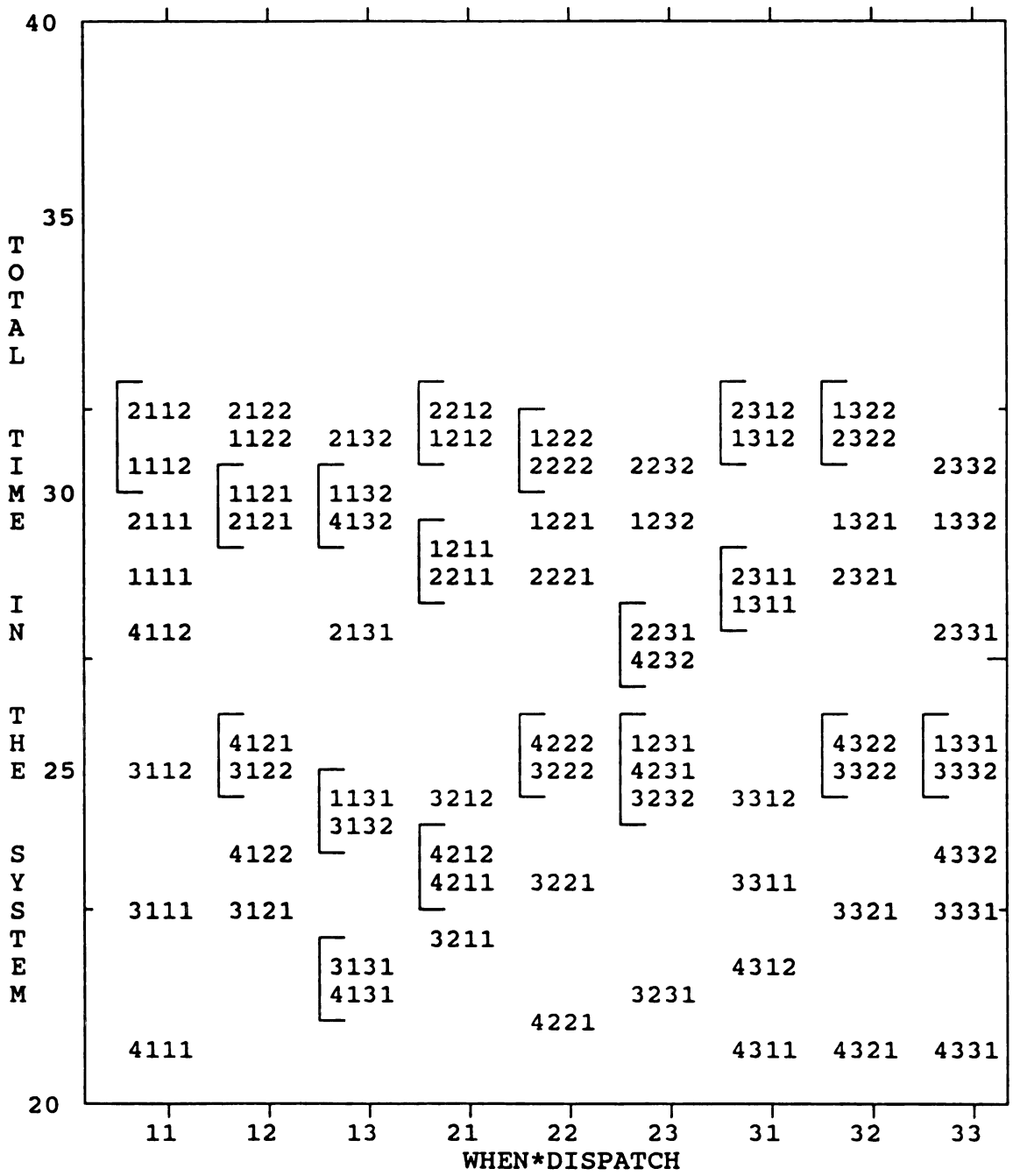
FIGURE 5-47
BONFERRONI COMPARISONS OF WIP INVENTORY[1]
ASCENDING ORDER: $\alpha = 0.05$



LEGEND FOR CELL MEANS
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)

[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03.

FIGURE 5-48
COMPARISONS OF TOTAL TIME IN SYSTEM BY ORR RULE



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE.
DIGIT 2 IS "WHEN" RULE. DIGIT 3 IS DISPATCH RULE.
DIGIT 4 IS PLANNING: (ORR, WHEN, DISPATCH, PLAN)
ORR: 1=IMM,2=MIL,3=CMS,4=MLC. WHEN: 1=QUE,2=JOB,3=CLR
DISPATCH: 1=FCFS,2=MODD,3=CMDD.

The trigger and job selection mechanisms are discussed in separate sections below.

5.5.1.1 TRIGGER MECHANISM

The trigger mechanism used by both the CMS and the MLC rules was to release work from the pool only if an operator was idle. Examining Table 5-26, it is clear that treatments using the CMS and MLC ORR rules performed the same on seven performance measures:

- the number of labor transfers;
- the flow time;
- the time in the order pool;
- the total time in the system;
- the work in process inventory; and,
- the waiting time in the shop.
- They also had the same mean hours tardy performance when the planning system was UNCONTROLLED.

The time based performance measures are discussed first.

The trigger mechanism had a major impact on the time based performance measures. First, the trigger kept the jobs in the order pool until their gateway work center had an empty queue. This created the large pool times for CMS and MLC (Figure 5-41). It also reduced their wait time (Figure 5-44) and flow time (Figure 5-46). However, the total time in the system (order pool time plus flow time) was reduced however. Multiple comparisons of the total time in the

system are shown in Figures 5-42 and 5-48. These comparisons demonstrate that the MLC and CMS rules produce a shorter time in the system than the IMM and MIL rules. Only four treatments (1131, 1231, 1331 and 2231) tie with either an MLC or CMS rule. With an $\alpha = 0.05$, 3.5 errors are expected when making this many comparisons, so having four treatments tied with MLC and CMS is not unreasonable. This result is significant because prior research has not found that ORR rules reduce the total time in the system (Melnyk and Ragatz, 1989; Kanet, 1988).

The total time in system for CMS and MLC is decreased by reducing the number of jobs on the floor. This does increase the number of labor transfers. The operators must move from machine to machine to obtain work.

CMS and MLC ORR rules do influence the mean hours tardy (Figure 5-43) when using the UNCONTROLLED planning system. This suggests that the planning system may be the major influence on the mean hours tardy.

Both the MLC and CMS ORR rules have equal levels of work in process (WIP) inventory. Their WIP inventory is significantly less than the WIP inventory for treatments using other ORR rules. This trigger mechanism gives the lowest level of WIP inventory for both planning systems.

Summary: The trigger mechanism can result in significant reductions in the total time in the system, but it will increase the number of labor transfers. The trigger mechanism can decrease the mean hours tardy, but the planning

system also influences the mean hours tardy. The trigger mechanism significantly reduces the WIP level in the shop.

5.5.2 JOB SELECTION MECHANISM

Because the only difference between the CMS and MLC rules is the job selection mechanism, it is assumed that the job selection mechanism is responsible for any performance differences. The job selection mechanism for MLC is a heuristic to obtain a finite load in the shop. The CMS job selection mechanism considers only congestion and job processing times while loading the shop. The MLC rule performs better than the CMS rule and the other ORR rules on five of the performance measures shown in Table 5-26: They were:

- mean total cost per week;
- total inventory per week;
- mean hours tardy per week with SMOOTH planning;
- finished goods inventory; and,
- percent of tardy jobs.

Figures 5-16 and 5-19, show the multiple comparisons of the total inventory performance measure for the SMOOTH and UNCONTROLLED planning systems respectively. With both planning systems, treatments using MLC have significantly less total inventory than the corresponding treatments using other ORR rules.

Referring to Figure 5-12, treatments using the MLC rule had significantly fewer hours tardy work per week with the SMOOTH planning system. If the planning system was UNCONTROLLED, four treatments using MLC tied with their corresponding treatments using CMS. However, treatments using MLC performed better than their corresponding treatments using other ORR rules when the performance measure was the percent of tardy jobs (Figure 5-45).

These results suggest that the job selection mechanism determines the finished goods inventory and tardiness performance of the shop for a given planning system. The total inventory level seems to be the combined results of finished goods inventory and work in process inventories.

5.5.3 PLANNING SYSTEM

The impact of the planning system on each performance measure is summarized in Table 5-26. For all the performance measures where there is a significant difference between the planning systems, the SMOOTH planning system performs better than the UNCONTROLLED planning system except for the total inventory.

The treatments using the MLC ORR rule and the SMOOTH planning system have less total inventory than the treatments using the other ORR rules and the UNCONTROLLED planning system. However, all the other treatments using the UNCONTROLLED planning system have less total inventory than treatments using the SMOOTH planning system.

The SMOOTH planning system performed better than the UN-CONTROLLED planning system on six performance measures:

- mean total cost;
- mean flow time;
- mean total time in the system;
- mean hours tardy per week;
- mean hours tardy per tardy job;
- mean percent tardy jobs.

These results suggest that the SMOOTH planning system improved all time related performance measures. It especially improved all the tardiness measures. Also, it suggests that the UNCONTROLLED planning system reduces total inventory if combined with the MLC ORR rule.

5.5.4 ORR RULES AND PERFORMANCE

As shown in Table 5-26, treatments using the CMS ORR rule and SMOOTH planning system increased total cost as compared to treatments releasing jobs immediately (IMM). IMM also reduced total labor transfers and time in the order pool compared to treatments using the ORR rules. However, on all other performance measures shown in Table 5-26, the ORR rules improved performance compared to IMM. This suggests that these ORR rules will perform better than IMM on time based, tardiness based or inventory based performance measures.

5.5.5 JOB DISPATCHING AND ORR RULES

Table 5-24 shows that all interactions of the ORR rules and job dispatching rules are significant. Figures 5-10 and 5-11 graphically show that the impact of the job dispatching rule on total cost performance is reduced by the trigger mechanism. The exception to this are those treatments using MLC and the SMOOTH planning system. These treatments had significant differences between the job dispatching rules. However, there is no apparent pattern about which job dispatching was the best rule.

Figures 5-25 and 5-29 show that the job dispatching rules used did not influence the mean tardiness.

When the performance measure was total inventory (Figures 5-26 and 5-30), there was no difference between the job dispatching rules unless the ORR rule was MLC. The best performing dispatch rules were MODD and CMDD.

Figures 5-27 and 5-31 show that as the trigger mechanism became more restrictive, job dispatching did not influence the number of labor transfers. For IMM and MIL either MODD or CMDD reduced the number of labor transfers.

The multiple comparisons of the flow time (Figure 5-40) were arranged by the job dispatching rule in Figure 5-49. Examining these comparisons in Figure 5-49, shows that as the trigger mechanism restricts the number of jobs in the shop, the ability of the job dispatching rule to affect performance diminishes.

The multiple comparisons of the total time in the system were organized by their job dispatching rule in Figure 5-50. These comparisons show that job

dispatching has no influence when it is combined with the CMS rule. It can impact performance when the MLC and the IMM ORR rules are used. This suggests that the trigger mechanism of the ORR rule limits the impact of the job dispatching rule, but that the job selection mechanism can create a role for job dispatching. However, there is no clear indication about which job dispatching rule is the best.

The multiple comparisons of the percent of tardy jobs were organized by their job dispatch rules and graphed in Figure 5-51. These comparisons show that job dispatching had little impact on the percent of tardy jobs for the IMM and MIL ORR rules and no influence on the CMS and MLC ORR rules.

5.5.6 "WHEN" ASSIGNMENT AND ORR RULES

All the interactions of the ORR and "when" rules are significant as shown in Table 5-24. However, there is no clear pattern about which "when" rule performs the best.

Figures 5-8 and 5-9 show that if the performance measure is total cost there are no significant differences between "when" rules with the UNCONTROLLED planning system. However there are significant differences between "when" rules if the IMM and MIL ORR rules are in use with the SMOOTH planning system.

Examination of the individual cost components does not reveal a clear pattern about the "when" rules. Figure 5-33 shows that with the SMOOTH

planning system, the "when" rule has no influence on the number of tardy hours per week. In Figure 5-34, the "when" rule makes a significant difference in the total inventory level if the MLC ORR rule is in use. In Figure 5-35 the "when" rule has no impact on the labor transfers when MLC and CMS are used, but with IMM and MIL, QUE is always the best labor assignment rule.

As illustrated in Table 5-26, examination of the secondary performance measures does not show a clear pattern of differences between the "when" rules. The "when" rules seem to have no significant impact on performance. An illustration of this is the one incident in Figure 5-52, where the QUE "when" rule results in the least time in the system with the IMM ORR rule and the SMOOTH planning system. But it increased the time in system twice when used with the MLC ORR and UNCONTROLLED planning system. For most treatments, there was no significant difference in the total time in the system due to the "when" rule.

5.5.7 SIGNIFICANCE OF JOB DISPATCHING

As summarized in Table 5-26, multiple comparisons of the total cost do not reveal any significant differences between job dispatching rules. The component costs were examined using multiple comparisons. These show that the job dispatching rule is significant only with certain ORR rules. If the ORR rule is IMM or MIL, the MODD and CMDD rules reduce the number of labor transfers

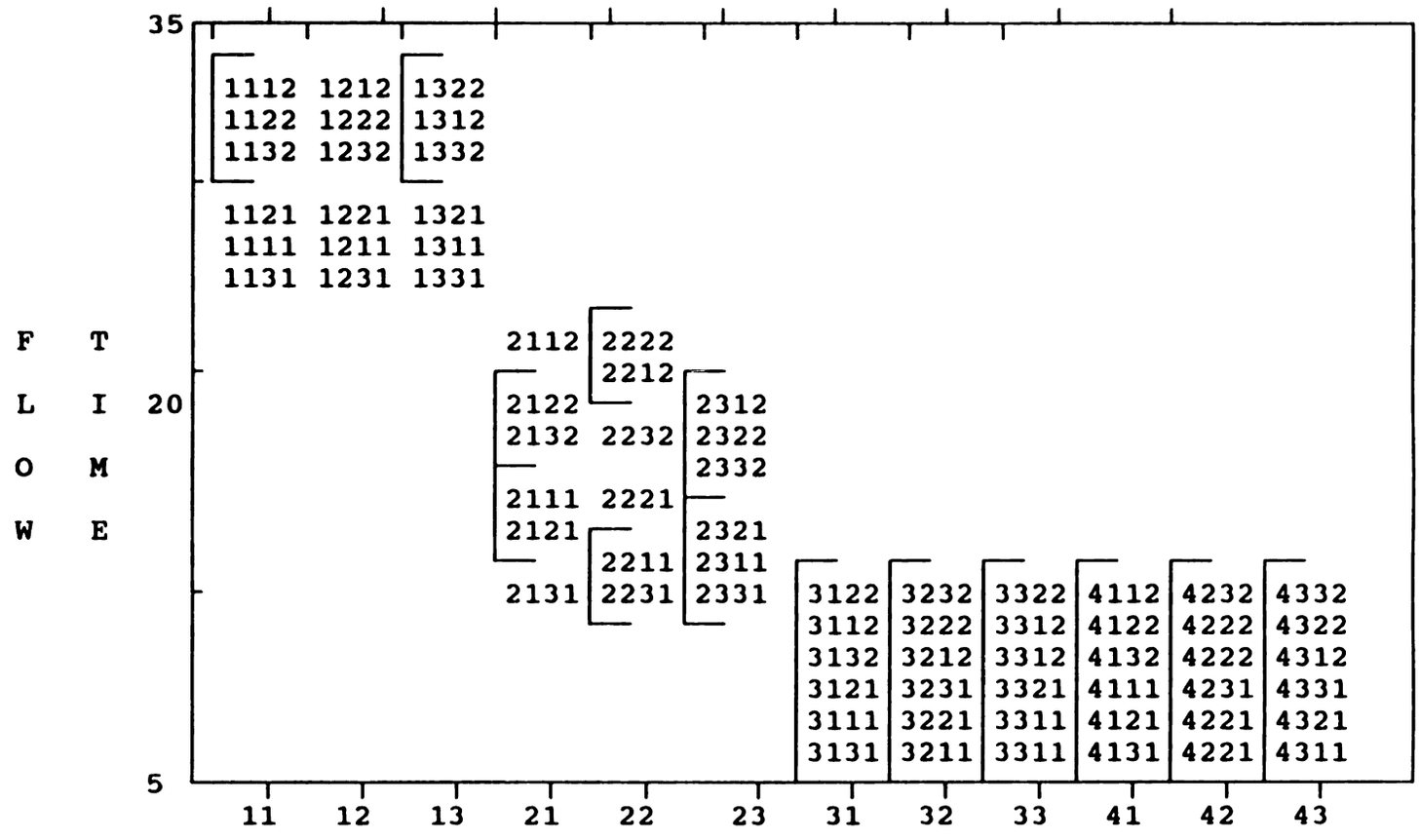
TABLE 5-26
PERFORMANCE MEASURE SUMMARY

PERFORMANCE MEASURE	ORR RULES RANKING	PLANNING SYSTEM RANKING	"WHEN" LABOR ASSIGNMENT	JOB DISPATCHING
MEAN TOTAL COST FIGURE 5-3,15,18	SMOOTH: MLC>MIL>IMM>CMS UNCONT: MLC>(CMS,MIL)>IMM	SMOOTH> UNCONTROLLED	NO DIFFERENCE	NO PATTERN
LABOR TRANSFERS FIGURES 5-17,20	IMM>MIL>(MLC,CMS)	NO DIFFERENCE	QUE>CLR>JOB FOR MIL,IMM	IMM,MIL: (MODD,CMDD)
FLOW TIME FIGURE 5-40, 49	(MLC,CMS)>MIL>IMM	SMOOTH> UNCONTROLLED	NO DIFFERENCE	IMM: CMDD
ORDER POOL TIME FIGURE 5-41	IMM>MIL>(CMS,MLC)	NO DIFFERENCE	NOT APPLICABLE	NOT APPLICABLE
TIME IN SYSTEM FIGURE5-42,48,50	(MLC,CMS)>MIL>IMM	SMOOTH > UNCONTROLLED	NO DIFFERENCE	IMM: CMDD
WAIT TIME: SHOP FIGURE 5-44	(MLC,CMS)>MIL>IMM	(MLC,CMS)>SMTH MIL> UNC MIL>SMTH IMM>UNC IMM	NO DIFFERENCE	NO DIFFERENCE
TOTAL INVENTORY FIGURE5-13,16,19	SMOOTH: MLC>(CMS,MIL,IMM) UNCONT: MLC>(CMS,MIL)>IMM	MLC UNC>MLC SMTH>UNC>SMTH	MLC ONLY DIFFERENCE	MLC:(MODD, CMDD) ONLY
WORK-IN-PROCESS FIGURE 5-47	(MLC,CMS)>MIL>IMM	FOR MIL & IMM, SMOOTH> UNCONTROLLED	NO DIFFERENCE	FOR MIL&IMM CMDD BEST
FINISHED GOODS FIGURE 5-46	IMM>MIL>MLC>CMS	UNCONTROLLED> SMOOTH	NO DIFFERENCE	NO DIFFERENCE
MEAN HOURS TARDY FIGURE 5-12	SMOOTH: MLC>(CMS,MIL)>IMM UNCONT: (MLC,CMS)>MIL>IMM	SMOOTH> UNCONTROLLED	NO DIFFERENCE	NO DIFFERENCE
HOURS TARDY/JOB FIGURE 5-43	NO DIFFERENCES	SMOOTH > UNCONTROLLED	NO DIFFERENCE	NO DIFFERENCE
PERCENT TARDY FIGURE 5-45, 51	SMOOTH: MLC>(CMS,MIL)>IMM UNCONT: MLC>CMS>MIL>IMM	SMOOTH > UNCONTROLLED	NO DIFFERENCE	NO DIFFERENCE

LEGEND: > INDICATES BETTER THAN
( ) INDICATES EQUAL PERFORMANCE
SMTH INDICATES SMOOTH PLANNING SYSTEM
: FOR THIS LEVEL OF FACTOR ONLY (e.g. UNCONT:)
UNC UNCONTROLLED PLANNING SYSTEM
UNCONT UNCONTROLLED PLANNING SYSTEM

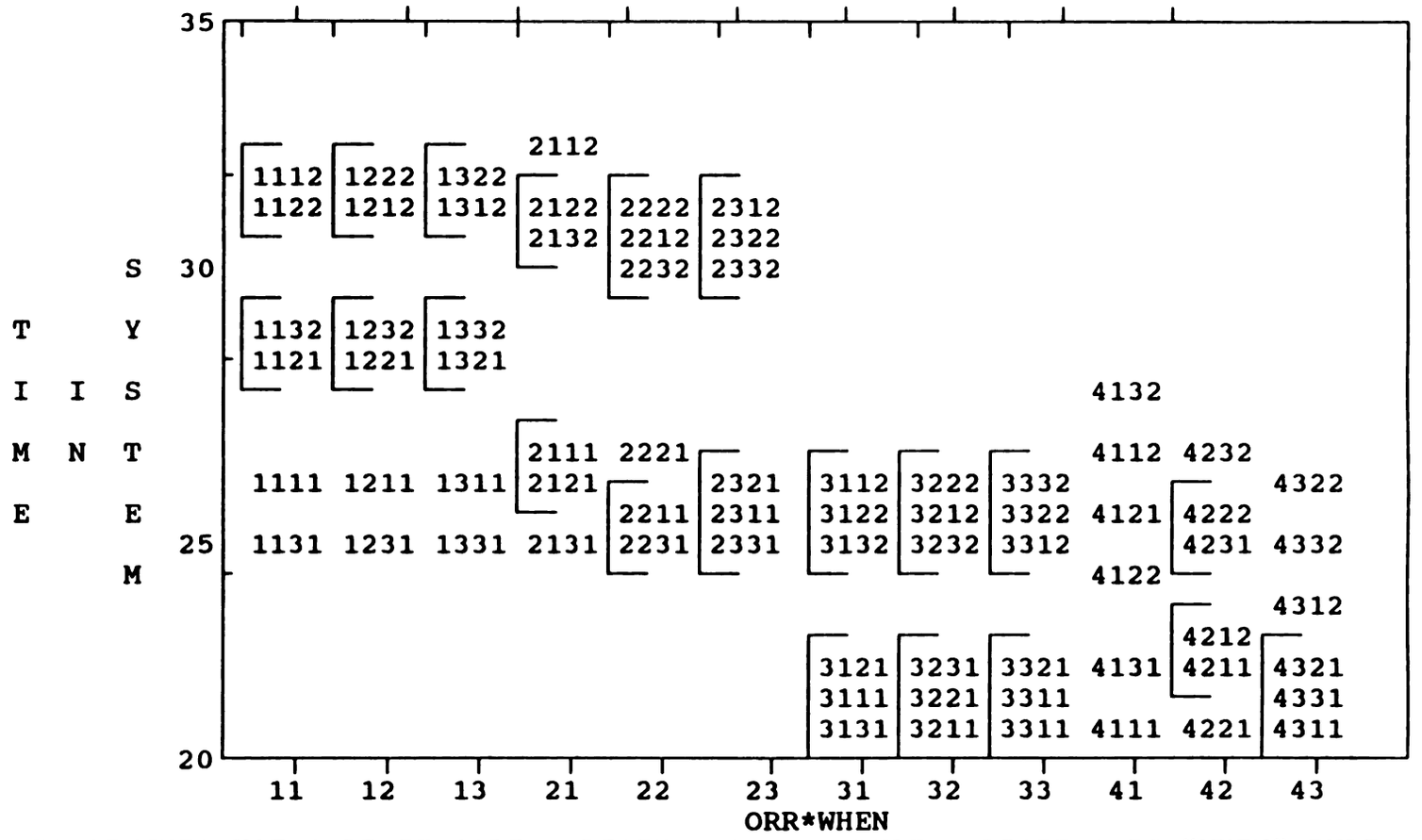
FIGURE 5-49
COMPARISONS OF FLOW TIME BY DISPATCHING

FIGURE 5-49
COMPARISONS OF FLOW TIME BY DISPATCHING



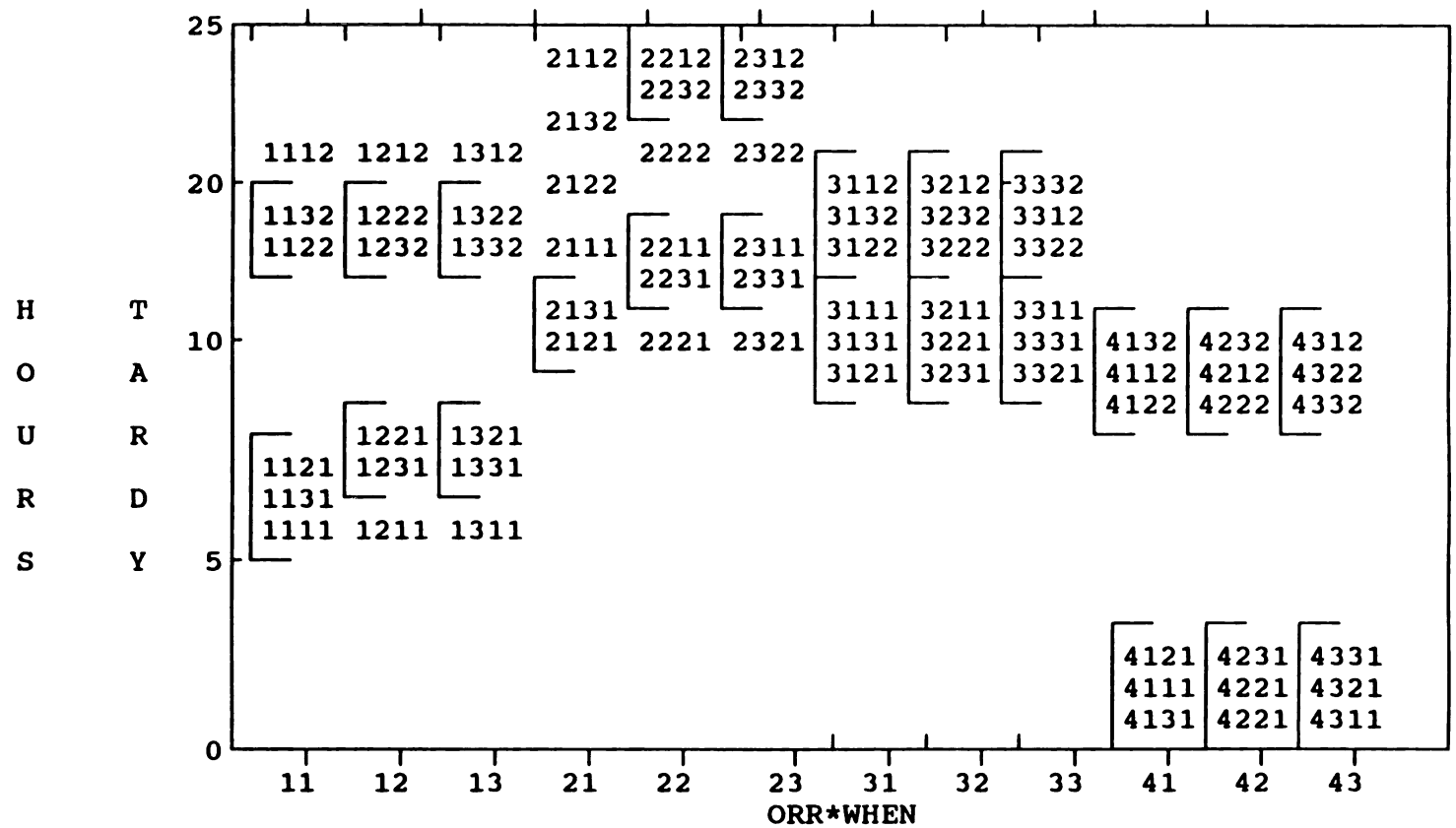
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-50
COMPARISONS OF TIME IN SYSTEM BY DISPATCHING



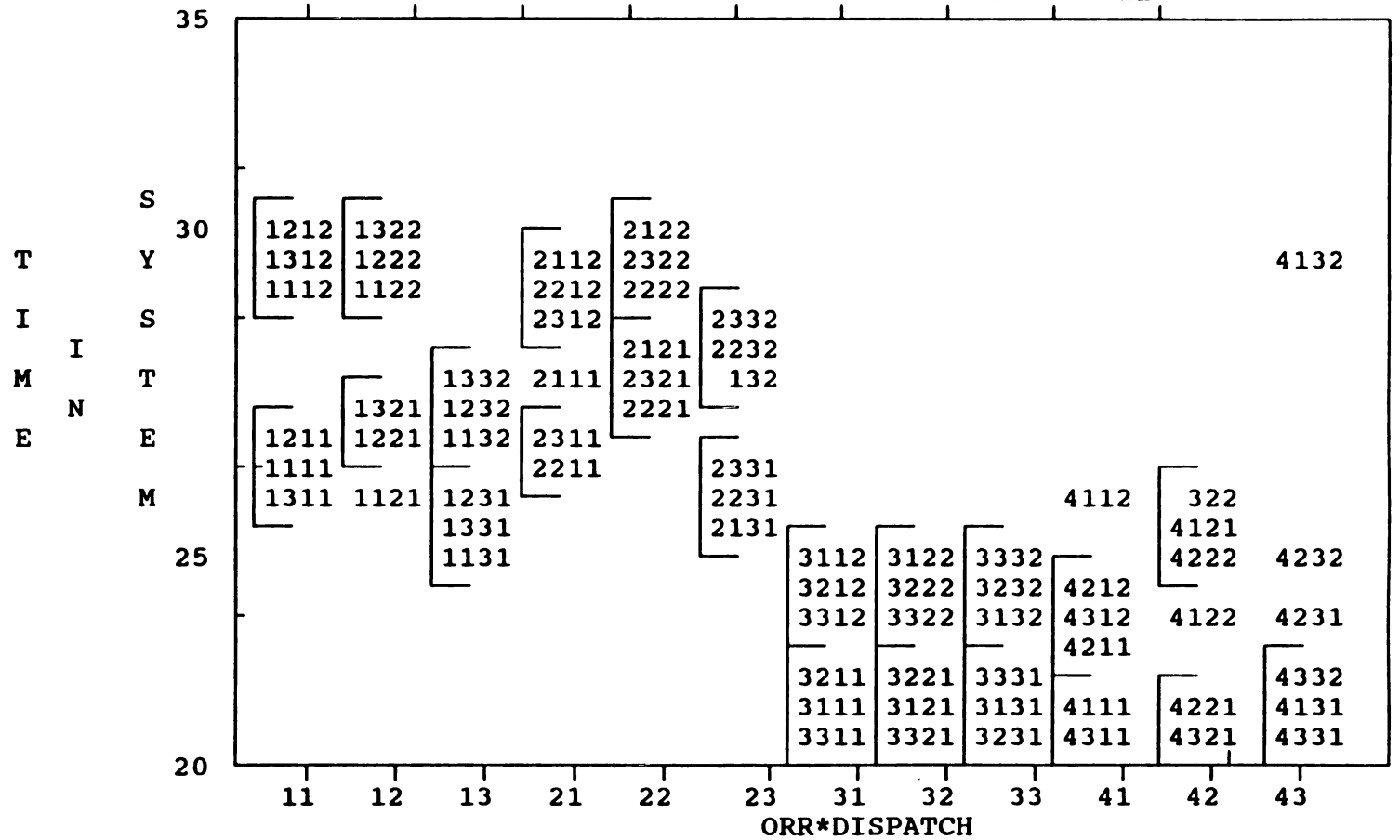
TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-51
COMPARISONS OF PERCENT JOBS TARDY BY DISPATCHING



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

FIGURE 5-52
COMPARISONS OF TIME IN SYSTEM: "WHEN" RULE



TREATMENTS USE 4 DIGIT CODE. DIGIT 1 IS ORR RULE. DIGIT 2 IS "WHEN" RULE.
DIGIT 3 IS DISPATCH RULE. DIGIT 4 IS PLANNING: ORR:1=IMM,2=MIL,3=CMS,4=MLC.
WHEN:1=QUE,2=JOB,3=CLR. DISPATCH:1=FCFS,2=MODD,3=CMDD. PLAN:1=SMOOTH,2=UNCONTROLLED

compared to FCFS. And, MODD and CMDD reduce the total inventory if MLC is the ORR rule.

The secondary performance measures do not exhibit a clear idea of when the job dispatching rule is significant either. In Figure 5-49, with the SMOOTH planning system and the IMM ORR rule, CMDD gives significantly less flow time than the other dispatching rules. However, with the other ORR rules, there is no significant difference between the job dispatching rules. In Figure 5-50, the CMDD rule gives significantly less total time in the system with the SMOOTH planning system and the IMM ORR rule. There is no significant differences with the other ORR rules.

In Figure 5-51 FCFS gives the lowest percent tardy jobs with the IMM and SMOOTH rules. But MODD gives the lowest percent of tardy jobs with the MIL and SMOOTH rules. And, job dispatching is not significant with the MLC and CMS ORR rules.

In Figure 5-47 CMDD gives the lowest WIP inventory if ORR rule is MIL or IMM. There is no significant differences between job dispatching rules with the other ORR rules.

As shown in Table 5-26, there are no significant differences between the job dispatching rules with the other performance measures. The only clear pattern from this examination of job dispatching is that as the ORR rules increasingly restrict the number of jobs on the floor, the job dispatching rule has reduced influence.

5.5.8 SUMMARY OF POST HOC ANALYSIS

The post hoc analysis concentrated on determining the reasons for the differences between the treatments.

The findings were:

- The SMOOTH planning system decreased tardiness compared to the UNCONTROLLED planning system.
- The SMOOTH planning system decreased the mean time in the system.
- The trigger mechanism of the CMS and MLC ORR rules decreased the total time in the system.
- The trigger mechanism of the CMS and MLC ORR rules decreased the WIP inventory.
- The job selection mechanism used in the MLC rule decreased the tardiness when used with either planning system.
- The job selection mechanism decreased the finished goods inventory level.
- The choice of the "when" rules had little impact on any performance measure except for the number of labor transfers.
- The choice of the job dispatching rule became insignificant as ORR rule restricted the number of jobs on the floor.

There is some overlapping of the contributions to improved performance made by the planning system, the trigger mechanism and the job selection

mechanism. But each clearly impacted the performance measurements discussed above.

The next section examines the sensitivity of the mean total cost performance measure to the cost structure used to calculate it.

5.6 SENSITIVITY ANALYSIS

The purpose of the sensitivity analysis was to determine if the results of the experiment were unique to the choice of the cost structure used as the performance measure. As explained in chapter 3, the cost structure was the ratio of tardy costs to inventory costs and the ratio of labor transfer costs to inventory costs. Thirty-six (36) different cost structures were examined during the sensitivity analysis. All the cost structures that were examined were shown in Table 5-20

Multiple pair-wise comparisons of the mean total cost for each treatment were done for each cost structure using the Bonferroni procedure described above. The best treatments for each cost structure were selected based on the results of the pair-wise t-tests. These "best" treatments were examined to estimate the sensitivity of the results to the cost structure.

While the sensitivity analysis did not require additional runs of the simulation, some additional runs were necessary to determine if the parameters used in the ORR rules were sensitive to the cost structure (Ragatz, 1985). Chapter 4 described how the Complex algorithm was used to select the set of

parameters which would give the lowest cost for the cost structure used in the main experiment. The Complex algorithm was used here to estimate the parameters for the ORR rules for each cost structure before conducting the sensitivity analysis.

TABLE 5-27
COST RATIOS FOR SENSITIVITY ANALYSIS[1]

TARDY COST RATIOS	INVENTORY COST RATIOS	LABOR TRANSFER COST RATIOS
20	1	10
10	1	5
5	1	1
1	1	.5
.5	1	.1
.1	1	.05

5.6.1 ESTIMATION OF ORR PARAMETERS

Analysis found that the parameters used in the ORR rules were stable for all the different cost structures used in this sensitivity analysis. Consequently, additional data was not needed to conduct the sensitivity analysis.

Because computer time was limited, it was not possible to run the Complex algorithm for each cost structure examined in the sensitivity analysis. The Complex algorithm was used to check the stability of the ORR parameters in only

---

[1]. The cost structures were expressed as ratios of the inventory costs. For example if tardy work costs were 20, this meant that the charge for each hour of tardy work was 20 times the charge for each hour of work in inventory. If labor transfer costs were 10, this meant that one transfer of a laborer between work centers costs 10 times as much as having one hour of work in inventory.

three different cost structures. The cost structures selected (10/1/1, .5/1/.05, and 10/1/10) represent a wide range of cost structures.

A cost structure of 10/1/1 meant that tardy costs were 10 times higher than inventory costs and that cost of one labor transfer equaled the cost of one hour of inventory. The Complex algorithm was run for all 54 treatments that used ORR rules. The Complex algorithm and the use of the data to determine the parameters generated by the algorithm was described in chapter 4. Figure 5-53 was representative of how the bubble graphs created with the Complex algorithm data looked.[1]

The parameters used in the ORR rule were placed on the X and the Y axis. A circle or bubble was drawn around the intersection of the parameters. These bubble graphs, as explained in chapter 4, represented the relative cost for each set of parameters by the size of the circle. Large circles represent the larger total costs.

To determine if the parameters used in the main experiment were appropriate for this cost structure, the bubble graphs were inspected to see if the original parameters selected were still the low cost parameters. Figure 5-54 and all the figures drawn for this cost structure demonstrated that the parameters used in the ORR rules for the main experiment were suitable for this cost structure also. All the rules were very stable over a wide range of parameters.

---

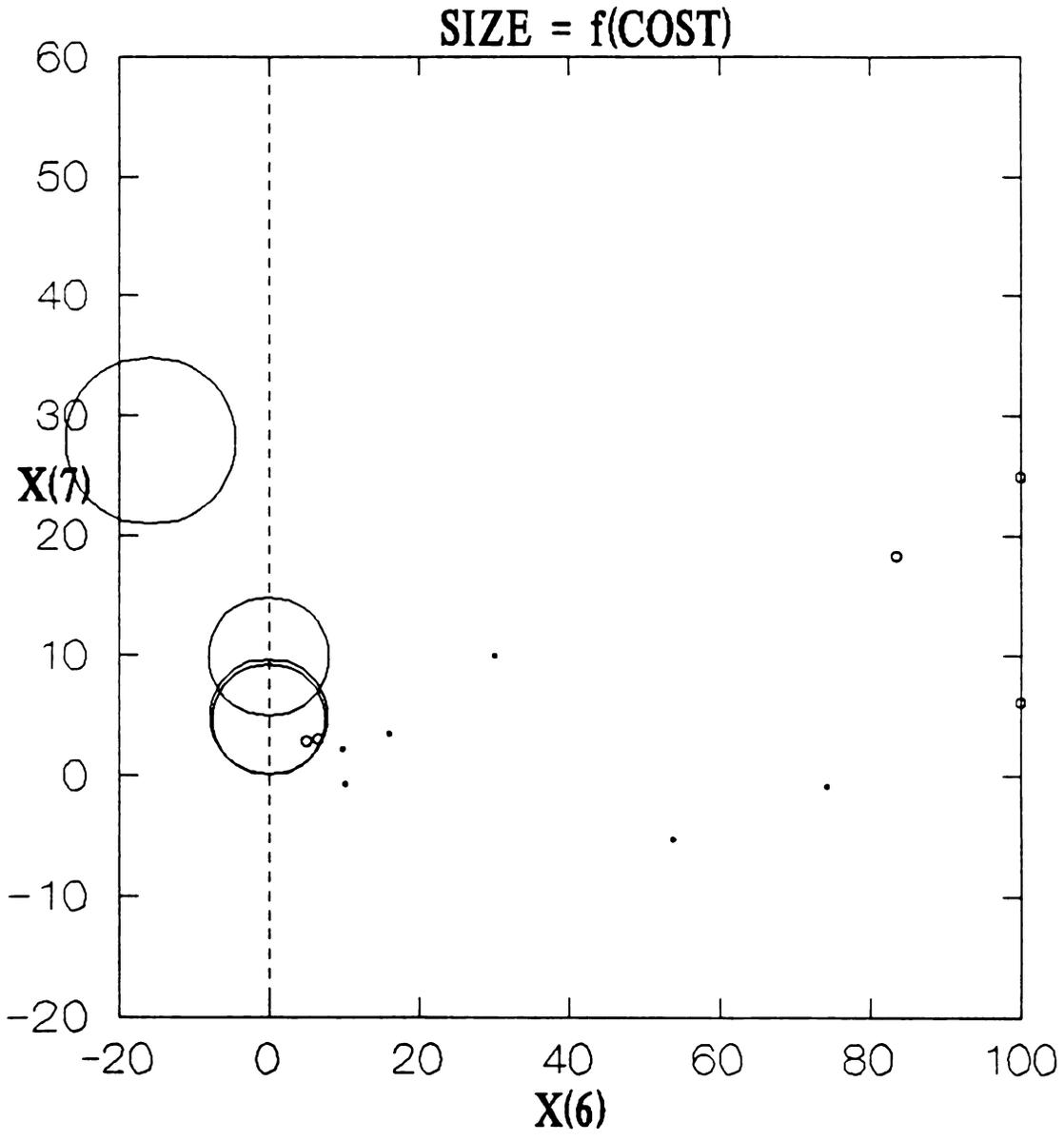
[1]. Bubble graphs were drawn with Sygraph version 1.0, (Wilkinson, 1989, p. 539).

The second cost structure examined was .5/1/.05. This structure meant that cost of one hour of tardy work was half the cost of one hour of inventory and that the cost of one labor transfer costs was 1/20th of one hour of inventory. This was selected as an extreme value as it seemed unlikely that this cost structure would be encountered in many real shops. This was an experiment to find the results if inventory was the largest cost. Figure 5-55 was representative of how all the bubble graphs of this cost structure looked. The figures showed that the parameters used in the ORR rules for the main experiment were suitable for this cost structure also.

The third cost structure examined was 10/1/10. This structure meant that the cost of one hour of tardy work was ten times as high as the cost of one hour of inventory and that the cost of one labor transfer was ten times as high as the cost of one hour of work in inventory. This was selected because it represented an extreme value for labor transfer costs. Figure 5-56 represented how all the bubble graphs of the Complex algorithm data for this cost structure looked. The figures showed that the parameters used in the ORR rules for the main experiment were suitable for this cost structure also, except for the following cells: 2221, 2231, 2311, 2321 and 2331. All these treatments used the MIL ORR rule and the SMOOTH planning system. Examining their cost at their lowest cost parameters, it was clear that they were not among the lowest cost treatments for this cost structure. Because of this, new data was not obtained for this cost structure.

FIGURE 5-53

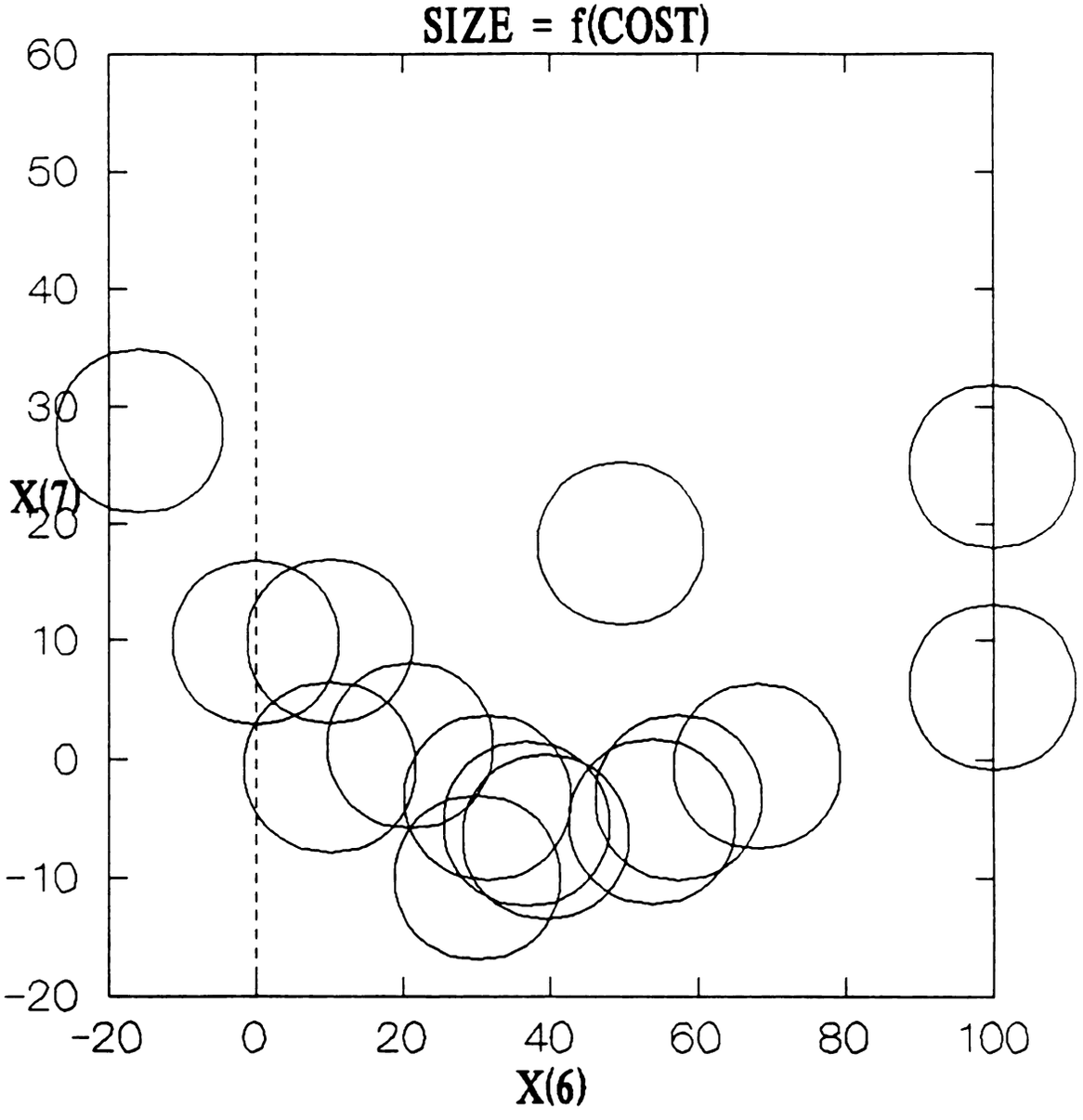
CELL 2111: COST STRUCTURE 10/1/1



MIL: RD = DD - X(6)*PROC - X(7)*CONG

FIGURE 5-54

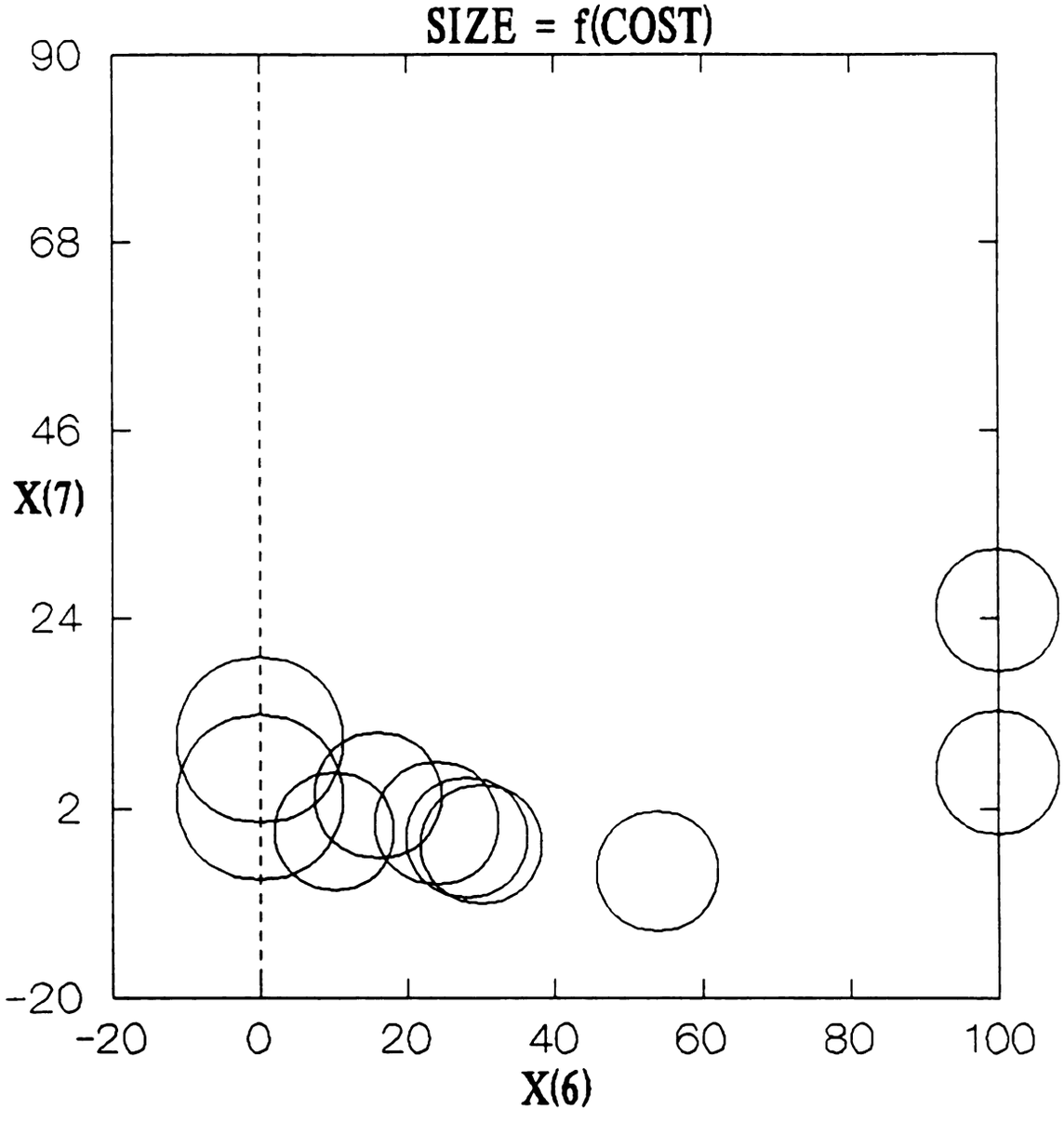
CELL 2112: COST STRUCTURE .5/1/.05



MIL: RD = DD - X(6)*PROC - X(7)*CONG

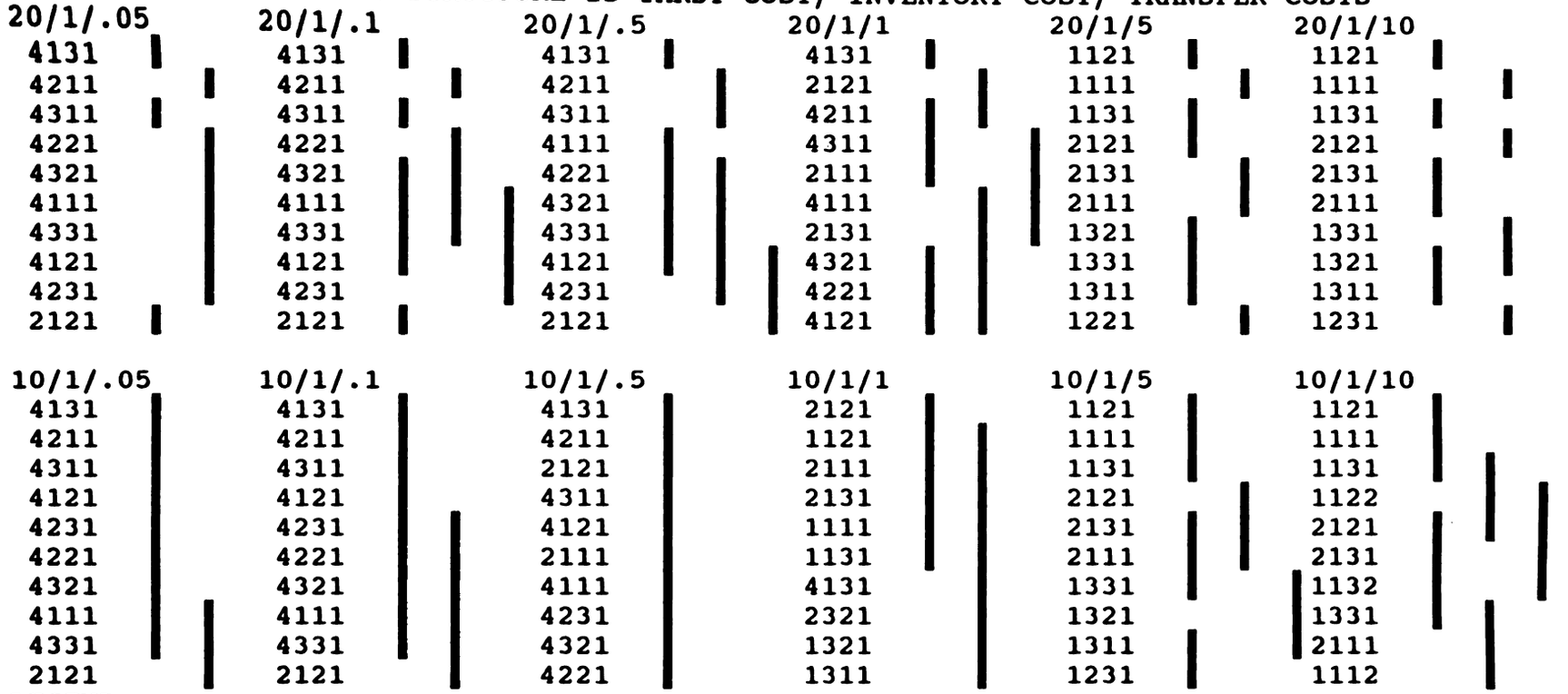
FIGURE 5-55

CELL 3121: COST STRUCTURE 10/1/10



CMS: RD = DD - X(6)*PROC - X(7)*CONG

FIGURE 5-56
BONFERRONI COMPARISONS FOR COST STRUCTURES[1]
EACH COST STRUCTURE IS TARDY COST/ INVENTORY COST/ TRANSFER COSTS

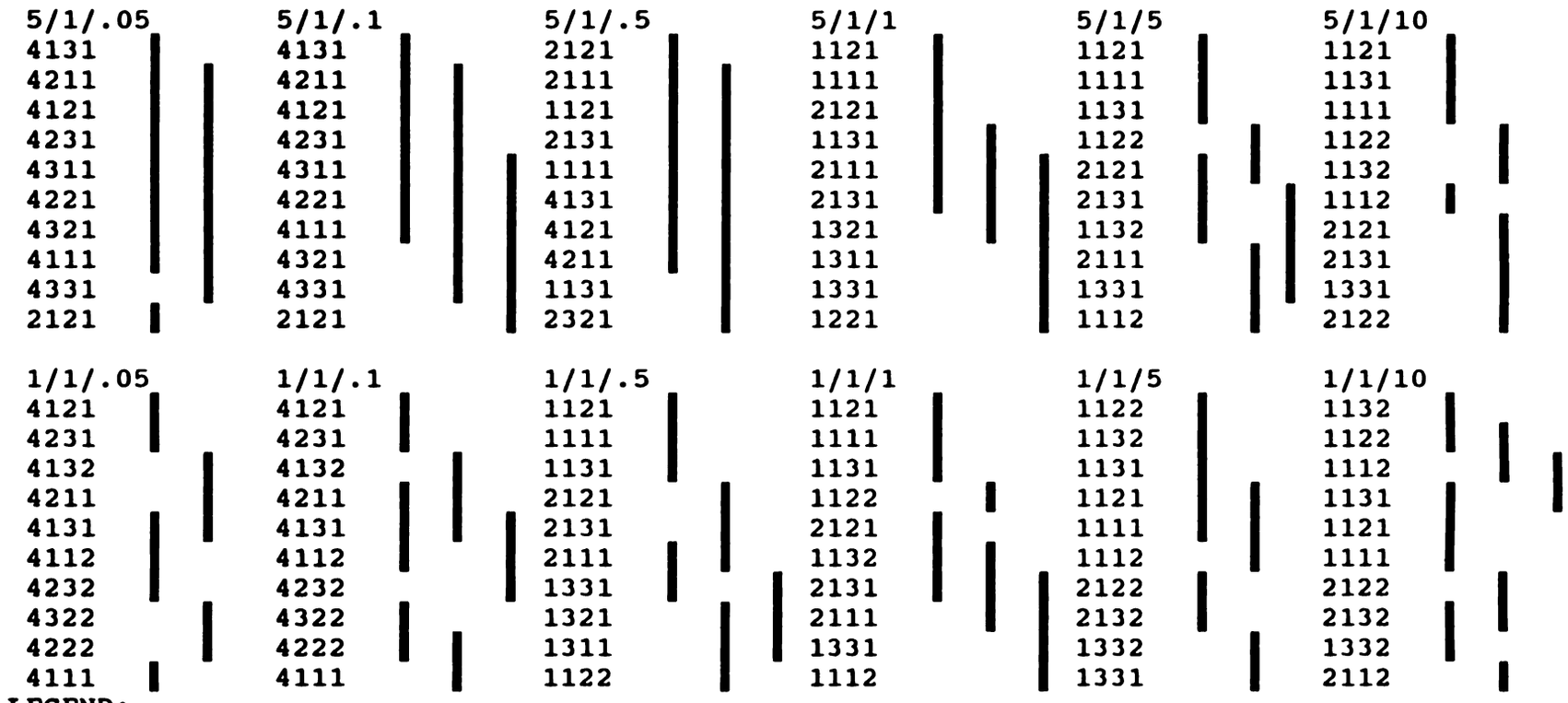


LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4:
DIGIT 1: ORR (1=IMM,2=MIL,3=CMS,4=MLC)
DIGIT 2: WHEN (1=QUE,2= JOB,3=CLR)
DIGIT 3: DISPATCH (1=FCFS,2=MODD,3=CMDD)
DIGIT 4: PLANNING (1=SMOOTH, 2=UNCONTROLLED)

---

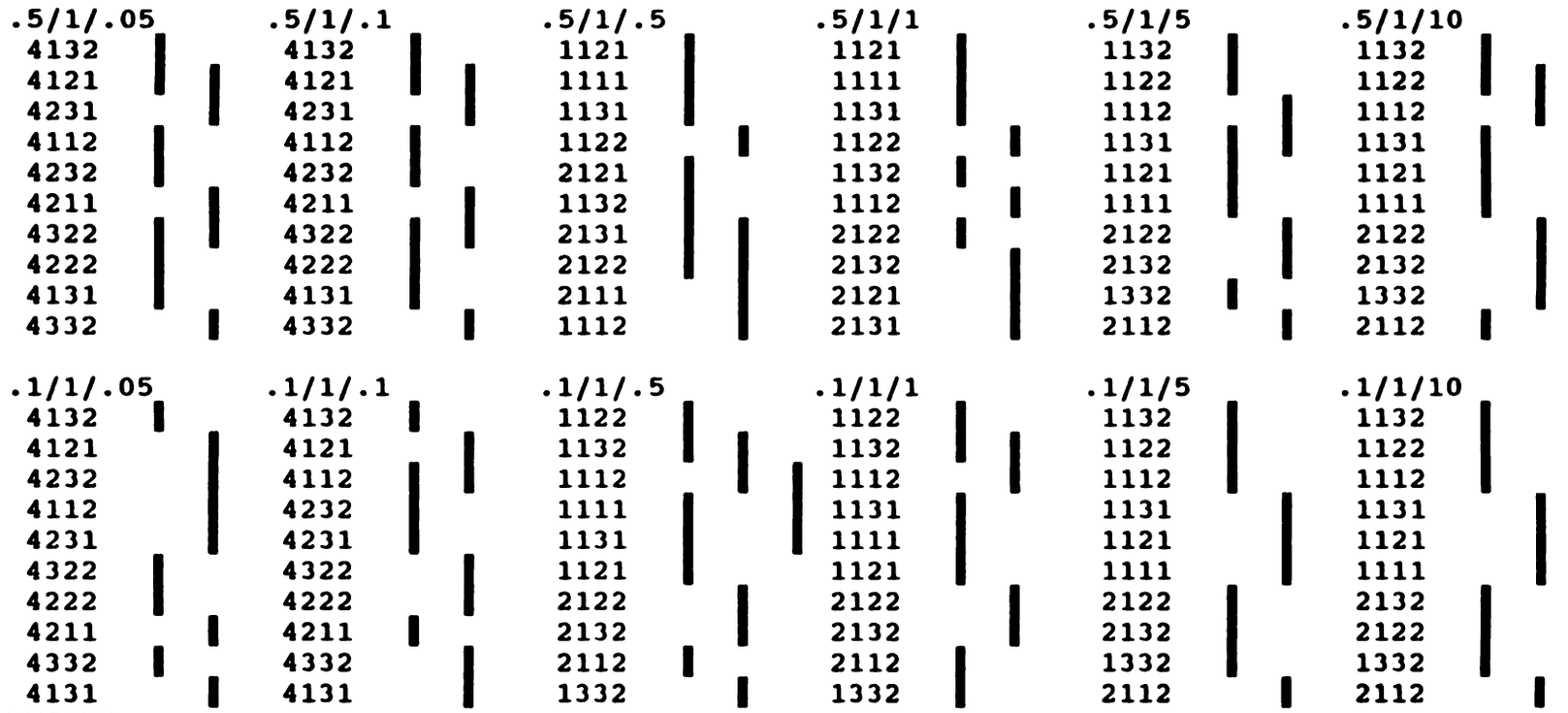
[1]. COMPARISONS WERE CONDUCTED USING SAS VERSION 6.06 USING THE GLM PROCEDURE DISCUSSED ON P.568 SAS/STAT USER'S GUIDE RELEASE 6.03. $\alpha = 0.05$

FIGURE 5-56 (cont'd)



LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1: ORR (1=IMM,2=MIL,3=CMS,4=MLC)
DIGIT 2: WHEN (1=QUE,2= JOB,3=CLR)
DIGIT 3: DISPATCH (1=FCFS,2=MODD,3=CMDD)
DIGIT 4: PLANNING (1=SMOOTH, 2=UNCONTROLLED)

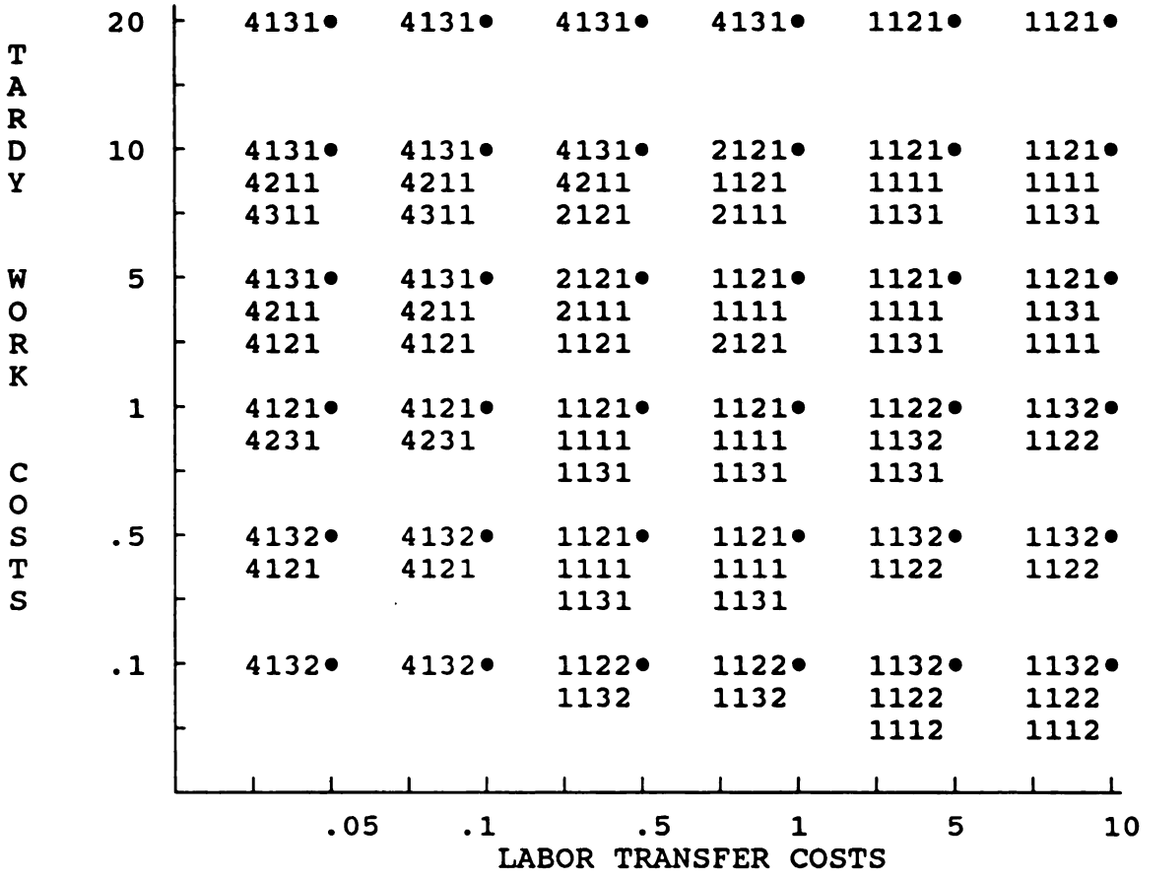
FIGURE 5-56 (cont'd)



LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1: ORR (1=IMM,2=MIL,3=CMS,4=MLC)
DIGIT 2: WHEN (1=QUE,2= JOB,3=CLR)
DIGIT 3: DISPATCH (1=FCFS,2=MODD,3=CMDD)
DIGIT 4: PLANNING (1=SMOOTH, 2=UNCONTROLLED)

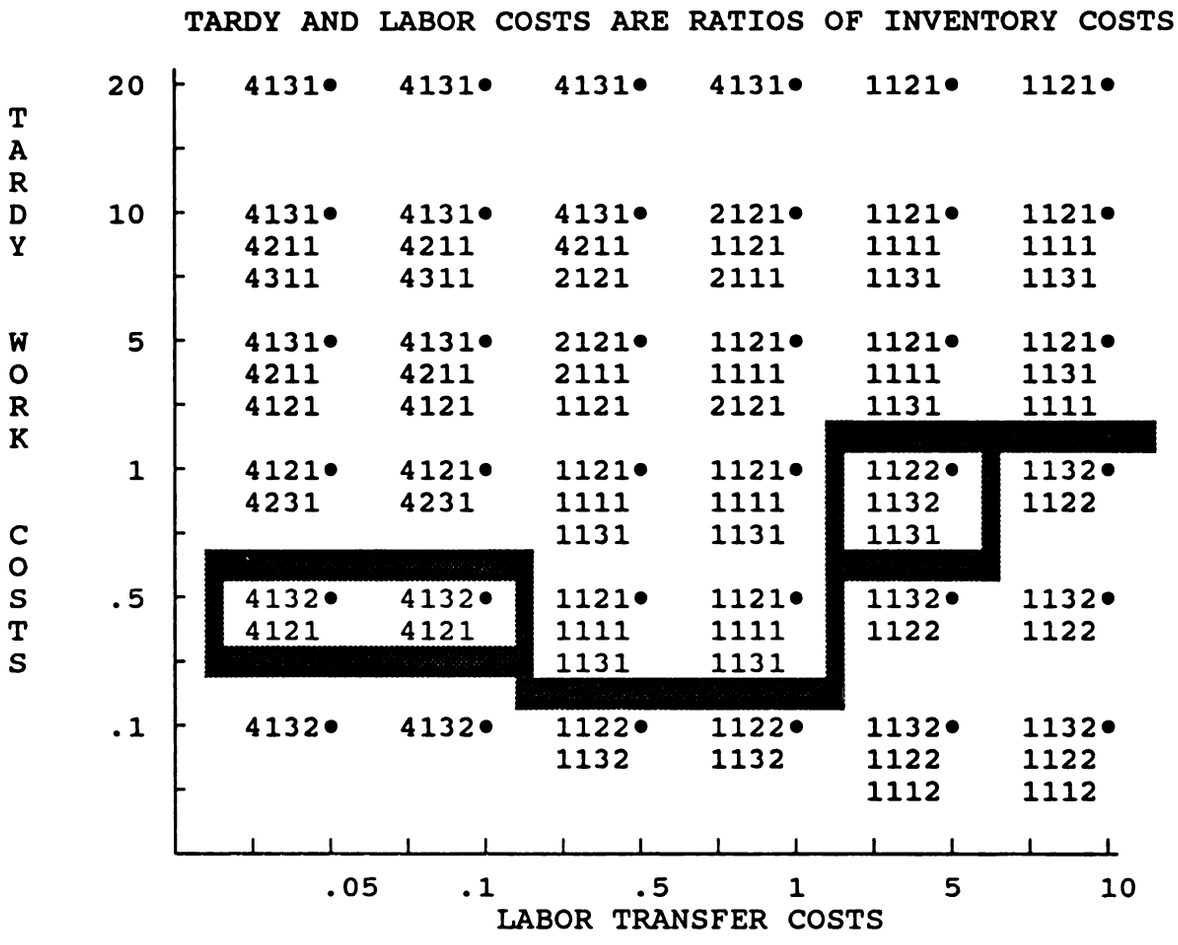
FIGURE 5-57
LOW COST TREATMENTS BY COST STRUCTURE

TARDY AND LABOR COSTS ARE RATIOS OF INVENTORY COSTS



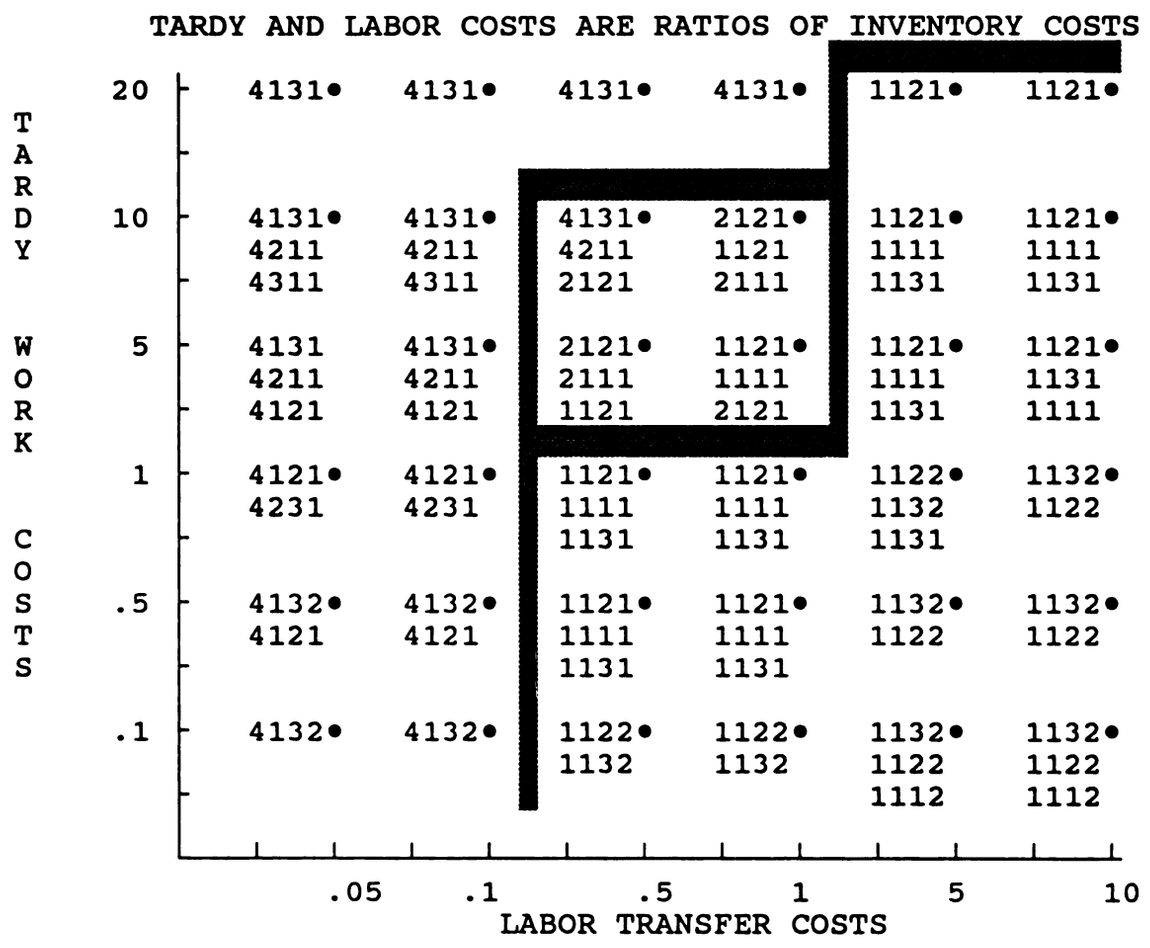
LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)
EXAMPLE (4132 INDICATES MLC,QUE,CMDD,UNCONTROLLED)

FIGURE 5-58
IMPACT OF COST STRUCTURE ON PLANNING



LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH, 2=UNCONTROLLED)
EXAMPLE (4132 INDICATES MLC,QUE,CMDD,UNCONTROLLED)

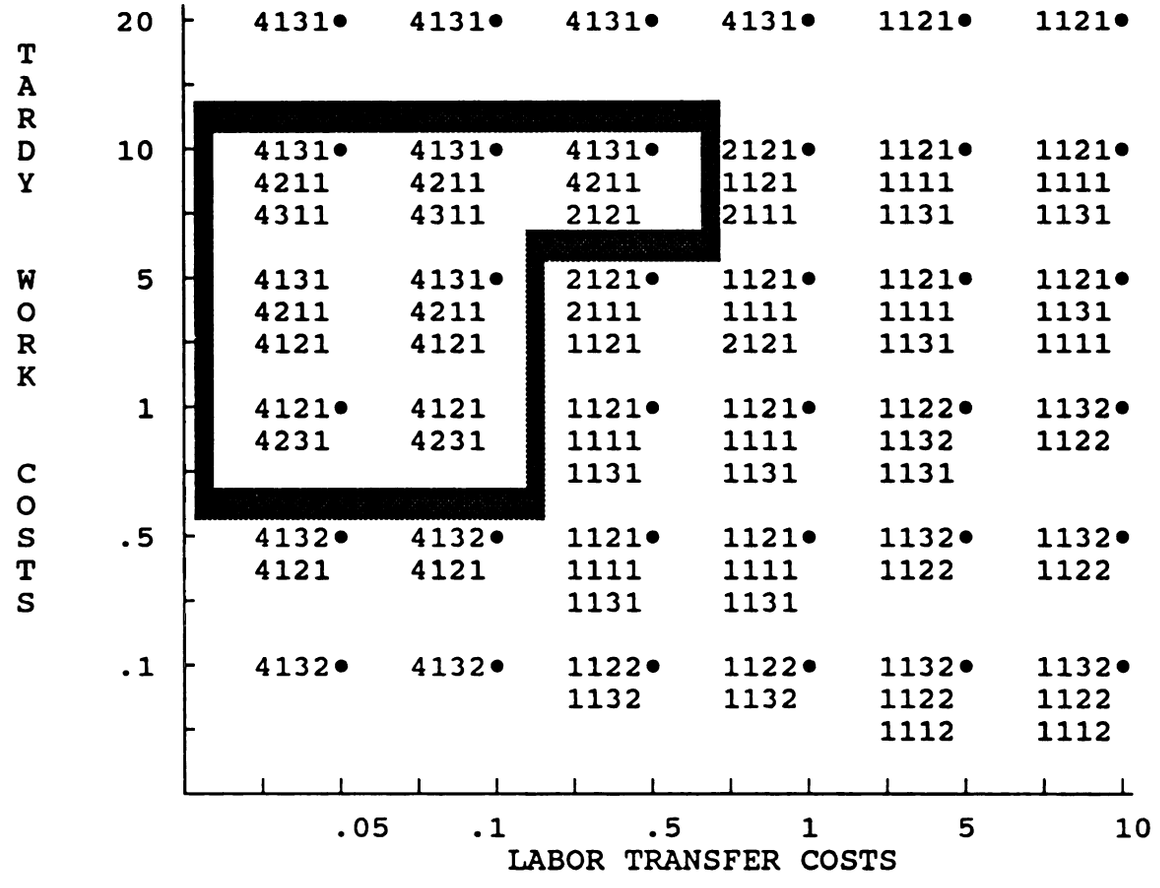
FIGURE 5-59
IMPACT OF COST STRUCTURE ON ORR RULES



LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1 = SMOOTH, 2 = EXXON)
EXAMPLE (4132 INDICATES MLC,QUE,CMDD,EXXON)

FIGURE 5-60
IMPACT OF COST STRUCTURE ON "WHEN" ASSIGNMENT

TARDY AND LABOR COSTS ARE RATIOS OF INVENTORY COSTS



LEGEND:
DIGIT 1, DIGIT 2, DIGIT 3, DIGIT 4
DIGIT 1 SPECIFIES ORR RULE (1=IMM, 2=MIL, 3=CMS, 4=MLC)
DIGIT 2 SPECIFIES WHEN RULE (1 = QUE, 2 = JOB, 3 = CLR)
DIGIT 3 SPECIFIES DISPATCH RULE (1=FCFS, 2 = MODD, 3 = CMDD)
DIGIT 4 SPECIFIES PLANNING SYSTEM (1=SMOOTH,2=UNCONTROLLED)
EXAMPLE (4132 INDICATES MLC,QUE,CMDD,UNCONTROLLED)

5.6.2 COST STRUCTURE

The sensitivity of the results to the cost structure used in the experiment are obtained by comparing the pair-wise multiple comparisons of the treatments in each cost structure to the treatments in the other cost structures. The 10 best treatments found based on the multiple comparisons for each cost structure are shown in Figures 5-56.

If a multiple comparison for a cost structure is identified as a unique "best" treatment combination for a particular cost structure, it is recorded in Figure 5-57. If more than one treatment is tied for the best treatment, the top three treatments are recorded in Figure 5-57.

Figure 5-58 contains the same data as in Figure 5-57, but a line is drawn to separate the region containing all SMOOTH planning system treatments from the UNCONTROLLED planning system treatments. This figure suggests that as the cost of tardy work decreases in proportion to the cost of inventory, the UNCONTROLLED planning system becomes the low cost planning system. The cost structure where the UNCONTROLLED planning system performs better than the SMOOTH planning system depends on the ratio of labor transfer costs to inventory costs. A box is placed around those cost structures where both the SMOOTH and the UNCONTROLLED planning systems perform equally well.

Figure 5-59 contained the same data as in Figure 5-57 but a line is drawn to separate the regions where one or the other ORR rule is preferred. As the ratio of labor transfer costs to inventory costs increases, the dominant ORR rule

changes from MLC, to a mixture of MLC, MIL and IMM and then the dominant ORR rule becomes IMM. There is a limited range in which MIL performs well. This area is represented by the box in the middle of the graph. There is no cost structure where the CMS rule is the best ORR rule. The point at which one ORR rule is preferred to a different ORR rule seems to be influenced by the ratio of tardy costs to inventory costs.

Figure 5-60 is the same data as figures 5-55 but the box represents a break where the "when" labor assignment rule becomes important. All treatments to the outside of the box use the QUE "when" labor assignment rule in the dominant treatment.

In examining figures 5-54 through 5-58 there does not seem to be a clear pattern for the choice of the job dispatching rule based on the cost structure. In many cases the different job dispatching rules performs equally well for a given cost structure.

5.7 SUMMARY

Fitting the reciprocal cost data to the ANOVA model demonstrated that the four way interaction is significant, so the analysis relied primarily on multiple pair-wise comparisons. The multiple comparisons in Figure 5-3 demonstrated that for all combinations of treatments considered, the SMOOTH planning system performs better than the UNCONTROLLED planning system. Once the planning system is selected, the second major influence on performance is the selection of

the ORR rule. The job dispatching rule and the "when" labor assignment rule had a very limited impact on performance.

In Figure 5-3, one treatment (4131) had lower cost than all the other treatments. This treatment use the MLC ORR rule, the QUE "when" labor assignment rule and the CMDD job dispatching rule. The planning system is SMOOTH. The next 11 low cost treatments included 8 treatments using the MLC ORR rule and 3 treatments using the MIL ORR rule. All 12 of these lowest cost treatments used the SMOOTH planning system.

The MLC rule was clearly the best performing ORR rule. It used LABOR information in its trigger mechanism and used a finite loading heuristic for the job selection mechanism. The CMS ORR rule used the same trigger mechanism as the MLC rule, but it used a different job selection mechanism. Given that the similarities in performance of the MLC and CMS rule was due to their shared trigger mechanism, the trigger mechanism was an important influence on seven performance measures: the number of labor transfers; the flow time; the time in the order pool; the total time in the system; the work in process inventory; the waiting time in the shop; and with the UNCONTROLLED planning system, the mean hours tardy.

The differences in performance of the MLC and CMS rules were attributed to the different job selection mechanisms used by each rule. The MLC ORR job selection mechanism performed better than all other ORR rules on five performance measures: mean total cost per week; total inventory per week; mean

hours tardy per week with the SMOOTH planning system; finished goods inventory; and, the percent of tardy jobs.

The use of POOL information in the job dispatching rule and in the "when" labor assignment rule did not improve performance regardless of the planning system. The job dispatching rule becomes less significant as the trigger mechanism decreased the number of jobs on the floor. The "when" rule is not significant except for the number of labor transfers.

The sensitivity analysis demonstrated that the results of the analysis are sensitive to the cost structure used. The dominant planning system, ORR rule and "when" labor assignment rule depended on the cost structure used. The selection of the best job dispatching rule did not seem to be influenced by the cost structure. The performance of the ORR rules did not seem to be very sensitive to the choice of the parameters used in the ORR rules.

These results will be discussed in Chapter 6.

CHAPTER 6

6.1 ANALYSIS AND CONCLUSIONS: INTRODUCTION

This chapter examines the implications of this study's findings for the research questions and scheduling theory. It also proposes future research directions based on these findings.

The two major questions that this research investigated were:

- the value of using different sources of information to make SFC decisions;
- and, the benefits of using a planning systems to reduce the variance in work loads sent from planning to the shop.

Each of these questions is addressed in turn.

6.2 INFORMATION USE IN SFC

Thee are three aspects of information usage in the SFC activities that are discussed. They are:

1. The impact of using POOL information in job dispatching rules;
2. The impact of using POOL information in the "when" labor assignment rules;
3. The impact of using LABOR information in the ORR rules;

Each aspect of these issues is addressed separately.

6.2.1 INFORMATION AND THE JOB DISPATCH RULES

A priori hypotheses 7 and 8 investigated the impact of using POOL information in the job dispatching rule in conjunction with the SMOOTH and UNCONTROLLED planning systems respectively. The alternative hypotheses that the use of POOL information in job dispatching rules decreased total cost were not supported.

The post hoc analysis also investigated the impact of using POOL information in the job dispatching rule. Multiple comparisons of the component cost measures and the secondary performance measures found that as the ORR rule restricts the number of jobs on the floor all the job dispatching rules performed the same. The post hoc analysis also found that the POOL information improved performance compared to FCFS with the IMM ORR rule for four performance measures:

- the number of labor transfers;
- the flow time;
- the time in system; and,
- work-in-process inventory.

The findings that the importance of the job dispatching rules decreased with the use of ORR rules was consistent with prior research. Prior researchers found that when ORR rules were used, the influence of the job dispatching rules on shop performance was reduced (Baker 1984; Kanet 1988; and, Melnyk and Ragatz, 1989). The probably reason for this effect is that there are very few jobs

in the queue when the ORR rules are used, consequently little information is actually needed to make the dispatching decision.

It is possible that additional information could improve performance of the job dispatching rule if it was used differently. Another method of using POOL information in the job dispatching rule would be to use the POOL information in a look ahead heuristic. There are two types of look ahead heuristics (Panwalkar and Iskander, 1977). One examines the future impact of scheduling a job now on another job that will be arriving in the queue. The second examines the future impact of scheduling a job now on that same job when it goes to the next queue on its route.

With either heuristic, the POOL information could be used to determine which work centers are critical as discussed in chapter 3. Both heuristics would give priority to the job going to the most critical work center.

This type of heuristic could improve the performance of the job dispatching rule in two ways. It would tend to create a larger workload in front of the critical work center and serve to prevent idle time at that work center. Second, the queue at the critical work center would become large enough so that there would be some jobs for the job dispatching rule to select from.

6.2.2 "WHEN" ASSIGNMENT AND INFORMATION

This research also investigated the use of POOL information in the "when" labor assignment decision. Prior research about this decision explored different methods of using SHOP and LABOR information only.

A priori hypotheses 9 and 10 investigated the impact of using POOL information with the SMOOTH and UNCONTROLLED planning systems respectively. The alternative hypotheses that POOL information decreased total cost was not supported.

The post hoc analysis also investigated the use of POOL information in the "when" labor assignment rules. Multiple comparisons of all the component cost measures, and all the secondary cost measures found that the choice of the "when" rule made a significant difference only for the number of labor transfers if IMM or MIL were the ORR rules.

Possible reasons why POOL information did not improve the "when" labor assignment rule's performance:

- The ORR rule limited the number of jobs in the queue. Because each rule determined the operator's eligibility to transfer based on the number of jobs present in the queue there was little difference in the operation of the rules.
- Because the ORR rules limited the number of jobs on the floor, there were few choices about where to transfer workers eligible for transfer.
- Assigning a cost to a labor transfer may not penalize an increased number of transfers as much as including labor transfer delays in the model. Treleven (1989) suggested that the "when" labor assignment rule was not significant when labor transfer delays were not included in the model.

Further investigation is needed about the role of information in the "when" labor assignment rule, to determine if the use of additional information will improve the performance of the rule if there are labor transfer delays.

6.2.3 INFORMATION AND THE ORR RULE

The CMS and MLC ORR rules incorporated labor information into their trigger mechanism. Neither used LABOR information in their job selection mechanism. The experiment was not designed to identify a separate effect for the trigger mechanism and for the job selection mechanism. The post hoc analysis did identify a separate effect but the a priori hypotheses treated the trigger mechanism and the job selection mechanism as one ORR rule. The a priori hypotheses are discussed first.

A priori hypotheses 2 and 3 investigated the impact of using LABOR information in these ORR rules with the SMOOTH and UNCONTROLLED planning systems respectively. The alternative hypotheses that using LABOR information in the ORR rule decreased total cost compared to the MIL rule were not supported. The MLC rule did well, but the CMS rule performed poorly compared to the MIL rule when using the SMOOTH planning system. When using the UNCONTROLLED planning system, CMS performed as well as the MIL rule.

A priori hypothesis 6 compared the treatments with ORR rules to treatments without ORR rules. The alternative hypothesis that the use of ORR rules never results in increased total costs compared to immediately releasing jobs to

the shop was not supported. The CMS ORR rule performed worse than immediately releasing jobs to the shop floor. This supports Kanet's (1988) and Melnyk and Ragatz's (1989) contention that ORR rules must be selected carefully to ensure that they actually improved performance.

The relative performances of the MLC, CMS and MIL rules were investigated further using post hoc multiple comparisons. The discussion of the significant differences that were identified is organized into two sections:

- information use in the trigger mechanism;
- and the job selection mechanism.

This discussion assumes that similarities in performance of the CMS and MLC rules is due to the trigger mechanism and that significant performance differences between them must be due to the job selection mechanisms.

6.2.3.1 INFORMATION AND THE TRIGGER MECHANISM

The trigger mechanism in the CMS and MLC ORR rules used LABOR information to limit the gateway work center inventory to zero. This inevitably reduced the job's flow time (processing time plus queue time) while increasing its time in the order pool.

This reduction of flow time has been identified in prior research using different ORR rules (e.g. Melnyk and Ragatz, 1989; Ragatz and Mabert, 1988; Kanet, 1988). But in prior research, the total time in the system (flow time plus time in the order pool) was not reduced by the ORR rules.

The post hoc analysis found that when the treatments were organized on the basis of common "when" and job dispatching rules, there were no instances

where an MLC or CMS rule had significantly longer time in the system than the corresponding MIL or IMM treatment. In most cases, the MLC or CMS rule had significantly less total time in system.

Kanet (1988) suggested a partial explanation of this finding. He found that the queue of the gateway work center could be kept at zero without introducing idle time into downstream work centers. So the wait time of the job was reduced more than the time in the order pool was increased.

The reduction in flow time achieved by this trigger mechanism was much more severe than any investigated in past research. However, this reduction in total time in the system did not automatically improve cost performance, because the CMS rule did not work better than the MIL rule. However, the MLC rule usually performed significantly better than both the CMS and MIL rules. The post hoc analysis of the inventory performance measures suggested that the trigger mechanism reduced the WIP inventory. Because the trigger mechanism used in the CMS and MLC rules severely restricted the number of jobs on the floor, jobs did very little waiting. Consequently, WIP was reduced.

This difference between the CMS and MLC rules was attributed to the job selection mechanism. The job selection mechanism is now examined.

6.2.3.2 INFORMATION USE IN JOB SELECTION MECHANISMS

The a priori hypotheses 4 and 5 compared the MLC job selection mechanism to the CMS job selection mechanism for the SMOOTH and UNCONTROLLED planning systems respectively. The MLC job selection mechanism used a heuristic to load the shop to a finite limit. The alternative hypotheses were

supported. The finite loading job selection mechanism of MLC decreased total cost compared to the CMS rule.

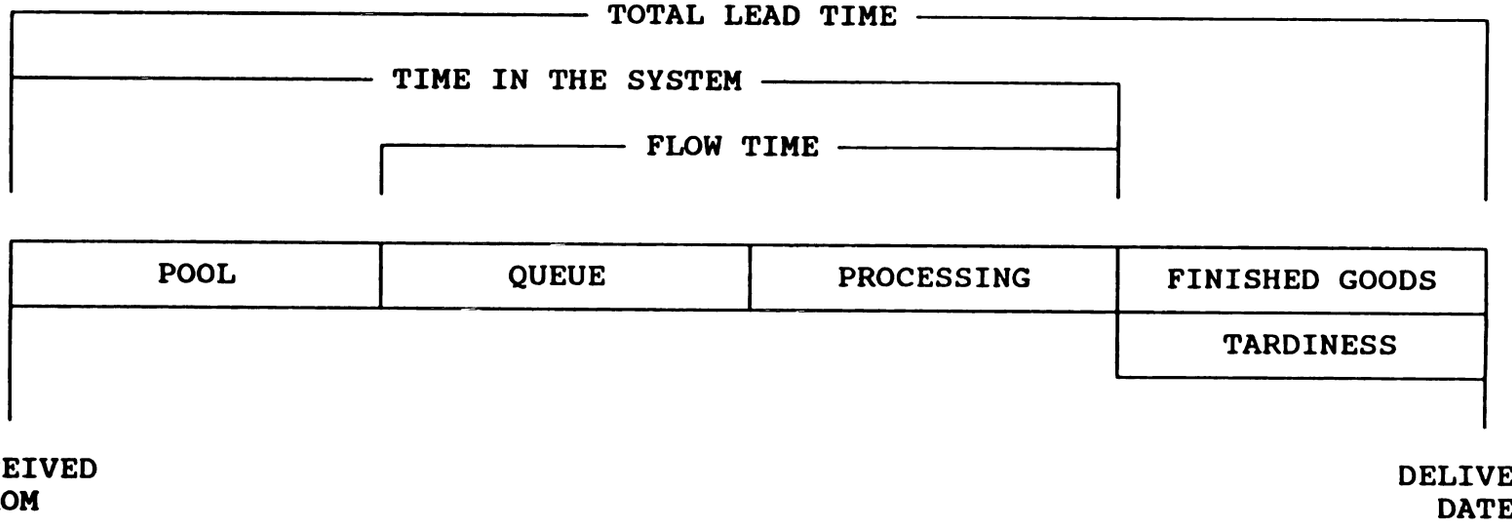
The post hoc analysis of the cost components and the secondary performance measures found that given the planning system, MLC always performed better or as well as CMS on the following performance measures:

- the total time in the system;
- the flow time;
- the total inventory;
- the finished goods inventory;
- mean hours tardy; and,
- the percent of tardy jobs.

As discussed earlier, the total time in the system for both CMS and MLC ORR rules is the same. Therefore, the differences in tardiness and total inventory between the MLC and CMS rules must be due to something else. These results are interpreted using Figure 6-1. This figure shows the elements of the total lead time of a job. A job can spend time in the order pool, on the shop floor in queues, processing on machines, or in finished goods inventory. If the job is tardy, part of the lead time would be tardiness and the time in finished goods inventory would be zero.

Because all MLC treatments have the lowest levels of total inventory, less finished goods inventory than the CMS treatments, and the least amount of tardiness, it is proposed that the job selection mechanism of MLC calculates a better release date than the CMS and MIL rules. The MLC rule release date

FIGURE 6-1
COMPONENTS OF TOTAL LEAD TIME



allows jobs to finish closer to their due date resulting in less tardiness and less finished goods inventory.

The reason the MLC release date performed well may be that it introduces less idle time at downstream work centers than the CMS release date. Kanet (1988) stated that the introduction of idle time at downstream work centers was a potential danger of ORR rules, because this idle time increases tardiness. CMS might introduce idle time because the release date is calculated as the due date minus processing time minus congestion. Since the CMS trigger mechanism prevents congestion at its work centers, the release date rule for CMS becomes:

RD = DD - 10* PROC

This formula[1] was explained fully in chapter 4 and the parameter was presented in Table 4-2. Since the due date was on average the time of arrival plus 38 and PROC was an average processing time of 4 hours, the average release date would be negative. The negative release date resulted in most jobs in the order pool being eligible for release at all times. The CMS rule selected the job with the earliest release date going to a critical work center. This job selection rule is in effect, the largest processing time (LPT) rule. Baker and Dzielinski (1960) found that the LPT job dispatch rule increased the average time in the system of the job compared to SOT.

---

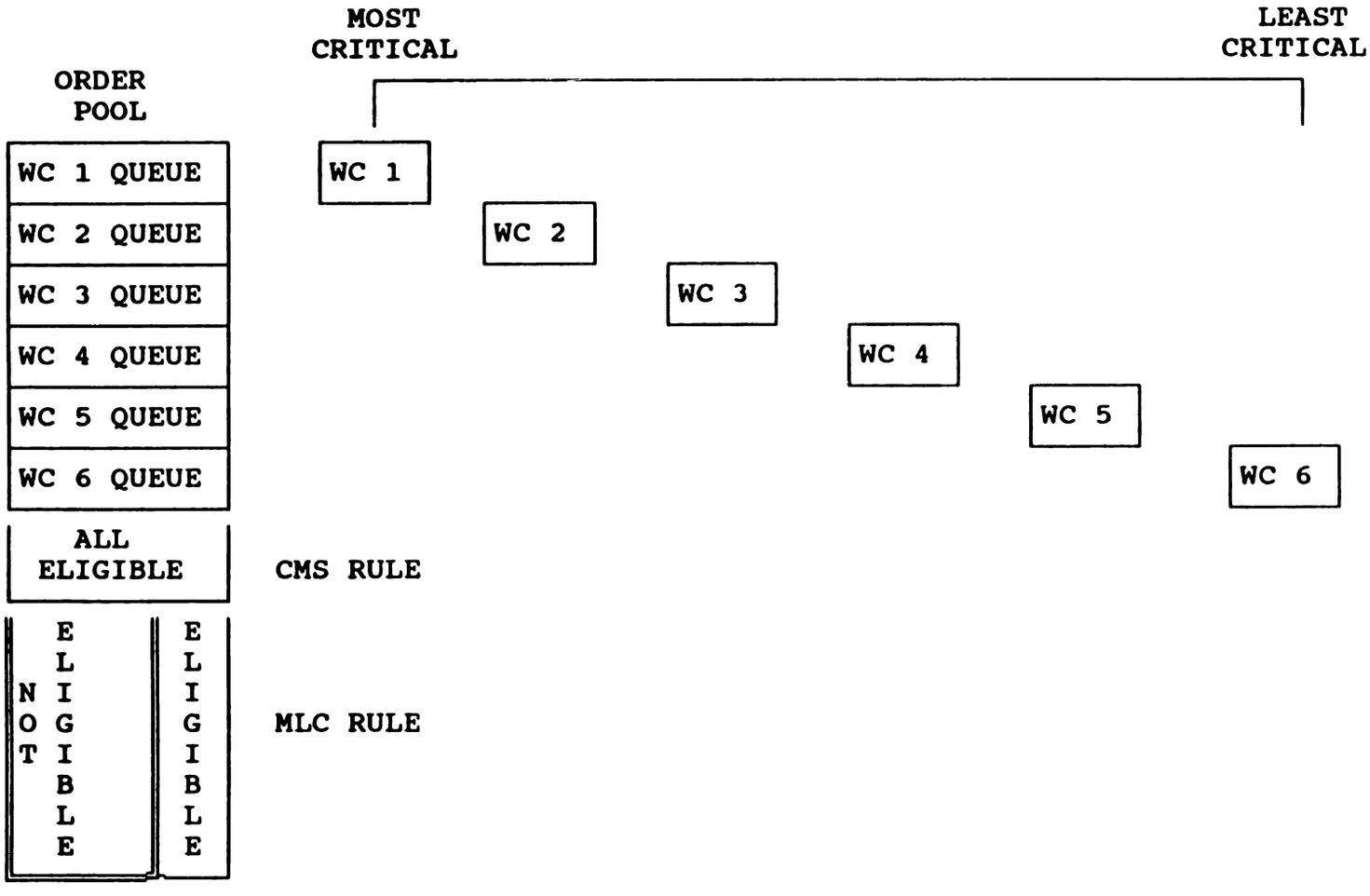
[1]. DD = Time of Arrival + 9.5*Number of operations. The average number of operations is 4. PROC = processing time of job. The processing time for each operation was drawn from a negative exponential distribution with a mean of 1.0. See chapter 3.

The job selection mechanism of the MLC rule is different than the CMS job selection mechanism. The MLC rule release date is primarily based on the job due date and the total expected load at the gateway work center. This in effect divides the order pool into separate queues of work in front of the gateway work centers. But more importantly, in each of these queues the job with the earliest release date is the job that goes to the most critical centers after leaving the gateway work center. This helps to prevent idle time at critical downstream work centers.

While MLC reduces the mean hours tardy per week compared to the other rules, it has the largest 90th percentile of tardiness for the SMOOTH planning system as shown in Table 5-6. This suggests that the MLC rule will cause some jobs to be very late.

These differences between the MLC and CMS rules are illustrated in Figure 6-2. The right hand side of Figure 6-2 arbitrarily ranks the work centers to illustrate the operation of the rules. Because of the trigger mechanism used by these rules, jobs in the pool are arranged in classes according to their gateway work center. With the CMS rule all jobs in each class are eligible for release. With the MLC rule only a few of the jobs in any class are eligible for release. Initially for both the MLC and CMS rules only the three most critical gateway work centers will be staffed. As jobs are completed and queues develop at other work centers, operators will be transferred to these less critical work centers.

FIGURE 6-2
COMPARISON OF JOB SELECTION MECHANISMS



With the CMS rule all jobs released will be to the most critical work center first. This means that one entire class of jobs will be removed from the order pool before jobs are removed from the other classes. So jobs going to critical gateway work centers finish early building FGS inventory because some of these jobs will be done well before their due date. And, jobs going to non critical gateway work centers are at risk of being tardy because they start late.

The MLC rule has only a few jobs eligible for release in each order pool class. Each class can be emptied and jobs from the less critical classes can be released to the floor. This creates on time delivery for a large number of the jobs but those jobs which have very little slack, but do not enter the shop at a gateway work center sit and wait, creating some very tardy jobs.

6.2.3.3 SUMMARY OF ORR RULES AND INFORMATION

There are two categories of findings. Those findings related to the trigger mechanism and those related to the job selection mechanism.

The trigger mechanism used LABOR information to:

- reduce the level of WIP inventory, by keeping the gateway work center inventory level at zero.
- reduce the flow time and the time in the system measures by reducing the wait time on the shop floor.

The job selection mechanism used SHOP and POOL information to:

- reduce the finished goods inventory level by setting a release date which was closer to the actual time the job needed to be released.

- reducing the tardiness of the jobs by setting a release date closer to the actual time the job needed to be released.

The MLC rule released most jobs so that they completed processing close to their due date and consequently they did not have to go into finished goods inventory, nor were they tardy.

6.2.3.4 FUTURE RESEARCH ABOUT ORR RULES

These results suggest the following hypotheses for future research:

- The trigger mechanism should keep inventory in the gateway queues at zero, but encourage inventory at critical downstream work centers.
- The job selection mechanism should consider all critical work centers and not just the critical gateway work centers when releasing work.
- The structure of the shop (e.g. the number of divisions and the type of work flow) influences the performance of both the trigger mechanism and the job selection mechanism.

The benefits of using planning systems to reduce the variance in the work loads sent to the shop will now be discussed.

6.3 THE PLANNING SYSTEM

The impact of the planning system was investigated using a priori hypothesis 1 and post hoc multiple comparisons. The alternative hypothesis 1 was

supported. There is strong support for the hypothesis that the SMOOTH planning system performed better than the UNCONTROLLED planning system. Based on the F values in the reciprocal cost ANOVA table, the choice of the planning system is a very important determinant of shop performance. All treatments using the SMOOTH planning system always performed better than treatments using the UNCONTROLLED planning system.

This result should be generalized to other settings with caution because the cost structure in this experiment did not include the possible costs of smoothing the work load. One possible cost of smoothing work would be canceled customer orders if due dates were pushed out too far. Another possible cost of smoothing work would be increased FGS inventory costs as jobs were released with an internal due date earlier than the customers due date. A final cost not considered was overtime costs. For example, Kingsman, et al. (1989) argued that work load smoothing included not only changing due dates, but also adjusting capacity to meet fluctuating demand.

The post hoc analysis found that the SMOOTH planning system performed better than the UNCONTROLLED planning system for two reasons:

- The SMOOTH planning system produced less tardy work than the UNCONTROLLED planning system.
- The SMOOTH planning system resulted in a smaller time in system than the UNCONTROLLED planning system.

This suggests that the SMOOTH planning system reduces overloading of shop resources and allows jobs to flow through the shop faster, and on time.

The multiple comparisons of mean tardiness show that all the treatments with the SMOOTH planning system have fewer tardy jobs than treatments using the UNCONTROLLED planning system. The exception to this was treatment 1131, which used immediate release of jobs, the QUE "when" rule and the CMDD job dispatch rule. This treatment tied with an UNCONTROLLED planning system. However the planning system and the ORR rule interact to determine the tardiness of the job. The rank of the ORR rules is different for each type of planning system. With the SMOOTH planning system, the ORR rules were ranked MLC, MIL, CMS, and IMM. With the UNCONTROLLED planning system the ORR rules were ranked MLC, CMS, MIL and IMM.

The reason that the CMS rule performed better with the UNCONTROLLED planning system may be because the UNCONTROLLED planning system creates periodic over loads and under loads on the shop. The CMS trigger mechanism limits the impact of the load variation by keeping a steady amount of work in the shop. With the SMOOTH planning system, the job selection mechanism becomes increasingly important.

Future research is needed to investigate the hypothesis that as the planning system levels the work load more, the role of the job selection mechanism becomes more important. And, the role of the trigger mechanism becomes less important.

6.4 IMPLICATIONS OF SENSITIVITY ANALYSIS

The sensitivity analysis explored the role of the cost structure in determining the results of this experiment. This discussion of the sensitivity analysis concentrated on the following three topics:

1. the relationship of the cost structure and the planning system.
2. the relative performance of the ORR rules in each cost structure.
3. the relative performance of the "when" labor assignment rules in each cost structure.

Each of these topics is addressed separately.

6.4.1 COST STRUCTURE AND PLANNING

The UNCONTROLLED planning system performed better as the labor transfer costs increased and tardiness costs decreased. As tardiness costs decreased, the inventory costs were relatively more important. The multiple comparisons of the total inventory demonstrated that the treatments with the lowest inventory level were those using the UNCONTROLLED planning system. Because the UNCONTROLLED planning system put more jobs in the shop, there were fewer labor transfers.

The significance of this result was not clear, because the inventory level for the SMOOTH planning structure may have been artificially high because the same due date rule was used with both planning systems. This due date rule

would create unnecessarily long due dates for the jobs with the SMOOTH planning system which would be converted into extra inventory. Kingsman et al. (1989) suggested that a different due date setting mechanism should be used with the SMOOTH planning system to eliminate this variation.

A second issue discussed earlier was whether the SMOOTH planning system received an unfair advantage in the comparison to the UNCONTROLLED planning system because there was no cost penalty to push a job back or pull a job forward. The result of the sensitivity analysis must be interpreted with caution because of these two issues. However, there was some support for the selection of the planning system being based on the relative level of inventory costs. When inventory costs were high, an UNCONTROLLED planning system would be preferred because it resulted in lower inventory levels.

6.4.2 COST STRUCTURE AND ORR RULE

The sensitivity analysis demonstrated that the use of ORR rules resulted in higher total costs as labor transfer costs increased. With lower labor transfer costs, the MLC rule was the best rule. As labor transfer costs increased, the MIL rule tied with IMM as the best ORR rule. As labor transfer costs increased further compared to tardiness costs, IMM was the best ORR rule.

The MLC rule had many more labor transfers than the other ORR rules. However, these results may be skewed by the method used to count the labor transfers. The method used would handicap the MLC rule because every time a

laborer left a machine a transfer was recorded. But because the MLC rule kept queue sizes so small, operators frequently left their machine and were reassigned. The operator could be reassigned back to the machine they just left if a job was released from the pool for that machine. This would increase the number of labor transfers and bias the measurements against the MLC rule. Despite this counting problem, its likely that if labor transfer costs are high compared to tardiness costs, that ORR rules will do worse compared to IMM. This was expected because IMM creates larger queues in front of the work centers and depending on which "when" rule was used, there is the possibility of fewer labor transfers being needed.

These results suggested that the selection of an ORR rule for a factory must be done carefully.

6.4.3 COST STRUCTURE AND "WHEN" ASSIGNMENT

The sensitivity analysis demonstrated that for most cost structures the QUE "when" labor assignment rule was the best rule. The exceptions to this occurred where labor transfer costs were low. When the cost of a labor transfer was low, there was no preferred method of determining when to transfer the operator between machines. This was explained by noting that the QUE rule gives the lowest number of labor transfers because it keeps an operator at a machine until every job in the queue is completed. Any other rule would increase the number of labor transfers.

6.5 IMPLICATIONS FOR SCHEDULING THEORY

Figure 6-3 lists all the experimental factors in this experiment on the left. On the right it represents the three components of the total cost structure of the shop. The experimental factors on the left influence the total cost of the shop by influencing one or more of the cost components of the total cost structure.

This experiment demonstrated that the choice of the planning system influenced all three cost components of the total cost. This was illustrated in Figure 6-3 as a double line running from the planning system to each cost component. Relative to the UNCONTROLLED planning system, the SMOOTH planning system can decrease tardiness, increase inventory and increase the number of labor transfers. Because the due date rule was not adjusted to reflect the differences between the planning systems and there was no cost penalty for moving jobs forward and backward, the precise influence of the planning system could not be determined based on this experiment. However, both the multiple comparisons and the ANOVA suggested that the choice of the planning system is the primary determinant of shop performance.

This experiment suggested that the ORR job selection mechanism influences the level of tardiness in the shop and the level of inventory in the shop. This is illustrated in Figure 6-3 with a single line from the job selection mechanism to the tardiness and inventory components of cost. The selection mechanism did not seem to influence the number of labor transfers. The best job selection

mechanism was the MLC mechanism. The MLC mechanism reduced inventory and tardiness compared to the other ORR rules.

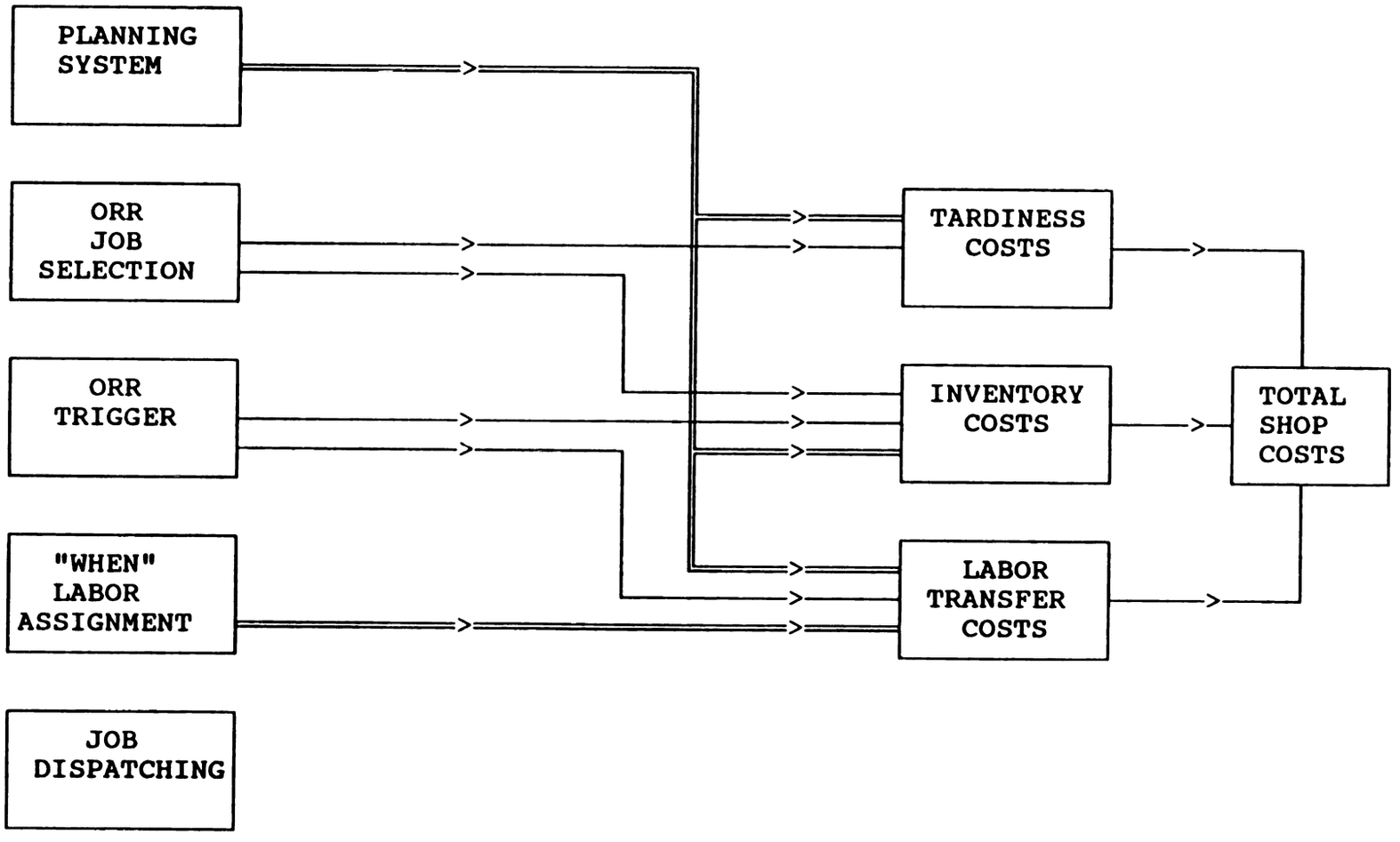
The trigger mechanism influenced the time the job spent waiting in the shop and consequently the level of WIP inventory. As jobs were held off the shop floor by the trigger mechanism, the number of labor transfers increased. The influence of the ORR rule is illustrated in Figure 6-3 which shows a single line running from the trigger mechanism to the inventory and labor transfer costs. The best trigger mechanism released a job only when an operator was idle.

This experiment indicated that the "when" labor assignment rule was relatively unimportant except when the cost structure included high labor transfer costs. This agreed with past research, which found that the "when" labor assignment rule was significant only if labor transfer delay times were considered (Treleven, 1989). The "when" labor assignment rule did influence the number of labor transfers as illustrated in Figure 6-3 with the double line.

The job dispatching rule was not found to influence any cost components in the shop if ORR rules were used. This supports Wight (1970), Melnyk et al. (1985) and others who theorized that the importance of job dispatching rules was overstated in scheduling research because most research did not include the planning system and the ORR mechanism in their decision.

FIGURE 6-3

IMPACT OF PLANNING AND SCHEDULING



6.6 FUTURE RESEARCH DIRECTIONS

This experiment suggested several directions for future extensions of research:

- First, the planning system needs to be modeled in more detail. This detail should include the cost of moving jobs forward and backward. It should also include different methods of capacity control and feedback mechanisms to set the due date as suggested by Melnyk and Ragatz (1989). This experiment established that is important because the planning system influences tardiness, inventory and the number of labor transfers. Also, the choice of the planning system interacts differently with each SFC rule.
- Second, further research is needed to investigate the impact of the trigger mechanism on shop performance. This research found that the trigger mechanism influences the level of WIP inventory and the number of labor transfers. More research is needed to determine what information should be used in the trigger mechanism. Should the trigger consider the overall shop load, the balance of work in the shop or the inventory at a critical work center. Prior researchers have not examined the trigger mechanism separately from the job selection mechanism (Baker, 1984; Ragatz and Mabert, 1988; Melnyk and Ragatz, 1989).

- Additional investigation of finite loading ORR rules is needed. The MLC rule was a heuristic to conduct finite loading of the shop. The heuristic heavily weighted the load at the gateway work center in its calculations. Another loading mechanism may be more effective if it weighted the critical work center more. Finite loading mechanisms were considered in prior research (e.g. Ragatz, 1985) but not with the trigger mechanism used here. It is possible that other finite loading mechanisms will perform well if used with different trigger mechanisms. This research is important because the job selection mechanism influences finished goods inventory and tardiness.

- Research is needed to determine if the effectiveness of ORR rules depends on the structure of the shop. This research was placed in a random job shop. How will the rules perform in job shops with dominate work flows. With a dominant work flow the ability of the job selection mechanism to distinguish between critical and non critical work centers could be important. With dominant flow, a job selection mechanism which keeps work in front of the critical work center to prevent idle time at that center may perform very well. This approach is described by Umble and Srikanth (1990, pp.133-177).

- More research about the interaction of the job selection mechanisms and the trigger mechanisms is needed. Questions to be answered are whether the job selection mechanism needs to use available information, or will a simple job selection mechanism perform well with a trigger that uses information. For example if the trigger mechanism kept the gateway queue at zero could the job selection mechanism be as simple as FCFS. Prior research with these simple job selection mechanisms (Baker, 1984; and Ragatz, 1985) did not find the simple job selection mechanisms to be beneficial. But these simple mechanisms were used with simple trigger mechanisms.
- Research is needed to determine if the effectiveness of the "when" labor assignment rule depends on the type of work flow in the shop. In a dominant flow shop there would be more critical work centers. The CLR rule might perform better because it was formulated to move resources to the critical work center. Prior research about labor assignment rules in dominant flow job shops (e.g. Holstein and Berry, 1972) did not consider the "when" rule. Umble and Srikanth (1990) suggested that shop performance improves when resources can be moved quickly to critical work centers.
- Additional research is needed about the implications of smoothing the workload sent from planning to the shop. The SMOOTH

workload used in this research is an extremely smooth workload. It is important to establish how the results of this research change for different degrees of smoothing. There are a variety of rough cut capacity planning techniques which can be tested in this setting as described in Melnyk, Ragatz and Fredendall (1990).

- Finally, this experiment considered only whether certain types of information were used. It did not consider how the information was used. Additional work needs to be done to determine the best methods of utilizing the available information for each SFC rule. This research considered information as belonging to one of three general classes (SHOP, POOL and LABOR). Actually there are many divisions of information in each of these classes. For example SHOP information includes information about particular jobs in a queue and also the total number of jobs in the shop.

6.7 CONCLUSION

Briefly, the major findings of this research were:

- the trigger mechanism influences the time in system, the WIP inventory levels and the number of labor transfers.
- The job selection mechanism influences the amount of tardiness and FGS inventory levels.

- The planning system influences the tardiness, the inventory level and the number of labor transfers.
- The "when" labor assignment rule influenced only the number of labor transfers.
- And finally, the job dispatching rule was not significant if ORR rules were used.

These findings support the hypothesis that research about job shop scheduling must consider many elements of the shop simultaneously because they all interact (McKay et al., 1988). This suggests that future research needs to be directed towards the development of a theory which integrates the separate elements of planning and SFC into a manufacturing system. Frameworks to do this (e.g. Melnyk et al., 1985) have been proposed, but there has been little experimental research done to develop them further.

This research found that information can play an important role in scheduling the shop, but that its use is complex. Using additional sources of information in each SFC rule did not always improve performance. Using LABOR information in the ORR rule improved performance, but using POOL information in the "when" labor assignment and job dispatching rules did not improve performance.

This experiment suggested that the shop's cost structure determines the relative importance of the planning system, and the SFC rules. As tardiness costs decreased the UNCONTROLLED planning system performed better than the

SMOOTH planning system. As LABOR transfer costs increased the ORR rules did not perform as well as immediately releasing jobs.

Additional research is needed to investigate the interaction of the planning system and the SFC rules. The separate impact of the trigger mechanism and the job selection mechanisms also need to be investigated. It is important to determine the influence of the shop structure and the work flow of the shop on the ORR rules and the "when" labor assignment rules. Finally, it is important to determine whether there is a role for job dispatching if used with ORR rules.

APPENDIX A

APPENDIX A

ATTRIBUTE STRUCTURE AND VARIABLE DICTIONARY

PART 1: ATTRIBUTE STRUCTURE

ATTRIBUTES		SUBROUTINE
A(1)	FIRST MACHINE IN ROUTING	[ARRIVE]
A(2)	SECOND MACHINE IN ROUTING	[ARRIVE]
A(3)	THIRD MACHINE IN ROUTING	[ARRIVE]
A(4)	FOURTH MACHINE IN ROUTING	[ARRIVE]
A(5)	FIFTH MACHINE IN ROUTING	[ARRIVE]
A(6)	PROCESSING TIME FIRST MACHINE	[ARRIVE]
A(7)	PROCESSING TIME SECOND MACHINE	[ARRIVE]
A(8)	PROCESSING TIME THIRD MACHINE	[ARRIVE]
A(9)	PROCESSING TIME FOURTH MACHINE	[ARRIVE]
A(10)	PROCESSING TIME FIFTH MACHINE	[ARRIVE]
A(11)	NUMBER OF OPERATIONS	[ARRIVE]
A(12)	CURRENT OPERATION NUMBER (A(1) TO A(5))	[ARRIVE, DEPART]
A(13)	JOB DUE DATE	[ARRIVE]
A(14)	RELEASE DATE	[DEPART, IMM,CMS,MIL]
A(15)	DISPATCHING PRIORITY	[DISPATCH, PRIOR,QRANK]
A(16)	TIME JOB ARRIVED IN SYSTEM	[ARRIVE]
A(17)	TIME JOB STARTS ON SHOP FLOOR	[IMM,MIL, SELECT,RELEASE]
A(18)	TIME JOB DEPARTS MACHINE	[DEPART]
A(19)	TOTAL PROCESSING TIME	[ARRIVE]
A(20)	REMAINING PROCESSING TIME	[ARRIVE, DEPART]

A(21) TO A(30) USED ONLY IN DEGUGGING

PART 2: VARIABLE DICTIONARY

X( ) ARRAY VARIABLES		SUBROUTINE
X(1)	ORDER REVIEW RELEASE RULE IN EFFECT 1 = IMMEDIATE RELEASE 2 = MODIFIED INFINITE LOADING (MIL) 3 = CRITICAL MACHINE SELECTION (CMS) 4 = MODIFIED LOAD CONVERSION (MLC)	EXPERIMENT
X(2)	"WHEN"+LABOR ASSIGNMENT RULE IN EFFECT 1 = QUE RULE 2 = JOB RULE 3 = CLR RULE	EXPERIMENT
X(3)	JOB DISPATCH RULE IN EFFECT 1 = FCFS 2 = MODD 3 = CMDD	
X(4)	NUMBER OF JOBS ARRIVING THIS PERIOD	[ARRIVE]
X(5)	NUMBER OF JOBS OR BATCHES TO TRUNCATE	[PRIME, RECEXT]
X(6)	k_1 IN MIL RULE: 0 < k	EXPERIMENT
X(7)	k_2 IN MIL RULE: 0 < k	EXPERIMENT
X(8)	k IN DUE DATE RULE 0 < k	EXPERIMENT
X(9)	LENGTH OF PLANNING PERIOD IN MLC RULE	EXPERIMENT
X(10)	k IN MLC RULE $0 \leq k \leq 1$	EXPERIMENT
X(11)	STATUS WORK CENTER 1	[DEPART,DISPATCH]
X(12)	STATUS WORK CENTER 2	[DEPART,DISPATCH]
X(13)	STATUS WORK CENTER 3	[DEPART,DISPATCH]
X(14)	STATUS WORK CENTER 4	[DEPART,DISPATCH]
X(15)	STATUS WORK CENTER 5	[DEPART,DISPATCH]
X(16)	STATUS WORK CENTER 6 FOR X(11) .. X(16) BOTH IDLE = 0; ONE BUSY = 1; BOTH BUSY = 2 X(11) ..X(19) USED IN COMPLEX TO GIVE START POINT	[DEPART,DISPATCH]
X(17)	CURRENT UTILIZATION OF MACHINES	[RECINV]
X(18)	CURRENT UTILIZATION OF LABOR	[RECINV]
X(19)	CUMULATIVE UTILIZATION MACHINES	[RECINV]
X(20)	CUMULATIVE UTILIZATION LABOR	[RECINV]
X(21)	CUMULATIVE TIME AVERAGE JOBS ON FLOOR	[RECINV]
X(22)	CURRENT NUMBER JOBS ON FLOOR	[RECINV]
X(23)	TIME NUMBER JOBS LAST RECORDED IN FILE	[RECEXT]
X(24)	NUMBER OF RELEASES IN SHIFT	[IMM,MIL RELEASE,SELECT,SHIFT]
X(25)	TIME DEPART INFORMATION LAST RECORDED	[RECINV, RECEXT]
X(26)	HOURS OF INV SINCE LAST JOB EXIT	[RECINV, RECEXT]
X(27)	# LABOR TRANSFERS SINCE LAST JOB EXIT	[WHEN, RECEXT]
X(28)	CUMULATIVE # TARDY HOURS	[RECEXT]

X(29) STATUS OF LABORERS [DEPART,DISPATCH]
NUMBER FREE = 6 INITIALLY
X(30) TIME RUN BEGAN [PRIME]
X(31) MAXIMUM NUMBER LABORERS ALLOWED [PRIME]
X(32) NUMBER OF JOBS IN FGS INVENTORY [DEPART,EXITS]
X(33) CURRENT FGS INVENTORY [DEPART,EXITS]
X(34) NUMBER OF POINTS FOR COMPLEX/FIBONNACI [FUNC,FIBON]
X(35) NUMBER OF WEEKS IN ONE BATCH [PRIME, FUNC, FIBDAT]
X(36) NUMBER OF WEEKS OF DATA COLLECTED [PERIOD]
X(37) COMPLETIONS SINCE LAST RECORDED [SHOP, RECEXT]
X(38) NUMBER JOBS TARDY SINCE LAST RECORDED [SHOP, RECEXT]
X(39) CURRENT WIP IN SHOP [DEPART]
X(40) NUMBER OF WORK CENTERS [PRIME]
X(41) NUMBER OF BATCHES TO COLLECT FOR SAMPLE [PRIME]
X(42) COST OF TARDINESS EXPERIMENT
X(43) COST OF INVENTORY EXPERIMENT
X(44) COST OF LABOR TRANSFERS EXPERIMENT
VALUE OF X(42), X(43) & X(44) CREATE COST STRUCTURE
X(45) TYPE OF SHOP BEING SIMULATED [PRIME]
1) M\M\1; 2) M\M\2; 3) 2 MACHINE LABOR CONSTRAINT
4) 5 MACHINE LABOR CONSTRAINT; 5) 5 WC (2 MACH) LAB CONSTRAINT; 6) 6 WC (2 MACH) LABOR CONSTRAINT
EARLY RELEASE JOBS ALLOWED FOR X(45) = 1,2,3: EARLY
RELEASE OF JOBS NOT ALLOWED FOR X(45) = 4,5,6
DATA COLLECTED BY INDIVIDUAL JOB FOR X(45) =1,2,3:
DATA COLLECTED AS BATCHES FOR X(45) = 4,5,6
X(46) HALF WIDTH FOR CONFIDENCE LEVEL [PRIME]
X(47) DETERMINE STARTING RUN NUMBER [PRIME]
X(48) NUMBER OF BATCHES COLLECTED [PRIME, PERIOD,FUNC,FIBTAB]
X(49) OPTIMIZATION ALGORITHM [PRIME]
0 = NO OPTIMIZATION: 1 = COMPLEX: 2= FIBONACCI
3 = COMPLEX FOR MLC: 4 COMPLEX START MIDDLE, X6, X7:
5 = COMPLEX MIDDLE X9, X10: 10=NO COMPLEX, REAL DATA
FOR ALL X(49) LE 10, CONFID. LEVEL IS 0.05
X(50) ARRIVAL PATTERN OF JOBS TO SHOP [PRIME]
1 = UNIFORM MEAN 54 JOBS
2 = POISSON MEAN 54 JOBS

SIMAN TALLIES, COUNTERS AND DSTAT VARIABLES

COUNTERS

1	INDIVIDUAL JOBS COMPLETED	(INF)	[RECEXT]
2	COMPLEX OR FIBONACCI COMPLETE	(1)	[FIN,FUNC]
3	TOTAL JOBS COMPLETED	(INF)	[RECEXT]
4	NUMBER OF JOBS LATE	(INF)	[RECEXT]
5	NUMBER OF BATCHES COLLECTED	(INF)	[PERIOD]
6	NUMBER OF LABORERS TRANSFERRED	(INF)	[WHEN]

TALLIES

1	WORK INTO SHOP BY WEEK	[ARRIVE]
2	QUEUE TIME	[RECEXT]
3	FLOW TIME	[RECEXT]
4	POOL TIME	[RECEXT]
5	NUMBER OPERATIONS IN JOB	[ARRIVE]
6	WORK CONTENT OF OPERATIONS	[ARRIVE]
7		
8		
9	WORK CONTENT LATE	[RECEXT,PERIOD]
10	NUMBER LABOR TRANSFERS	[WHEN,PERIOD]

DSTAT VARIABLES

1	JOBS ON FLOOR X(22)
2	CURRENT WIP IN SHOP X(39)
3	CURRENT FGS INVENTORY X(33)
4	MACHINE UTILIZATION X(17)
5	LABOR UTILIZATION X(18)

FORTRAN COMMON BLOCK VARIABLES

COMMON/CRIT/JRANK(6),SHIFTEND,RANK(6),SHIFTBEG,AVGWL

JRANK	ARRAY TO INDICATE RELATIVE CRITICALITY OF WC
SHIFTEND	TIME CURRENT SHIFT WILL END
RANK	ARRAY INDICATES LOAD OF CORRESPONDING WC IN JRANK
SHIFTBEG	TIME CURRENT SHIFT STARTED
AVGWL	AVERAGE TOTAL WORKLOAD OF WC AT START OF SHIFT

SIMAN FILES		PRIORITY RULE
1	ORDER POOL	LVF(14)
2	WORK CENTER 1 QUEUE	LVF(15)
3	WORK CENTER 2 QUEUE	LVF(15)
4	WORK CENTER 3 QUEUE	LVF(15)
5	WORK CENTER 4 QUEUE	LVF(15)
6	WORK CENTER 5 QUEUE	LVF(15)
7	WORK CENTER 6 QUEUE	LVF(15)
8	TEMPORARY FILE FOR SORTING	LVF(14)
9	TEMPORARY IN DISPATCH FOR REAL JOBS WHILE COPIES OF JOBS PROCESSED	LVF(18)

SIMAN EVENT CALENDAR

1.	JOB ARRIVAL FROM PLANNING (40 HRS.)
3.	BEGINNING OF A SHIFT (8 HRS.)
3.	JOB DEPARTS A MACHINE
4.	JOB LEAVES FINISHED GOODS INVENTORY
5.	JOB GO INTO SHOP FROM ORDER POOL (MIL ONLY)
6.	COLLECT DATA FOR RELEASE AND ARRIVAL RECORD
7.	COLLECT BATCH DATA

USER SUBROUTINE CALLS

CALLING	CALLED
PRIME	
EVENT	ARRIVE, SHIFT, DEPART, EXITS, RELEASE
ARRIVE	CRITICAL, ORDER
DEPART	WHEN, SELECT, RECINV, RECEXT
WHEN	WHERE, DISPATCH
EXITS	RECINV, RECEXT
RELEASE	
ORDER	IMM, MIL, CMS, MLC
IMM	WHERE
CMS	SELECT
MIL	WHERE
MLC	SELECT
DISPATCH	
SELECT	WHERE
SHIFT	CRITICAL, ORDER
WRAPUP	
RECINV	
RECEXT	OVER
WHERE	DISPATCH
OVER	WRAPUP

APPENDIX B

APPENDIX B

SIMULATION CODE

PART 1: SIMAN EXPERIMENT PROCESSOR CODE

```
BEGIN;
PROJECT, DETERMIN, FREDENDALL, 09/15/1990;
DISCRETE,300,20,9;
;PARAM 1=UN(MIN,MAX): 2=NOP: 3=PROC: 4=MACH:6=POI(MEAN)
PARAMETERS:1,53,55:2,3,6:3,1.0:4,1,7:5,1,1:6,54.0:7,1.111:
     8,4.0,5.0,6.0:9,200,500;
;REPLICATE, 2,0.0,,NO,NO,0.0;
REPLICATE, 25,0.0,,NO,NO,0.0;
COUNTERS:1,JOBS COMPLETED,,YES:2,BATCHES COMPLETED,1,YES:
     3,TOTAL JOBS COMPLETED,,YES:4,NUMBER JOBS LATE,,YES:
     5,NUMBER BATCHES COMPLETED,,YES:6,NUMBER TRANSFERS,,YES;
;TRACE,0.0,200.0,,NQ(1),NQ(2),NQ(3),NQ(4),NQ(5),NQ(6),NQ(7),
;      X(11),X(12),X(13),X(14),X(15),X(16);
RANKINGS:1,LVF(14):2-7,LVF(15):8,LVF(14):9,LVF(18):10,LVF(18);
;X(34) INDICATES NUMBER OF POINTS INVESTIGATED WITHIN COMPLEX
;X(40) NUMBER WC:(41) NUMBER OF SAMPLES OF SIZE X(8) COLLECT
;X(42) COLLECT DATA AS JOB (1) or BATCH SIZE ONE WEEK (2)
;      X(41) NUMBER OF SAMPLES IN ONE BATCH
;      X(43) CONFIDENCE LEVEL (ALPHA)
;X(45) TYPE OF SHOP BEING SIMULATED: 1 = M/M/1: 2= M/M/2:
; 3= 2 MACH WITH LABOR CONSTRAINT: 4= 5 MACH. NO LAB CONSTR
; 5 = 5 WORK CENTER (2 MACH EACH) & LABOR CONSTRAINT (PARK)
; 6 = 6 WORK CENTER (2MACH) & LABOR CONSTRAINT (LDF)
;X(46) DESIRED HALF WIDTH FOR CONFIDENCE INTERVAL
;X(48) NUMBER OF SAMPLES COLLECTED TOWARDS X(41)
;X(49) OPTIMIZING ALGORITHM IN USE: 0 = NONE;
; 1 = COMPLEX FOR MIL & CMS; 3 = COMPLEX FOR MLC
; 4 = RUNS FOR TRIAL SAMPLES; 10 = RUNS FOR DISSERTATION
DSTAT:1,X(22), JOBS ON FLOOR: 2,X(39),CURRENT WIP IN SHOP:
    3,X(33), CURRENT FGS INV IN SHOP:4,X(17),MACHINE UTIL:
    5,X(18),LABOR UTIL;
TALLIES:1,WORK INTO SHOP:2,QUEUE TIME:3,FLOW TIME:4,POOL TIME:
  5,NUM OPN PER JOB: 6, WORK PER OPN: 7, NUM JOBS COMPLETE:
  8, NUM JOBS LATE: 9, WORK CONTENT LATE:10,NUM LABOR TRANS:
  11,HOURS TARDY :12,LABOR TRANS :13,AVG INVEN PER WK:
  14,NUMBER TRANSFERS PER WEEK;
INITIALIZE, X(4)=0,X(8) =9.5,X(29)=6,X(31)=6,X(35)= 10,
  X(41)= 10,X(48)=1,X(50)= 1,
  X(1)= 3,X(2)= 1,X(3)= 3,X(36)=1,X(49)= 1,
  X(46)=100,X(47)=  0,
  X(42)=1,X(43)=.1,X(44)=.1,x(45)=6,X(6)= 50,X(7)= 50,
  X(9)= 100,X(10)= .1;
```

```
; CHECKLIST FOR DISSERTATION: 1) X(49) = 10: 2) X(35) = 90
; X(41) = 1: X(47) = 0: REPLICATE = 100: SEED 4 IS RANDOM
; MAIN EXP: X(42)=2,X(43)=.1,X(44)=.1 COSTS TARDY/INV/TRANS
; SENS. #1:X(42)=1,X(43)=.1,X(44)=.1 COSTS TARDY/INV/TRANS
; SENS. #2:X(42)=1,X(43)=2,X(44)=.1 COSTS TARDY/INV/TRANS
; SENS. #3:X(42)=1,X(43)=.1,X(44)=1 COSTS TARDY/INV/TRANS
;SEEDS:1,21913,C: are for runs 46 - 65 (20)
;SEEDS:1,21913,C:2,10433,C:3,16794,C:4,30703,C;
;SEEDS:1,16026,C:are for runs 66 - 85(20)
;SEEDS:1,16026,C:2,28775,C:3,27453,C:4,18315,C;
;SEEDS:1,10951, are for runs 86 to 100 (15)
;SEEDS:1,10951,C:2,32109,C:3,11850,C:4,17422,C;
;SEEDS:1,18392,C: ARE FOR RUNS 26-45 (20)
;SEEDS:1,18392,C:2,18144,C:3,14099,C:4,28616,C;
;SEEDS:1,30948 ARE FOR FIRST 25 RUNS
SEEDS:1,30948,C:2,31948,C:3,11155,C:4,29111,C;
END;
```

SIMAN MODEL PROCESSOR CODE

```
BEGIN;
END;
```

PART 2: FORTRAN CODE FOR USER WRITTEN SUBROUTINES

```
SUBROUTINE PRIME
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J  ,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
COMMON/QUART/PER(400,2),NTH,TOT_COST
*     JBAT IS ORIGINAL BATCH SIZE FOR X(35)
*     REL DETERMINES HOW FREQUENTLY DATA COLLECTED
      JBAT = X(35)
      REL = 40
      X(30) = TNOW
      SHIFTBEG = TNOW
      SHIFTEND = TNOW + 8.0
      NO = 11
      NI = 12
      NJ = 13
      NL = 14
      KJ = 15
      DO 18 I = 1, 400
        PER(I,1) = 0.0
        PER(I,2) = 0.0
18     CONTINUE
      IF ((NRUN .GT. 1) .AND. (X(49) .EQ. 10)) GOTO 899
      IF ((NRUN .GT. 1) .AND. (X(49) .LT. 10)) GOTO 15
      CALL CREATE(JOB)
      CALL SCHED(JOB,1,0.0)
      CALL CREATE(JOBTWO)
      CALL SCHED(JOBTWO,2,8.0)
      CALL CREATE(JOBTHRE)
      CALL SCHED(JOBTHRE,6,REL)
*     SET UP MACHINES AND WORK CENTERS:
*     ESTABLISH # MACH & LABORERS
*     X(40) NUMBER OF WORK CENTERS
      IF (X(45) .LT. 4.0) THEN
*       M/M/1 OR M/M/2 OR TAKACS TYPE SHOP
*       INV. RELEASED WHEN WORK COMPL: NOT HELD UNTIL DD
        IF (X(45) .LT. 3.0) THEN
*           M/M/1 OR M/M/2 TYPE SHOP
          X(40) = 1.0
          DO 99 I = 2,6
            X(10 + I) = 2.0
99         CONTINUE
          IF (X(45) .EQ. 1.0) THEN
            X(11) = 1.0
          ELSE
            X(11) = 0.0
          ENDIF
```

```
        ELSE
          X(40) = 2.0
          DO 107 I = 3,6
            X(10 + I) = 2.0
107         CONTINUE
          DO 109 I = 1,2
            X(10 + I) = 1.0
109         CONTINUE
        ENDIF
      ELSE
*       EARLY RELEASE INVENTORY NOK: INV HELD UNTIL DD
*       COLLECT STATISTICS BY BATCH
        CALL CREATE(JOBFOUR)
        CALL SCHED(JOBFOUR,7,40.0)
        IF (X(45) .EQ. 4.0) THEN
*         SIMULATES RAGATZ (1985) SHOP
          X(16) = 2.0
          DO 121 I = 1,5
            X(10 + I) = 1.0
121         CONTINUE
          X(40) = 5.0
        ENDIF
        IF (X(45) .EQ. 5.0) THEN
*         SIMULATES PARK (1987) SHOP
          X(16) = 2.0
          X(40) = 5.0
        ENDIF
        IF (X(45) .EQ. 6.0) THEN
*         THIS EXPERIMENT
          X(40) = 6.0
        ENDIF
      ENDIF
      IF (X(49) .EQ. 10) GOTO 899
15    IF (NRUN .EQ. 1) THEN
        X(6) = 30
        X(7) = -10
        X(9) =  25
        X(10) = .6
      ENDIF
      IF (NRUN .EQ. 2) THEN
        X(6) = 10
        X(7) = 10
        X(9) = 100
        X(10) = .8
      ENDIF
      IF (NRUN .EQ. 3) THEN
        X(6) = 30
        X(7) = 0
        X(9) = 50
```

```
      X(10) = .7
    ENDIF
    IF (NRUN .EQ. 4) THEN
      X(6) = 30
      X(7) = 20
      X(9) = 20
      X(10) = .4
    ENDIF
    IF (NRUN .EQ. 5) THEN
      X(6) = 10
      X(7) = -5
      X(9) = 50
      X(10) = .9
    ENDIF
    IF (NRUN .EQ. 6) THEN
      X(6) = 20
      X(7) = -1
      X(9) = 90
      X(10) = .1
    ENDIF
    IF (NRUN .EQ. 7) THEN
      X(6) = 10
      X(7) = -10
      X(9) = 30
      X(10) = .9
    ENDIF
    IF (NRUN .EQ. 8) THEN
      X(6) = 9
      X(7) = 9
      X(9) = 31
      X(10) = .1
    ENDIF
    IF (NRUN .EQ. 9) THEN
      X(6) = -5
      X(7) = 7
      X(9) = 31
      X(10) = .1
    ENDIF
899   IF (NRUN .EQ. 1) CALL BEGN
    WRITE(NO,*) 'SUB PRIME: BEGIN RUN NUMBER=',NRUN
    IF ((NRUN .GT. 1).AND.(X(49) .EQ. 10)) CALL CLEAR
902   FORMAT(1X,10(1X,F6.2))
901   IF (NRUN .EQ. 1 ) THEN
       DO 929   I = 11,19
         X(I) = 0.0
929      CONTINUE
    ENDIF
    AC = X(6)
    AD = X(7)
```

```
      AE = X(9)
      AF = X(10)
      IF (NRUN .EQ. 1) THEN
IF((X(49).EQ.1.0).OR.(X(49).EQ.3.0).OR.(X(49).EQ.10))THEN
          CL = 0.05
        ENDIF
WRITE(NO,*)'ORR,LAB,DISP,X6, X7, X9,   X10,  SIZE,CL, PLAN'
WRITE(NO,902)X(1),X(2),X(3),AC,AD,AE,AF,X(35),CL,X(50)
WRITE(NJ,*)'ORR,LAB,DISP,X6,X7,X9,X10,  SIZE,CL, PLAN'
WRITE(NJ,902)X(1),X(2),X(3),AC,AD,AE,AF,X(35),CL,X(50)
WRITE(NL,*)'ORR,LAB,DISP,X6,X7,X9,X10,  SIZE,C.L., PLAN'
WRITE(NL,902)X(1),X(2),X(3),AC,AD,AE,AF,X(35),CL,X(50)
      ENDIF
      IF (X(49) .EQ. 1.0) CALL COMP1
      IF (X(49) .EQ. 3.0) CALL COMP1
*      PRINT *, 'END OF SUB PRIME'
*      WRITE(NO,*) 'END OF SUB PRIME: X(49)=',X(49)
915    RETURN
      END

*-------------------------------------------------------
      SUBROUTINE BEGN
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J  ,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
COMMON/QUART/PER(400,2),NTH,TOT_COST
      X(5) = 40
21     FORMAT(1X,8(F5.2),'BEGIN COMPLEX RUN =',1X,I2)
OPEN(UNIT=10,FILE='A:\REL',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT=11,FILE='A:\DAT1',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT=12,FILE='A:\DAT2',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT=13,FILE='A:\DAT3',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT=14,FILE='A:\DAT4',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT=15,FILE='A:\DAT5',FORM='FORMATTED',STATUS='NEW')
WRITE(10,21) X(1),X(2),X(3),X(6),X(7),X(8),X(10),X(50),NRUN
WRITE(10,*)'TME,ARRIV,POL,REL,CMPL,FGS,FLOR,LATE,TRAN,IDL,INV'
WRITE(NO,*) 'COST = 2*TARDY + .1*COMB_INV + .1*TRANSFER'
100    FORMAT(1X,4(2X,F8.2))
      IF (X(49) .LE. 10) CL = 0.05
      WRITE(NI,*) 'ORR, LABOR, DISP, PLAN'
      WRITE(NI,*) 'BATCH SIZE, CONF. LEVEL, DEG. FREEDOM'
      WRITE(NI,313) X(1),X(2),X(3),X(50)
      WRITE(NI,314) X(35),CL,X(48)
WRITE(NI,*) 'COST = 2*TARDY + .1*COMB_INV + .1*TRANSFER'
WRITE(NI,*)'CAT,NRUN,X(1),X(2),X(3),X(50),TARD,CMB_INV,TRAN,      COST'
313    FORMAT(1X,4(2X,F6.2))
314    FORMAT(1X,3(2X,F8.2))
*      WRITE(NJ,*) 'ORR, LABOR, DISP, PLAN'
*      WRITE(NJ,*) 'BATCH SIZE, CONF. LEVEL, DEG. FREEDOM'
*      WRITE(NJ,313) X(1),X(2),X(3),X(50)
```

```
*       WRITE(NJ,314) X(35),CL,X(48)
        IF(X(49).EQ.10) THEN
         WRITE(NJ,*)'CAT,NRUN,X(1),X(2),X(3),X(50)'
WRITE(NJ,*)'    (CON.) TARD HR/JOB TARD,FGS_INV,WIP_INV'
         WRITE(NJ,*) '   (con.)Q_POOL,Q_SHP,FLOW,FLOW9,TARD9'
*        WRITE(NL,*) 'ORR, LABOR, DISP, PLAN'
*        WRITE(NL,*) 'BATCH SIZE, CONF. LEVEL, DEG. FREEDOM'
         WRITE(NL,*)'CAT,NRUN,X(1),X(2),X(3),X(50)'
WRITE(NL,*)'(CON.) MACH UTIL,LABOR UTIL,PCT_TARDY,COMP'
         WRITE(KJ,*)'CAT,RUN,ORR,WHEN,DISP,PLAN'
         WRITE(KJ,*)'   TARDY/WK,INV/WK,TRANSFER/WK,COST/WK'
WRITE(KJ,*)'   HR TARD/TRD JOB, QPOOL/JOB,QSHOP/JOB,MEAN FLOW'
         WRITE(KJ,*)'   90th FLOW, 90th TARDY, WIP/WK,FGS/WK'
         WRITE(KJ,*)'   UTIL_MCH,UTIL_LAB,PCT_TARDY'
        ENDIF
        WRITE(NO,400) X(5)
400     FORMAT(1X,'TRUNCATE',1X,F4.0,'WEEKS IN 1ST REPLIC. ')
*       DETERMINE 90TH PERCENTILE OF FLOW & TARDINESS
        NTH = INT(X(35)*54*.9)
        IF (NTH .GT. 400) NTH = 400
        RETURN
        END
*-------------------------------------------------------
        SUBROUTINE EVENT(JOB,N)
*       THERE ARE 8 POSSIBLE EVENTS
*       1)JOB ARRIVES FROM PLANNING (40hrs)
*       2)BEGINNING OF SHIFT (8hrs)
*       3)JOB DEPARTS MACHINE
*       4)JOB EXITS FGS INV.
*       5)JOB IS RELEASED FROM ORDER POOL (MIL)
*       6)RECORD INFORMATION ABOUT RELEASES & ARRIVALS
*       7)COLLECT WEEKLY DATA
*        IF (TNOW .GT. 40.0) WRITE(NO,*)'EVENT NUMBER=',N
        GOTO (1,2,3,4,5,6,7),N
1       CALL ARRIVE(JOB)
        RETURN
2       CALL SHIFT(JOB)
        RETURN
3       CALL DEPART(JOB)
        RETURN
4       CALL EXITS(JOB)
        RETURN
5       CALL RELEASE(JOB)
        RETURN
6       CALL SHOP(JOB)
        RETURN
7       CALL PERIOD(JOB)
        RETURN
        END
```

```
*==================
*     THIS GROUP OF SUBROUTINES IS CONCERNED WITH ARRIVAL
*==================
      SUBROUTINE ARRIVE(NEWJOB)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      INTEGER*2 NROUTE(5)
      SHIFTBEG = TNOW
      SHIFTEND = SHIFTBEG + 8.0
*     SCHEDULE NEXT BATCH OF JOBS FROM PLANNING
*      PRINT *,'BEGIN SUBROUTINE ARRIVE: TNOW =',TNOW
      IF (X(45) .LT. 4.0) THEN
*        M\M\1 OR M\M\2 TYPE OF SHOP OR TAKAS LAB CONSTR.
         DT = EX(7,1)
         N = 1
         X(4) = X(4) + 1.0
      ELSE
         DT = 40.0
         N = 2
         IF (X(50) .EQ. 1.0) THEN
            NUM = INT(UN(1,1))
         ELSE
            NUM = INT(NP(6,1)+0.5)
         ENDIF
         X(4) = X(4) + REAL(NUM)
      ENDIF
      CALL SCHED(NEWJOB,1,DT)
*     WRITE DATA TO FILE FOR PAST WEEK: 1ST UPDATE TO WK END
      CALL RECINV
      GOTO (1,2),N
1     CALL CREATE(NEWJOB)
*      PRINT *,'STARTING SECTION FOR M\M\s OR TAKACS: N=',N
      CALL SETA(NEWJOB,16,TNOW)
      CALL SETA(NEWJOB,11,1.0)
      CALL SETA(NEWJOB,12,1.0)
      IF (X(45) .LT. 3.0) THEN
         RMACH = 1.0
      ELSE
*        TAKACS 2 MACHINE SYSTEM: TAKE INTEGER FIRST !!
         MACH = INT(UN(9,4))
         RMACH = REAL(MACH)
      ENDIF
      CALL SETA(NEWJOB,1,RMACH)
      PROC = EX(3,3)
      CALL SETA(NEWJOB,6,PROC)
      CALL TALLY(1,PROC)
      DD = X(8)*NOP + TNOW
      CALL SETA(NEWJOB,13,DD)
      CALL INSERT(NEWJOB,1)
```

```
      GOTO 100
*     CREATE JOBS FOR THIS WEEK
2     WEEK_WRK = 0.0
*     SECTION FOR RAGATZ, PARK OR FREDENDALL
      DO 10 I = 1,NUM
        DO 20 L = 1,5
          NROUTE(L) = 0
20       CONTINUE
        CALL CREATE(NEWJOB)
*       DETERMINE ATTRIBUTES OF JOB
        CALL SETA(NEWJOB,16,TNOW)
        NOP = INT(UN(2,2))
*       RECORD NUMBER OF OPERATIONS
        CALL SETA(NEWJOB,11,REAL(NOP))
        CALL TALLY(5,REAL(NOP))
        CALL SETA(NEWJOB,12,1.0)
        TOT_PROC = 0.0
        DO 30 K = 1,NOP
*         DETERMINE PROCESSING TIME/OPERATION
21         PROC = EX(3,3)
          CALL TALLY(6,PROC)
*         DETERMINE MACHINE ASSIGNMENT
25         IFLAG = 0
          MACH = INT(UN(4,4))
          RMACH = REAL(MACH)
          DO 40 L = 1,K
              IF (MACH .EQ. NROUTE(L)) IFLAG = 1
40         CONTINUE
          IF (IFLAG .EQ. 1) GOTO 25
          NROUTE(K) = MACH
          CALL SETA(NEWJOB,K,RMACH)
          CALL SETA(NEWJOB,K+5,PROC)
          TOT_PROC = TOT_PROC + PROC
30       CONTINUE
*       RECORD TOTAL PROCESSING TIME OF JOB
        CALL SETA(NEWJOB,19,TOT_PROC)
        CALL SETA(NEWJOB,20,TOT_PROC)
*         PRINT *,'ARRIVE: A(19)=',TOT_PROC
*       RECORD TOTAL WORK ENTERING SHOP
        WEEK_WRK = WEEK_WRK + TOT_PROC
*       ASSIGN JOB DUE DATE: DD = k*NOP + TNOW
        DD = X(8)*NOP + TNOW
        CALL SETA(NEWJOB,13,DD)
*       JOB PLACED IN ORDER POOL
        CALL INSERT(NEWJOB,1)
10     CONTINUE
      CALL TALLY(1,WEEK_WRK)
*      PRINT *,'PREPARE TO CALL CRITICAL'
*     DETERMINE CRITICALNESS OF EACH WORK CENTER
```

```
100   CALL CRITICAL
*     PRIORITIZE JOBS IN ORDER POOL
      CALL ORDER
*     PRIORITIZE JOBS IN MACHINE QUEUES USING DISPATCH RULES
      CALL QRANK
*      WRITE(NO,*)'END OF SUBROUTINE ARRIVE'
*      PRINT *,'END OF SUBROUTINE ARRIVE'
      RETURN
      END

*-------------------------------------------------------------
      SUBROUTINE CRITICAL
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      REAL WORK
      LOGICAL SORTED
      IF (X(45) .LT. 3.0) THEN
*     RANK(1,1) IS MACHINE: RANK(1,2) IS WORKLOAD: MACH CONST
        RANK(1,1) = 1
        RANK(1,2) = 1
        RANK(2,1) = 2
        RANK(2,2) = 1
        RETURN
      ENDIF
      DO 5, I=1,6
        RANK(I,1) = 0.0
        RANK(I,2) = 0.0
        WC_CRIT(I) = 0.0
5     CONTINUE
*
      DO 10, I = 1,9
*       DETERMINE WORK IN POOL, MACH QUEUES & ON MACHINES
*       FILE 1 IS ORDER POOL: FILES 2 TO 7 ARE MACH QUEUES
*       FILE 8 IS SORTING FILE AND SHOULD BE EMPTY
*       FILE 9 IS JOBS IN PROCESS ON MACHINES
        IF (NQ(I) .EQ. 0) GOTO 22
        NEXT = LFR(I)
20       IF (NEXT .NE. LLR(I)) NEXTJOB = LSUCC(NEXT)
        JCUR_OPN = INT(A(NEXT,12))
        DO 30, L = INT(A(NEXT,12)), INT(A(NEXT,11))
*          DET WORK FOR CURR OPERATION AND FUTURE OPER
          MACH = INT(A(NEXT,L))
          WC_CRIT(MACH) = WC_CRIT(MACH) + A(NEXT,L+5)
30       CONTINUE
        IF (NEXT .EQ. LLR(I)) GOTO 22
        NEXT = NEXTJOB
        GOTO 20
22       NK = 1
10    CONTINUE
```

```
*     CALCULATE THE MEAN WORK LOAD
      TOT = 0.0
      DO 55, I = 1,6
        TOT = TOT + WC_CRIT(I)
55    CONTINUE
      AVGWL = TOT/(X(40))
*
*     RANK THE WORK CENTERS FOR CRITICALNESS
      DO 90 I = 1,6
        RANK(I,1) = REAL(I)
        RANK(I,2) = WC_CRIT(I)
90    CONTINUE
*     SORT RANK INTO DESCENDING ORDER
      SORTED = .FALSE.
100   IF (.NOT. SORTED) THEN
        SORTED = .TRUE.
        DO 200 I = 1,INT(X(40)-1)
          IF (RANK(I,2) .LT. RANK(I+1,2)) THEN
            TEMP_ONE = RANK(I,1)
            TEMP_TWO = RANK(I,2)
            RANK(I,1) = RANK(I+1,1)
            RANK(I,2) = RANK(I+1,2)
            RANK(I+1,1) = TEMP_ONE
            RANK(I+1,2) = TEMP_TWO
            SORTED = .FALSE.
          ENDIF
200     CONTINUE
        GOTO 100
      ENDIF
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE SHIFT(JSHIFT)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      *     UPDATES ORDER POOL PRIORITIES EVERY 8 HOURS
*      PRINT *,'BEGIN SUBROUTINE SHIFT: TIME NOW=',TNOW
      SHIFTEND = TNOW + 8.0
      SHIFTBEG = TNOW
      CALL SCHED(JSHIFT,2,8.0)
*     RECORD DATA GATHERED BY PERIOD
*     DO NOT COMPLETE IF START OF WEEK
      JW = MOD(TNOW,40)
      IF (JW .EQ. 0) RETURN
*     DETERMINE CRITICALNESS OF EACH WORK CENTER
      CALL CRITICAL
*     RANK ORDER POOL BY PRIORITY
      CALL ORDER
*      PRINT *,'RETURN FROM ORDER SUBROUTINE'
```

```
*     RANK MACHINE QUEUES
      CALL QRANK
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE TRUNC
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*     THIS SUB TRUNCATES NUMBER OF JOBS SPECIFIED BY X(5)
*      WRITE(NO,*) 'BEGIN SUBROUTINE TRUNC: X(5) = ',X(5)
      IF (X(5) .GT. 0.0) THEN
        X(5) = X(5) - 1
        CALL CLEAR
      ENDIF
      IF (X(5) .LT. 0.0) X(5) = 0.0
      RETURN
      END
```

```
*================
*     THIS GROUP IS CONCERNED WITH END OF SERVICE
*================
      SUBROUTINE DEPART(JOB)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      CALL RECINV
*     DISPOSE OF BOGUS ENTITY: OBTAIN CORRECT ENTITY
      CALL DISPOS(JOB)
      JOB = LFR(9)
      JCUR_OPN = INT(A(JOB,12))
      MACH = INT(A(JOB,JCUR_OPN))
      CALL REMOVE(JOB,9)
*     RELEASE LABOR AND MACHINE
      MCH_FREE = MACH
      X(MACH + 10) = X(MACH + 10) - 1
      X(29) = X(29) + 1
*     INCREASE AMOUNT OF WIP ON SHOP FLOOR
      X(39) = X(39) + A(JOB,JCUR_OPN+5)
*     DECREASE AMOUNT OF REMAINING TO BE DONE ON JOB
      REM_PROC = A(JOB,20) - A(JOB,JCUR_OPN+5)
      CALL SETA(JOB,20,REM_PROC)
*     CHECK IF JOB LEAVING MACHINE NEEDS MORE WORK
      IF (A(JOB,12) .EQ. A(JOB,11)) THEN
*        JOB FINISHED:  DECREASE CURRENT WIP ON SHOP FLOOR
         PROC = 0
         DO 30, I = 1, INT(A(JOB,11))
            PROC = PROC + A(JOB,I+5)
30       CONTINUE
         X(39) = X(39) - PROC
*        ALL JOBS GO TO FINISHED GOODS: SOME FOR ZERO TIME
         X(32) = X(32) + 1.0
         X(33) = X(33) + PROC
         DT = A(JOB,13) - TNOW
         IF (DT .LT. 0.0) DT = 0.0
         IF (X(45) .LT. 4.0) DT = 0.0
         CALL SCHED(JOB,4,DT)
      ELSE
*        TRANSFER JOB TO QUEUE FOR NEXT OPERATION
         CUR_OPN = A(JOB,12) + 1.0
         CALL SETA(JOB,12,CUR_OPN)
         CALL PRIOR(JOB)
      ENDIF
*     REASSIGN LABOR
      CALL WHEN(MCH_FREE)
      IF (X(29) .GT. 0.0) CALL LABOR
      CALL RECINV
*      PRINT *,'END OF DEPART SUBROUTINE'
      RETURN
      END
```

```
*-----------------------------------------------------
      SUBROUTINE EXITS(JOB)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
*     JOB IS READY TO EXIT FGS: RECORD INFORMATION
*     RECORD COST INFORMATION UNTIL NOW
      CALL RECINV
*     RECORD EXIT INFORMATION
      CALL RECEXT(JOB)
      PROC = 0.0
      DO 30, I = 1, INT(A(JOB,11))
        PROC = PROC + A(JOB, I + 5)
30    CONTINUE
*     REDUCE AMOUNT OF INVENTORY IN FGS INVENTORY
      X(32) = X(32) - 1.0
      X(33) = X(33) - PROC
      CALL RECINV
      CALL DISPOS(JOB)
*      PRINT *,'END OF EXITS: JOB DISPOSED =',JOB
      RETURN
      END

*-----------------------------------------------------
      SUBROUTINE LABOR
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
*     DETERMINES IF IDLE LABOR CAN BE REASSIGNED
*     LABOR IS AVAILABLE, BUT ARE ANY MACHINES FREE?
      I = 1
10    MACH = 7
      CALL WHERE(MACH)
      IFL = MACH + 1
      IF (MACH .LT. 7) CALL DISPATCH(IFL)
      IF ((X(29) .GT. 0.0) .AND. (MACH .LT. 7)) THEN
*       THERE MAY BE OTHER JOBS WAITING AT MACHINES
        I = I + 1
        IF ( I .EQ. 7) GOTO 20
        GOTO 10
      ENDIF
20    IF ((X(29) .GT. 0.0) .AND. (X(1) .GT. 2.0)) CALL SELECT
*     PRINT *,'END OF LABOR SUBROUTINE'
      RETURN
      END

*-----------------------------------------------------
      SUBROUTINE WHEN(MACH)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J   ,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      COMMON/ADAT/ALLDAT(100,8),LAB
*     THREE DIFFERENT "WHEN" RULES IN THE EXPERIMENT
*     1. QUE RULE
```

```
*     2. JOB RULE
*     3. CLR RULE
*     WHEN THE QUEUE IS EMPTY ALL THREE RULES BEHAVE THE SAME
*     MACH IS MACHINE THE LABORER IS BEING RELEASED FROM
*     LAB IS USED TO TRACK NUMBER OF LABOR TRANSFERS FOR SHOP
*      PRINT *,'START OF SUBROUTINE WHEN'
      IFL = MACH + 1
      IF (NQ(IFL) .EQ. 0) THEN
*       QUEUE EMPTY ALL RULES BEHAVE THE SAME
        TRANS = 1.0
        GOTO 100
      ENDIF
*     DETERMINE THE "WHEN" QUE RULE TO USE
      N = INT(X(2))
      GOTO (1,2,3),N
*
*     QUE RULE
*     RELEASE LABORER WHEN ALL WORK IN QUEUE IS COMPLETE
1     CALL DISPATCH(IFL)
*     THERE WAS NO LABOR TRANSFER
      TRANS = 0
      GOTO 100
      RETURN
*
*     JOB RULE:  RELEASE LABORER AFTER EVERY JOB
2     KOLD_MACH = MACH
      CALL WHERE(MACH)
      IF (KOLD_MACH .EQ. MACH) THEN
        TRANS = 0
*       LABOR REASSIGNED TO SAME MACHINE:  NO LABOR TRANSFER
      ELSE
        TRANS = 1
      ENDIF
      IFL = MACH + 1
      CALL DISPATCH(IFL)
      GOTO 100
      RETURN
*
*     CRITICAL LABOR RELEASE (CLR)
*     IF MACHINE IS CRITICAL USE QUE RULE
*     MACHINE NOT CRITICAL (i.e.LOAD< MEAN LOAD) USE JOB
*
3     IF (WC_CRIT(MACH) .GT. AVGWL) THEN
*       WC IS CRITICAL:STAY CURR MACH: NO LABOR TRANSFER
        GOTO 1
      ELSE
*       WORK CENTER NOT CRITICAL SO REASSIGN LABOR
        GOTO 2
      ENDIF
```

```
100   X(27) = X(27) + TRANS
      LAB = LAB + TRANS
      IF (TRANS .GT. 0.0) CALL COUNT(6,1)
      IF (X(29) .GT. 0.0) CALL LABOR
      RETURN
      END

*-----------------------------------------------------
      SUBROUTINE WHERE(MACH)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
*    CRITICAL WORK CENTER (CWC) "WHERE" LABOR ASSIGNMENT RULE
*    CWC ASSIGNS LABORERS TO MOST CRITICAL WC
*    IF 2 WC TIED, THE WC WAITING THE LONGEST RECEIVES LABOR
*      PRINT *,'START OF SUBROUTINE WHERE'
      IF (X(29) .LT. 1.0) RETURN
      DO 50 I = 1,6
        NEW_MACH = INT(RANK(I,1))
IF((X(NEW_MACH+10).EQ.2.0).OR.(NQ(NEW_MACH+1).LT.1)) GOTO 49
*       CHECK IF TWO WC ARE TIED FOR CRITICALITY
        IF (I .LT. 6) THEN
          NEXT_MCH = INT(RANK(I+1,1))
          IF(WC_CRIT(NEW_MACH) .EQ. WC_CRIT(NEXT_MCH)) THEN
            NUM_ONE = NQ(NEW_MACH+1)
            NUM_TWO = NQ(NEXT_MCH+1)
IF((NQ(NEXT_MCH+1).GT.0).AND.(X(NEXT_MCH+10).LT.2.0))THEN
       IF (NUM_ONE .LT. NUM_TWO) NEW_MACH = NEXT_MCH
            ENDIF
          ENDIF
        ENDIF
*       ASSIGN LABOR TO WORK CENTER
        MACH = NEW_MACH
        RETURN
49       JK = 1
50    CONTINUE
      RETURN
      END
```

```
*==================
*     THIS GROUP IS CONCERNED WITH DISPATCHING
*==================
      SUBROUTINE DISPATCH(IFL)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
*     IDENTIFY DISPATCHING RULE IN EFFECT
*     1. FCFS:  2. MODD:  3. CMDD
      IF (NQ(IFL) .EQ. 0) RETURN
      N = INT(X(3))
      GOTO (1,2,3),N
*     FIRST COME FIRST SERVED (FCFS): QUEUE ALREADY ORGANIZED
1     GOTO 300
*
*     MODIFIED OPERATION DUE DATE (MODD): DYNAMIC
*     OPERATION FLOW TIME=[JOB DD-ARRIVAL TIME]/NUMBER OPER
*     OPNDD = (OPN NUMBER)*FLOWTIME + TIME ARRIVAL
*     MODD = MINIMUM[MAX(SOT+TNOW, OPNDD)]
2     NEXT = LFR(IFL)
      IF (NEXT .EQ. 0) GOTO 200
      CALL REMOVE(NEXT,IFL)
      SOT = SPT(NEXT)
      OPN_DD = OPERD(NEXT)
      SOT = SOT + TNOW
      IF (SOT .GE. OPN_DD) THEN
        CALL SETA(NEXT,15,SOT)
      ELSE
        CALL SETA(NEXT,15,OPN_DD)
      ENDIF
      IF (NEXT .NE. LLR(IFL)) THEN
        CALL INSERT(NEXT,8)
        GOTO 2
      ELSE
        CALL INSERT(NEXT,8)
      ENDIF
*     PUT JOBS BACK INTO THEIR OWN FILE
200   NEXT = LFR(8)
      IF (NEXT .EQ. 0) GOTO 300
      CALL REMOVE(NEXT,8)
      IF (NEXT .NE. LLR(IFL)) THEN
        CALL INSERT(NEXT,IFL)
        GOTO 200
      ELSE
        CALL INSERT(NEXT,IFL)
      ENDIF
      GOTO 300
*
*     CRITICAL MACHINE DUE DATE (CMDD): STATIC FOR ONE SHIFT
*     END OF SHIFT, SUBROUTINE ORDER CHANGES QUEUE PRIORITY
```

```
*     USE SOT FOR CRITICAL MACHINE; USE OPNDD OTHERWISE
*     CRITICAL MACHINE HAS LOAD GREATER THAN AVERAGE LOAD
*     QUEUE ALREADY ORGANIZED
3     GOTO 300
*
*
*     REMOVE TOP PRIORITY JOB FROM QUEUE AND PROCESS IT
300    NEXT = LFR(IFL)
      CALL REMOVE(NEXT,IFL)
*     CREATE NEW ENTITY TO SCHEDULE JOB PROCESSING
      JCUR_OPN = INT(A(NEXT,12))
      DT = A(NEXT,JCUR_OPN + 5)
      SCHED_DT = DT + TNOW
      CALL SETA(NEXT,18,SCHED_DT)
      CALL INSERT(NEXT,9)
      CALL CREATE(JOB)
      WC = REAL(IFL - 1)
      CALL SETA(JOB,1,WC)
      CALL SCHED(JOB,3,DT)
*     SET MACHINE TO BUSY
      MACH = INT(IFL-1)
      X(MACH + 10) = X(MACH + 10) + 1
*     SET LABOR TO BUSY
      X(29) = X(29) - 1.0
      CALL RECINV
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE PRIOR(JOB)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J
,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
*     SET APPROPRIATE PRIORITY FOR JOB IN QUEUE
*     DETERMINE WHICH QUEUE JOB IS PLACED INTO
*      PRINT *,'START SUBROUTINE PRIOR: JOB LOCATION =',JOB
      JCUR_OPN = INT(A(JOB,12))
      MACH = INT(A(JOB,JCUR_OPN))
      IFL = MACH + 1
*     DETERMINE CORRECT JOB DISPATCH RULE TO USE
*     1 = FCFS:  2 = MODD:  3 = CMDD
      N = INT(X(3))
      GOTO (1,2,3),N
*     FCFS
1     CALL SETA(JOB,15,TNOW)
      GOTO 100
*     MODD = MIN[MAX(SOT+TNOW, OPNDD)]
*     DYNAMIC RULE WHERE CALCULATIONS ARE REDONE WHENEVER
*     A JOB IS SELECTED FROM THE QUEUE
```

```
2     CALL SETA(JOB,15,TNOW)
      GOTO 100
*     CMDD = CRITICAL MACH USE SOT AND NON CRITICAL MACH
*     USE OPN_DD.  STATIC FOR AN ENTIRE SHIFT.
3     IF (WC_CRIT(MACH) .GT. AVGWL) THEN
*       WC IS CRITICAL
        SOT = SPT(JOB)
        CALL SETA(JOB,15,SOT)
      ELSE
*       WC NOT CRITICAL
        OPN_DD = OPERD(JOB)
        CALL SETA(JOB,15,OPN_DD)
      ENDIF
100   CALL INSERT(JOB,IFL)
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE QRANK
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
*     RANK MACHINE QUEUES AT BEGINNING OF EACH SHIFT UNLESS
*     IT IS A DYNAMIC QUEUE
*      PRINT *,'START OF QRANK: DISPATCH RULE =',X(3)
      N = INT(X(3))
      GOTO (1,2,3),N
*     FCFS: NOT DYNAMIC: QUEUE ALREADY ORGANIZED
1     RETURN
*     MODD IS DYNAMIC: PRIORITY DETERMINED EVERY TIME A JOB
*     IS RELEASED
2     RETURN
*     CMDD IS STATIC FOR ENTIRE SHIFT
3     DO 100 I = 2,7
        IF (NQ(I) .EQ. 0) GOTO 99
        MACH = I - 1
        IF (WC_CRIT(MACH) .GT. AVGWL) THEN
*         WORK CENTER IS CRITICAL
          DO 150, K = 1,NQ(I)
            NEXT = LFR(I)
            CALL REMOVE(NEXT,I)
            SOT = SPT(NEXT)
            CALL SETA(NEXT,15,SOT)
            CALL INSERT(NEXT,8)
150       CONTINUE
        ELSE
*         WORK CENTER NOT CRITICAL
          DO 170, K = 1,NQ(I)
            NEXT = LFR(I)
            CALL REMOVE(NEXT,I)
```

```
            OPN_DD = OPERD(NEXT)
            CALL SETA(NEXT,15,OPN_DD)
            CALL INSERT(NEXT,8)
170         CONTINUE
          ENDIF
99         DO 190 K = 1, NQ(8)
            NEXT = LFR(8)
            CALL REMOVE(NEXT,8)
            CALL INSERT(NEXT,I)
190        CONTINUE
100      CONTINUE
       RETURN
       END
```

```
*==================
*     THIS GROUP IS CONCERNED WITH ORDER RELEASE
*==================
      SUBROUTINE CMS
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      REAL Q(6),CONG
      DATA Q/6*0.0/,CONG/0.0/,PROC/0.0/
*     RULE PRIORITIZES POOL USING POOL, SHOP AND LABOR INFO
*     CODE ADAPTED RAGATZ (1985): RD = f(PROC,CONGESTION)
*     RD SET AS IN MIL: JOBS NOT RELEASED UNTIL LABOR IS FREE
*     DETERMINE NUMBER OF JOBS IN EACH QUE & ON EACH MACH
*      IF (NQ(1) .EQ. 0) RETURN
      DO 10 I = 1,6
*        CALC NUMBER OF JOBS IN QUEUES/SERVICE AT EACH MACH
         Q(I) = NQ(I+1) + X(10+I)
10     CONTINUE
11     NEXT = LFR(1)
      IF (NEXT .EQ. 0) GOTO 25
      CONG = 0.0
      CALL REMOVE(NEXT,1)
      PROC = A(NEXT,19)
      NOP = INT(A(NEXT,11))
      DO 20 K = 1,NOP
        MACH = INT(A(NEXT,K))
        CONG = CONG + Q(MACH+1)
20     CONTINUE
      DD = A(NEXT,13)
      RD = DD - X(6)*PROC - X(7)*CONG
      CALL SETA(NEXT,14,RD)
      CALL INSERT(NEXT,8)
      IF (NEXT .NE. LLR(1)) GOTO 11
*     PUT JOBS BACK INTO ORDER POOL
25     NUM = NQ(8)
      DO 30 I = 1, NUM
        NEXT = LFR(8)
        IF (NEXT .EQ. 0) GOTO 100
        CALL REMOVE(NEXT,8)
        CALL INSERT(NEXT,1)
30     CONTINUE
100    IF (X(29) .GT. 0.0) CALL LABOR
*      PRINT *,'END OF SUBROUTINE CMS'
      RETURN
      END

*-------------------------------------------------
      SUBROUTINE IMM
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
*     IMMEDIATE RELEASE OF JOBS TO FLOOR
```

```
*      PRINT *,'START OF IMM: TOT JOBS IN=',X(4),'JOBS OUT='
      DO 10 I = 1,NQ(1)
        NEXT = LFR(1)
        CALL REMOVE(NEXT,1)
        CALL SETA(NEXT,14,TNOW)
        CALL SETA(NEXT,17,TNOW)
*       INSERT JOBS INTO MACH QUE, SET DISPATCH PRIORITY
        CALL PRIOR(NEXT)
*       RECORD NUMBER OF RELEASES FROM POOL: X(24)
        X(24) = X(24) + 1.0
10    CONTINUE
*     START PROCESSING JOBS
      IF (X(29) .GT. 0.0) CALL LABOR
*      NUM = NQ(2)+NQ(3)+NQ(4)+NQ(5)+NQ(6)+NQ(7)+NQ(9)
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE MIL
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      REAL Q(6,2),CONG
      DATA Q/12*0.0/,CONG/0.0/,PROC/0.0/
*     RULE PRIORITIZES POOL USING POOL AND SHOP INFORMATION
*     CODE FROM RAGATZ (1985): RD=f(NOP,CONGESTION)
*     DETERMINE # OF JOBS IN EACH QUEUE AND ON EACH MACHINE
      IF (NQ(1) .EQ. 0) RETURN
      DO 10 I = 1,6
*       CALC # OF JOBS IN QUEUES/SERVICE AT EACH MACHINE
        Q(I,1) = NQ(I+1) + X(10+I)
*       CALC PROC TIME OF JOBS IN QUEUES AT EACH MACHINE
*        IF ((NQ(I+1) .EQ. 0) .OR. (X(45) .EQ. 6)) GOTO 9
*        NEXT = LFR(I+1)
*        DO 11 K = 1,NQ(I+1)
*          IF (NEXT .NE. LLR(I+1)) JSUCC = LSUCC(NEXT)
*          Q(I,2) = Q(I,2) + A(NEXT,20)
*          NEXT = JSUCC
*11       CONTINUE
*9        NEXT =1
10    CONTINUE
*     CALCULATE RELEASE DATE
12    NEXT = LFR(1)
      IF (NEXT .EQ. 0) GOTO 25
      CONG = 0.0
      CALL REMOVE(NEXT,1)
      DO 20 K = 1,INT(A(NEXT,11))
        MACH = INT(A(NEXT,K))
*       IF (X(45) .EQ. 4) THEN
*         CONG DEPENDS ON # OF JOBS AT MACHINES (RAGATZ)
```

```
*          CONG = CONG + Q(MACH,1)
*        ELSEIF (X(45) .EQ. 5) THEN
*          CONG IS PROC TIME OF JOBS AT MACH (PARK)
*          CONG = CONG + Q(MACH,2)
*        ELSEIF (X(45) .EQ. 6) THEN
*          CONGESTION DEPENDS ON NUMBER OF JOBS AT MACHINES
          CONG = CONG + Q(MACH,1)
*        ENDIF
20     CONTINUE
       DD = A(NEXT,13)
       IF (X(45) .EQ. 6) THEN
         RD = DD - X(6)*A(NEXT,19) - X(7)*CONG
       ELSEIF (X(45) .EQ. 5) THEN
         RD = DD - X(6)*A(NEXT,19) - X(7)*CONG
       ENDIF
       CALL SETA(NEXT,14,RD)
       IF (NEXT .EQ. LLR(1)) THEN
         CALL INSERT(NEXT,8)
         GOTO 25
       ELSE
         CALL INSERT(NEXT,8)
         GOTO 12
       ENDIF
*      PUT JOBS BACK INTO ORDER POOL
25     NEXT = LFR(8)
       IF (NEXT .EQ. 0) GOTO 40
       CALL REMOVE(NEXT,8)
       IF (A(NEXT,14) .GT. SHIFTEND) THEN
         CALL INSERT(NEXT,1)
         GOTO 25
       ELSE
         DT = A(NEXT,14) - TNOW
         IF (DT .LT. 0.0) DT = 0.0
*        JOBS ARE RELEASED THIS SHIFT AS EVENTS
         CALL SCHED(NEXT,5,DT)
         GOTO 25
       ENDIF
40     IF (X(29) .GT. 0.0) CALL LABOR
*      PRINT *,'END OF MIL: RETURNED FROM LABOR'
       RETURN
       END

*-------------------------------------------------------
       SUBROUTINE MLC
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
       REAL DF(6), DLOAD(6), CF(6), WC(6,5)
*      PRINT *,'START OF MLC'
       DO 5, I = 1,6
```

```
        DF(I) = 0.0
        DLOAD(I) = 0.0
        CF(I) = 0.0
        WC(I,1)= 0.0
        WC(I,2)= 0.0
        WC(I,3)= 0.0
        WC(I,4)= 0.0
        WC(I,5)= 0.0
5     CONTINUE
*     CALC DISCOUNT FACTOR FOR EACH STAGE GIVEN X(10)
      DO 50, I = 1,5
        DF(I) = (X(10))**(I-1)
50    CONTINUE
*
      DO 10, I = 1,9
*       DETERMINE WORK IN POOL, MACH QUEUES & ON MACHINES
*       FILE 1 IS ORDER POOL: FILES 2 - 7 ARE MACHINE QUEUES
*       FILE 8 IS EMPTY: FILE 9 CONTAINS JOBS IN PROCESS
        NEXT = LFR(I)
        DO 20, M = 1,NQ(I)
          IF (NEXT .NE. LLR(I)) ISUCC = LSUCC(NEXT)
          JCUR_OPN = INT(A(NEXT,12))
          K = 1
          DO 30, L = JCUR_OPN,INT(A(NEXT,11))
            MACH = INT(A(NEXT,L))
            WC(MACH,K) = WC(MACH,K) + A(NEXT,L+5)
            K = K + 1
30        CONTINUE
          NEXT = ISUCC
20      CONTINUE
10    CONTINUE
*     CALCULATE DISCOUNTED LOAD FOR EACH MACHINE
      DO 175 I = 1,6
        DO 185 K = 1,5
          DLOAD(I) = DLOAD(I) + WC(I,K)*DF(K)
185     CONTINUE
175   CONTINUE
*
*     CALC DELAY BECAUSE OF CONGESTION USING DISCOUNTED LOAD
*      LP = 40 - MOD(SHIFTBEG,40)
*      DO 200, I = 1,6
*        CF(I) = DLOAD(I)/LP
*200   CONTINUE
*
*     CALCULATE RELEASE DATE
      NUM = NQ(1)
      DO 315 I = 1, NUM
        CONG = 0.0
        NEXT = LFR(1)
```

```
        CALL REMOVE(NEXT,1)
        MACH = INT(A(NEXT,1))
        CONG = DLOAD(MACH)
        DO 320 K = 2,INT(A(NEXT,11))
          MACH = INT(A(NEXT,K))
          CONG = CONG + DLOAD(MACH)/X(9)
320       CONTINUE
        RD = A(NEXT,13) - CONG - A(NEXT,19)
        CALL SETA(NEXT,14,RD)
        CALL INSERT(NEXT,8)
315     CONTINUE
*
*       PUT JOBS BACK INTO ORDER POOL
        DO 430 I = 1, NUM
          NEXT = LFR(8)
          CALL REMOVE(NEXT,8)
          CALL INSERT(NEXT,1)
430     CONTINUE
*       IF LABOR IS IDLE, SELECT A JOB
        IF (X(29) .GT. 0.0) CALL LABOR
        RETURN
        END

*-------------------------------------------------------
        SUBROUTINE ORDER
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
*       RANKS JOBS IN ORDER POOL USING CURRENT RELEASE RULE
*       THIS EXPERIMENT USES FOUR DIFFERENT PRIORITIES
*       1. IMMEDIATE RELEASE (IMM)
*       2. MIL
*       3. CMS
*       4. MLC
*       CHECK IF THERE ARE ANY JOBS IN POOL TO ORDER
*        PRINT *,'BEGIN SUBROUTINE ORDER: TNOW =',TNOW
        IF (NQ(1) .EQ. 0) RETURN
*       DETERMINE WHICH RULE IS IN USE
        N = INT(X(1))
        GOTO (1,2,3,4), N
*
*       IMMEDIATE RELEASE OF JOBS TO SHOP FLOOR
1       CALL IMM
        RETURN
*
*       MODIFIED INFINITE LOADING (MIL)
*       MIL RELEASES JOBS CONTINUOUSLY AT ASSIGNED RD
*       WHEN RD IS DURING A SHIFT IT IS SCHEDULED AS AN EVENT
2       CALL MIL
*        PRINT *,'SUBROUTINE ORDER, RETURN FROM MIL'
        RETURN
```

```
*
*     CRITICAL MACHINE SELECTION (CMS)
*     RD IS SET WITH MIL LOGIC.  JOBS ARE NOT RELEASED
*     UNTIL LABOR IS IDLE.
3     CALL CMS
      RETURN
*
*     MODIFIED LOAD CONVERSION (MLC)
*     JOBS ARE NOT RELEASED UNTIL LABOR IS IDLE.
4     CALL MLC
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE RELEASE(JOB)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
*     EVENT # 5 IN MIL:  RELEASES JOB FROM POOL IN MIL
*     PUTS JOBS INTO QUEUE OF FIRST MACHINE IN ITS ROUTE
*      PRINT *,'BEGIN OF SUBROUTINE RELEASE'
      CALL SETA(JOB,17,TNOW)
      CALL PRIOR(JOB)
*     RECORD NUMBER OF RELEASES FROM ORDER POOL: X(24)
      X(24) = X(24) + 1.0
*     IF LABOR IS IDLE SCHEDULE LABORER
      IF (X(29) .GT. 0.0) CALL LABOR
**      PRINT *,'END OF SUBROUTINE RELEASE'
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE SELECT
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
*      RNUM = X(11)+X(12)+X(13)+X(14)+X(15)+X(16)
      M = 1
*     SUBROUTINE CALLED FOR MLC AND CMS JOB SELECTION MECH
*     LABOR IDLE & THERE ARE NO JOBS IN THE SHOP TO WORK ON.
10     DO 120 K = 1,6
*       IDENTIFY MOST CRITICAL MACH & FIND ELG JOB FOR MACH
        MACH_CR = INT(RANK(K,1))
        IF (X(MACH_CR+10) .EQ. 2.0) GOTO 290
        NEXT = LFR(1)
21       IF (NEXT .EQ. 0) GOTO 290
        IF (NEXT .NE. LLR(1)) NEXT_JOB = LSUCC(NEXT)
        IF (A(NEXT,14) .GT. SHIFTEND) GOTO 290
*       IF HERE, JOB HAS ELIGIBLE RELEASE DATE
        MACH = INT(A(NEXT,1))
*        PRINT *,'FIRST MACHINE IN ROUTE=',MACH
        IF (MACH .EQ. MACH_CR) GOTO 170
```

```
      NEXT = NEXT_JOB
      IF (NEXT .NE. LLR(1)) GOTO 21
290     NE = 1
*      EITHER CRIT MACH NOT FREE, OR NO JOBS IN POOL
120   CONTINUE
*     PRINT *,'END SELECT: NO JOBS TO ASSIGN: # LOOPS =',K-1
      RETURN
170   CALL REMOVE(NEXT,1)
      IFL = INT(A(NEXT,1)) + 1
      CALL SETA(NEXT,17,TNOW)
      CALL PRIOR(NEXT)
      PROC = 0.0
      DO 180 I = 1,5
*       PRINT*,'OPERATION =',I,' PROCESSING=',A(NEXT,I+5)
        PROC = PROC + A(NEXT,I+5)
180   CONTINUE
*     PRINT*,'TOTAL PROCESSING =',PROC
      CALL DISPATCH(IFL)
*     RECORD RELEASES FROM ORDER POOL
      X(24) = X(24) + 1.0
      IF (X(29) .GT. 0.0) GOTO 300
*     PRINT*,'END OF SELECT'
      RETURN
300   M = M + 1
      IF (M .LT. 7) GOTO 10
*     PRINT *,' END OF SELECT'
      RETURN
      END
```

```
*=================
*     THIS GROUP IS CONCERNED WITH RECORD KEEPING
*=================
      SUBROUTINE PERIOD(JOB)
COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      COMMON/QUART/PER(400,2),NTH,TOT_COST
      REAL CONST(200,3)
*     THIS SUBROUTINE COLLECTS DATA FOR SPECIFIED PERIOD
*      WRITE(NO,*) 'SUBROUTINE PERIOD: X(36) =',X(36)
      IF (X(45) .LT. 4.0) RETURN
      CALL SCHED(JOB,7,40.0)
      CALL TALLY(9,X(28))
      CALL TALLY(10,X(27))
      IF (X(5) .GT. 0.0) THEN
*        TRUNCATE THE NUMBER OF WEEKS = X(5) IN SUB  BEGN
         CALL TRUNC
         X(48) = 1
         GOTO 400
      ENDIF
      IF (X(35) .LE. X(36)) GOTO 200
      X(36) = X(36) + 1
      X(28) = 0
      X(27) = 0
      RETURN
*
200    COMB_INV = DAVG(2) + DAVG(3)
      TARDY = TAVG(9)
      TRANSFER = TAVG(10)
      WRITE(NO,*) 'TRANSFER, DAVG(2), DAVG(3), COMB_INV'
201    FORMAT(1X,4(2X,F8.3))
      WRITE(NO,201) TRANSFER, DAVG(2), DAVG(3), COMB_INV
100    FORMAT(1X,4(1X,F10.3))
*     X(42) IS COST OF TARDY WORK: X(43) IS COST OF INVENTORY
*     X(44) IS COST OF LABOR TRANSFERS
TOT_COST = X(42)*TARDY + X(43)*COMB_INV + X(44)*TRANSFER
      NM = INT(X(48))
IF ((X(49) .EQ. 1) .OR.(X(49) .EQ. 3)) TOT_COST = -1*TOT_COST
      SAMP(NM) = TOT_COST
      WRITE(NO,*) 'TARDY, COMB_INV, TRANSFER, COST'
      WRITE(NO,100) TARDY, COMB_INV, TRANSFER, TOT_COST
      IT = INT(X(1))
      IU = INT(X(2))
      IV = INT(X(3))
      IW = INT(X(50))
      IF(X(49).EQ.10) THEN
         TC = TOT_COST
         TRAN = TRANSFER
```

```
      COM = COMB_INV
      TAR = TARDY
      INJ = NRUN + INT(X(47))
      WRITE(NI,101)IT,IU,IV,IW,INJ,IT,IU,IV,IW,TAR,COM,TRAN,TC
101   FORMAT(1X,4(I1),2X,I2,2X,4(I1,1X),4(1X,F10.3))
      WRITE(NO,*) 'END OF SUB PERIOD'
      X(36) = 1
      CALL COUNT(2,1)
      RETURN
    ENDIF
300   IF ((X(48) .GE. X(41)).AND. (X(49) .LT. 10)) THEN
*      ENOUGH SAMPLES COLLECTED FOR ANALYSIS
      AVG1 = 0
      ADEV1 = 0
      SDEV1 = 0
      VAR1 = 0
      SKEW1 = 0
      CURT1 = 0
      CALL MOMENT(AVG1,ADEV1,SDEV1,VAR1,SKEW1,CURT1)
      T_TEST = 0.0
      DF = X(48)
      CALL TEST(T_TEST,DF)
      AB = REAL(NM)
      AC = X(6)
      AD = X(7)
      AE = X(9)
      AF = X(10)
      IF (X(49) .LE. 10) CL = 0.05
     IF((X(49).EQ.1.0).OR.(X(49).EQ.3))
            CALL FUNC(AVG1,SDEV1,T_TEST)
      IF ((X(48) .EQ. 1.0) .OR. (X(48) .GE. X(41))) THEN
313       FORMAT(1X,4(2X,F13.4))
314       FORMAT(1X,3(2X,F13.4))
        WRITE(NO,*)'LAST POINT WAS:'
WRITE(NO,*)'X6, X7, X9, X10, AVG1, C.L.,DF, SDEV, T,SIZE'
       WRITE(NO,316)AC,AD,AE,AF,AVG1,CL,AB,SDEV1,T_TEST,X(35)
        WRITE(NO,*) 'NOW FIND NEW POINT'
        WRITE(NO,313) X(1),X(2),X(3),X(50)
        WRITE(NO,314) X(35),CL,X(48)
        WRITE(NO,*) 'THE NEW POINT IS: X6, X7, X9, X10'
        WRITE(NO,313) X(6),X(7),X(9),X(10)
316   FORMAT(1X,4(1X,F7.2),1X,F13.4,2(1X,F6.2),3(1X,F8.3))
WRITE(NI,*)'X6,  X7,X9,X10,  AVG1, C.L., DF, SDEV, T, SIZE'
  WRITE(NI,316)AC,AD,AE,AF,AVG1,CL,AB,SDEV1,T_TEST,X(35)
        WRITE(NI,*)' '
        WRITE(NI,*) ' '
        SD = SDEV1
        TT = T_TEST
317   FORMAT(1X,4(I1),1X,4(1X,F7.2),1X,F13.4,
```

```
        2(1X,F6.2),3(1X,F8.3))
    WRITE(NJ,317)IT,IU,IV,IW,AC,AD,AE,AF,AVG1,CL,AB,SD,TT,X(35)
  WRITE(NO,*)'TOT_COST = 2*TARDY + .1*COMB_INV + .1*TRANSFER'
          IF (X(49) .EQ. 0) CALL COUNT(2,1)
        ENDIF
      ELSE
        X(48) = X(48) + 1.0
      ENDIF
400   X(28) = 0
      X(27) = 0
      X(36) = 1
      CALL CLEAR
      CALL RECINV
*      PRINT *,'END SUBROUTINE PERIOD: X(48)=',X(48)
      RETURN
      END

*----------------
      SUBROUTINE  RECEXT(JOB)
 COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      COMMON/QUART/PER(400,2),NTH,TOT_COST
*     RECORDS COST INFORMATION TO FILE WHEN JOB EXITS SYSTEM
*     RECORD THE NUMBER OF JOBS COMPLETED
      X(37) = X(37) + 1.0
      CALL COUNT(3,1)
      CALL COUNT(1,1)
      TME_TARD = 0
*     COUNTER DOES NOT CONTROL RUNS: IT TRACKS COMPLETIONS
      IF (A(JOB,13) .LT. A(JOB,18)) THEN
*       A(JOB,18) IS THE TIME JOB FINISHED LAST OPERATION
        DIFF = A(JOB,18) - A(JOB,13)
        TME_TARD = DIFF
        CALL COUNT(4,1)
        CALL TALLY(7,TME_TARD)
        IF ( DIFF .LT. 40.0) THEN
          JPER = 1
        ELSE
          IF (DIFF .LT. 80.0) THEN
            JPER = 2
          ELSE
            IF (DIFF .LT. 120.0) THEN
              JPER = 3
            ELSE
              JPER = 4
            ENDIF
          ENDIF
        ENDIF
```

```
*    JOB TARDY: MULTIPLY NUMBER HOURS OF WORK LATE X PERIODS
       X(28) = X(28) + JPER*A(JOB,19)
       X(38) = X(38) + 1.0
     ENDIF
*   RECORD FLOW TIME AND TIME ON FLOOR
     QUE_TME = A(JOB,18) - A(JOB,17) - A(JOB,19)
     FLOW = A(JOB,18) - A(JOB,17)
     IF (QUE_TME .LT. 0.0001) QUE_TME = 0.0
     POOLTME = A(JOB,17) - A(JOB,16)
     CALL TALLY(2,QUE_TME)
     CALL TALLY(3,FLOW)
     CALL TALLY(4,POOLTME)
     IF (X(49) .EQ. 10) THEN
       N = NTH
 IF ((TME_TARD .GT. 0) .AND. (TME_TARD .GT. PER(N,1))) THEN
         CALL PERCNT(TME_TARD,FLOW)
  ELSEIF ((FLOW .GT. 0) .AND. (FLOW .GT. PER(N,2))) THEN
         CALL PERCNT(TME_TARD,FLOW)
       ENDIF
     ENDIF
     IF (X(45) .GT. 3.0) RETURN
     CALL COUNT(1,1)
101   FORMAT(1X,F13.2)
     IF (NRUN .EQ. 1) WRITE(11,101) QUE_TME
     IF (NRUN .EQ. 2) WRITE(12,101) QUE_TME
     IF (NRUN .EQ. 3) WRITE(13,101) QUE_TME
     IF (NRUN .EQ. 4) WRITE(14,101) QUE_TME
     IF (NRUN .EQ. 5) WRITE(15,101) FLOW
     IF (NRUN .EQ. 6) WRITE(16,101) FLOW
     IF (NRUN .EQ. 7) WRITE(17,101) FLOW
     IF (NRUN .EQ. 8) WRITE(18,101) FLOW
     IF (NRUN .EQ. 9) WRITE(19,101) FLOW
*     RESET VARIABLES TO ZERO
910   X(25) = TNOW
     X(26) = 0.0
     PRINT *,'END OF SUBROUTINE RECEXT'
     RETURN
     END

*-----------------------------------------------------
     SUBROUTINE RECINV COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,-
TNOW,TFIN,J,NRUN
*     STORE COST INFORMATION UNTIL JOB EXITS SYSTEM
*      PRINT *,'START OF SUBROUTINE RECINV'
     TME = TNOW - X(25)
     X(25) = TNOW
*     NUMBER JOBS IN QUEUES
     X(21) = X(21) + X(22)*TME
     IF (X(49) .EQ. 10) THEN
```

```
        X(17) = X(11) + X(12) + X(13) + X(14) + X(15) + X(16)
        X(18) = 6 - X(29)
        RETURN
      ENDIF
      HR_WIP = X(39)*TME
      HR_FGS = X(33)*TME
      X(26) = X(26) + HR_WIP + HR_FGS
*     MACHINE UTILIZATION
      X(19) = X(19) + X(17)*TME
*     LABOR UTILIZATION
      X(20) = X(20) + X(18)*TME
      CNT = 0.0
      X(17) = 0.0
      DO 100 I = 1,INT(X(40))
*       COUNT THE NUMBER OF BUSY MACHINES
        IF (X(45) .EQ. 6.0) THEN
          X(17) = X(17) + X(I+10)
          CNT = CNT + 2.0
        ELSE
          X(17) = X(17) + X(I+10) - 1.0
          CNT = CNT + 1.0
        ENDIF
100   CONTINUE
      X(17) = X(17)/CNT
*     COUNT NUMBER OF JOBS IN QUEUES
 X(22) = NQ(2) + NQ(3) + NQ(4) + NQ(5) + NQ(6) + NQ(7) + NQ(9)
*       PRINT *,'END OF RECINV SUBROUTINE'
      RETURN
      END

*-------------------------------------------------------
      SUBROUTINE SHOP(JOB) COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTN-
OW,TNOW,TFIN,J,NRUN COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,W-
C_CRIT(6),REL
      COMMON/ADAT/ALLDAT(100,8),LAB
*     COLLECTS INFO. TO ILLUSTRATE EFFECT OF RULES
*     ARRIVALS, JOBS IN POOL, RELEASES, COMPLETIONS (X(37),
 INV
IF (TNOW .LT. 100.0) CALL SCHED(JOB,6,REL)
      JB = NQ(2) + NQ(3) + NQ(4) + NQ(5) + NQ(6) +NQ(7)+NQ(9)
        H = X(39) + X(33)
        ID = INT(X(29))
        LA = INT(X(38))
*  ID = IDLE LABOR: LA= LATE JOBS: LAB=LABOR TRANS: H=INVENTOR
25       FORMAT(1X,2(1X,F7.0),1X,I6,3(1X,F7.0),4(1X,I4),1X,F7.2) WRITE(10,25)TNO-
W,X(4),NQ(1),X(24),X(37),X(32),JB,LA,LAB,ID,H
        X(4) = 0.0
        X(24) = 0.0
        X(37) = 0.0
```

```
        X(38) = 0.0
        LAB = 0.0
      RETURN
      END

*----------------------------
      SUBROUTINE WRAPUP COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),-
DTNOW,TNOW,TFIN,J,NRUN
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      COMMON/QUART/PER(400,2),NTH,TOT_COST
*       PRINT *,'START OF WRAPUP'
*       WRITE(NO,*) 'START OF WRAP: X(49) = ',X(49)
*      END OF SIMULATION RUN
*       X(28) = -1
*       X(26) = -1
*       X(27) = -1
        TA = TAVG(7)
*        TALLY 7 IS HOURS TARDY FOR JOBS TARDY: SO IT IS AVG/TARDY JOB
        TRANSFER = TAVG(10)
        FGS_INV = DAVG(3)
        WIP_INV = DAVG(2)
        WRITE(NO,*) 'TRANSFER, FGS, WIP'
11       FORMAT(1X,3(2X,F8.3))
        WRITE(NO,11) TRANSFER, FGS_INV, WIP_INV
        QUE_POOL = TAVG(4)
        QUE_SHOP = TAVG(2)
        FLOW = TAVG(3)
        LATE = NC(4)
*        DETERMINE HOW MANY JOBS WERE LATE: 90TH PERC *
TARDINESS
        JK = INT(LATE*.1)
        TARD9 = PER(JK,1)
        FLOW9 = PER(NTH,2)
62       FORMAT(1X,5(1X,F8.3))
        WRITE(NO,*) 'TARDY/JOB,                    FGS_INV,WIP_INV,QUE_POOL,QUE-
_SHOP'
        WRITE(NO,62) TARDY,FGS_INV,WIP_INV,QUE_POOL,QUE_SHOP
        WRITE(NO,*) 'FLOW,FLOW9,TARD9'
        WRITE(NO,62) FLOW,FLOW9,TARD9
        WP = WIP_INV
        FG = FGS_INV
        QP = QUE_POOL
        QS = QUE_SHOP
        F = FLOW
        F9 = FLOW9
        T9 = TARD9
        A = TA
        W = WIP_INV
        G = FGS_INV
```

```
      F = FLOW
      I = INT(X(1))
      J = INT(X(2))
      K = INT(X(3))
      L = INT(X(50))
      N = NRUN + INT(X(47))
    IF (X(49) .EQ. 10) THEN
      WRITE(NJ,63)I,J,K,L,N,I,J,K,L,TA,FG,WP,QP,QS,F,F9,T9
63     FORMAT(1X,4(I1),2X,I2,2X,4(I1,1X),8(1X,F8.3))
    ENDIF
      COMP = NC(1)
      IF (COMP .LE. 0) GOTO 64
      P = LATE/COMP
64     UM = DAVG(4)/12
      UL = DAVG(5)/6
      AR = X(4)
      WRITE(NO,*)'MACHINE UTIL, LABOR UTIL,                    PCT_TARDY,-
COMPL,ARRIVE'
      WRITE(NL,68)I,J,K,L,N,I,J,K,L,UM,UL,P,COMP
      WRITE(NO,65) IT,IU,IV,IW,N,UM, UL,P,COMP,AR
65     FORMAT(1X,4(I1),2X,I2,2X,5(1X,F10.3))
68     FORMAT(1X,4(I1),2X,I2,2X,4(I1,1X),1X,5(1X,F10.3))
      R =TAVG(10)
      C = DAVG(2)+DAVG(3)
      T = TAVG(9)
      TC = TOT_COST WRITE(KJ,1)I,J,K,L,N,I,J,K,L,T,C,R,TC,A,QP,QS,F,F9,T9,W,G,U-
M,UL,P
1      FORMAT(1X,4(I1),1X,I2,1X,4(I1,1X),4(1X,F9.3),11(1X,F6.3))
*      CLEAR PERCENTILE ARRAY
      DO 66 I = 1,400
*        WRITE(NO,67) I,PER(I,1),PER(I,2)
67       FORMAT(1X,'PERCNT CNT=',1X,I3,'TARDY=',F8.2,'FLOW=',F8.2)
        PER(I,1) = 0
        PER(I,2) = 0
66     CONTINUE
      X(4) = 0.0
*      PRINT *,'END OF SUBROUTINE WRAPUP'
      RETURN
    END
```

```
*========
*     THESE ARE FUNCTIONS USED IN THE PROGRAM
*========
      REAL FUNCTION SPT(JOB) COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DT-
NOW,TNOW,TFIN,J,NRUN
      INTEGER JCUR_OPN
*     THIS FUNCTION CALCULATES THE SHORTEST OPERATION PROCESSING TIME
*     FOR THE JOB
      JCUR_OPN = INT(A(JOB,12))
      SPT = A(JOB,JCUR_OPN + 5)
      RETURN
      END

*-------------------
      REAL FUNCTION OPERD(JOB) COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),-
DTNOW,TNOW,TFIN,J,NRUN
      REAL DD,RD,OPN,OPN_FLOW,PROC,OP_PROC
*     BASED ON PROCEDURE IN PARK, 1987, P.55
*     OBTAIN TOTAL PROCESSING TIME
      TOT_PROC = A(JOB,19)
      COMP_PROC = A(JOB,19) - A(JOB,20)
      IF (COMP_PROC .LT. 1.0) COMP_PROC = 1.0
OPERD = COMP_PROC*(A(JOB,13) - A(JOB,14))/TOT_PROC + A(JOB,14)
      RETURN
      END
```

```
*============
*     THESE ARE RELATED TO COMPLEX ALGORITHM
*     ALL CODE ADAPTED FROM KUESTER AND MIZE, 1973
*     MCGRAW-HILL BOOK COMPANY, NEW YORK, 1973
*============
      SUBROUTINE COMP1 COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,-
TNOW,TFIN,J,NRUN COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),J-
CUR
      COMMON/ADAT/ALLDAT(100,8),LAB
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ  COMMON/-
START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*     ADAPTATION OF COMPLEX ALGORITHM OF BOX:
*      WRITE(NO,*)'START OF SUBROUTINE COMP1:'
10    FORMAT(1X,3(2X,F9.4),2(2X,I9))
*      PRINT *,'START SUB COMP1'
      IC = 0
      JCUR = 1
      X(34) = 1
      NUMPNT = INT(X(34))
      N = 2
      M = 2
      K = N + 1
      DO 20 I = 1,100
        DO 21 IJ = 1,8
          ALLDAT(I,IJ) = 0.0
21       CONTINUE
20    CONTINUE
      DO 25 I = 1,K
        DO 30 IJ = 1,N
          XPNT(I,IJ) = 0.0
          R(I,IJ) = 0.0
30       CONTINUE
        F(I) = 0.0
        G(I) = 0.0
        H(I) = 0.0
        XC(I) = 0.0
25    CONTINUE
      DO 40  I = 1,200
        SAMP(I) = 0.0
40    CONTINUE
      KOUNT = 1
      DELTA = .1
      JGAMMA = 8
      JG = JGAMMA
*     JG IS PASSED VIA THE COMMON STATEMENT
      ALPHA = 1.3
      BETA = X(46)
      IF (X(49) .LE. 10) CL = 0.05
      ITMAX = 15
```

```
      IT = 0
      LOW = 0
*     FIRST POINT IN COMPLEX
      IF (X(49) .EQ. 1.0) THEN
        CALL CONST1
        XPNT(1,1) = X(6)
        XPNT(1,2) = X(7)
*       CHECK TO ENSURE FIRST POINT OK: CHANGE X(6),X(7)
        CALL CHECK
        X(6) = XPNT(1,1)
        X(7) = XPNT(1,2)
      ELSEIF (X(49) .EQ. 3.0) THEN
        CALL CONST2
        XPNT(1,1) = X(9)
        XPNT(1,2) = X(10)
*       CHECK TO ENSURE FIRST POINT OK:
        CALL CHECK
        X(9) = XPNT(1,1)
        X(10) = XPNT(1,2)
      ENDIF
*     GENERATE RANDOM NUMBERS
      DO 90, I = 2, K
        R(I,1) = RA(5)
        R(I,2) = RA(5)
90    CONTINUE
15    CONTINUE
      WRITE(NO,16)
16    FORMAT (18X,'COMPLEX PROCEDURE OF BOX')
      IF (X(49) .EQ. 1.0) WRITE(NO,19)X(6),X(7)
      IF (X(49) .EQ. 3.0) WRITE(NO,19)X(9),X(10)
19    FORMAT(1X,'POINTS BEING EVALUATED ARE ',2(1X,F6.2))
      WRITE(NO,018)
018   FORMAT (2X,'PARAMETERS' )
      WRITE(NO,11) N, M, K, ITMAX, IC
      WRITE(NO,9) ALPHA, BETA, JGAMMA, DELTA
11    FORMAT(2X,'N=',I5,'M=',I5,'K=',I5,'ITMAX=',I5,'IC=',I5)
9     FORMAT('ALPHA=',F8.4,'BETA=',F8.4,2X,' JGAMMA=',I5,'DELTA=',F8.4)
*
      WRITE(NO,012)
012   FORMAT (2X,'RANDOM NUMBERS')
      DO 200 IJ=2,K
        WRITE(NO,13) R(IJ,1),R(IJ,2)
200   CONTINUE
13    FORMAT (1X,2(2X,F6.4))
*     CALL CONSX1  *HAVE FIRST POINT ABOVE
*     WRITE(NO,*)'END OF SUBROUTINE COMP1: '
*     PRINT *,'END SUB COMP1'
      RETURN
      END
```

```
*-----------------------------------------------
      SUBROUTINE CONSX1 COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW-
,TNOW,TFIN,J,NRUN COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),J-
CUR  COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*     FIRST HALF OF ORIGINAL SUBROUTINE CONSX
C     COORDINATES SUBROUTINES FOR COMPLEX ALGORITHM
      WRITE(NO,*) 'START SUBROUTINE CONSX1: '
*      PRINT *,'START SUB CONSX1'
C     CALCULATE COMPLEX POINTS AND CHECK AGAINST CONSTRAINTS
      NUMPNT = INT(X(34))
      WRITE(NO,*)'PNT=G(IJ)+R(NUMPNT,IJ)*(H(IJ)-G(IJ))'
      DO 50 IJ =1,N
        XPNT(NUMPNT,IJ) = G(IJ) + R(NUMPNT,IJ)*(H(IJ)-G(IJ))
        WRITE(NO,*)'IJ,G(IJ),R(NUMPNT,IJ),H(IJ)'
        WRITE(NO,*) IJ,G(IJ),R(NUMPNT,IJ),H(IJ)
50    CONTINUE
      CALL CHECK
*     NEED TO RETURN UNTIL FUNCTION VALUES FOUND
      IF (X(49) .EQ. 1.0) THEN
        X(6) = XPNT(NUMPNT,1)
        X(7) = XPNT(NUMPNT,2)
      ELSEIF (X(49) .EQ. 3.0) THEN
        X(9) = XPNT(NUMPNT,1)
        X(10) = XPNT(NUMPNT,2)
      ENDIF
*      PRINT *,'END SUB CONSX1'
*     TRUNCATE FIRST BATCH TO ELIMINATE INITIALIZATION EFFECT
      X(5) = 1
      RETURN
      END

*-----------------------------------------------------
      SUBROUTINE CONSX2 COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW-
,TNOW,TFIN,J,NRUN COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),J-
CUR  COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*     MIDLE THIRD OF ORIGINAL SUBROUTINE CONSX
*     OBTAINS NEW POINT FOR COMPLEX ALGORITHM, REPLACE LOWEST
*      WRITE(NO,*) 'START SUBROUTINE CONSX2'
*      PRINT *,'START SUB CONSX2'
*     TRUNCATE FIRST BATCH TO ELIMINATE INITIALIZATION EFFECT
      X(5) = 1
C     FIND POINT WITH LOWEST FUNCTION VALUE
80    IEV1 = 1
      DO 100 ICM=2,K
        IF (F(IEV1)-F(ICM)) 100,100,90
```

```
90       IEV1 = ICM
100    CONTINUE
C
C      FIND POINT WITH HIGHEST FUNCTION VALUE
      IEV2 = 1
      DO 120 ICM=2,K
        IF (F(IEV2)-F(ICM)) 110,110,120
110       IEV2 = ICM
120    CONTINUE
*
*      CHECK IF THIS LOW POINT WAS LOW POINT BEFORE
      IF (LOW .EQ. IEV1) THEN
*        THE POINT WAS LOW IN PREVIOUS ITERATION
        DO 145 JJ = 1,N
          XPNT(IEV1,JJ) = (XPNT(IEV1,JJ) + XC(JJ))/2.0
145       CONTINUE
        GOTO 180
      ELSE
        LOW = IEV1
      ENDIF
C      REPLACE POINT WITH LOWEST FUNCTION VALUE
150    CALL CENTR
161 FORMAT(1X,'REPLACE PNT1, PNT2,OBJ.FN',2(1X,F6.3),1X,F13.4)
      WRITE(NO,161) XPNT(IEV1,1),XPNT(IEV1,2),F(IEV1)
      JCUR = IEV1
      DO 160 JJ=1,N
        XPNT(IEV1,JJ)=(1.0+ALPHA)*(XC(JJ))-ALPHA*(XPNT(IEV1,JJ))
160    CONTINUE
180    CALL CHECK
*      OBTAIN VALUE FOR NEW POINT
      IF (X(49) .EQ. 1.0) THEN
        X(6) = XPNT(JCUR,1)
        X(7) = XPNT(JCUR,2)
      ELSEIF (X(49) .EQ. 3.0) THEN
        X(9) = XPNT(JCUR,1)
        X(10) = XPNT(JCUR,2)
      ENDIF
      F(JCUR) = 0.0
201 FORMAT(1X,'NEW POINT: NUMBER, PNT1,PNT2',1X,I3,2(1X,F8.2))
      WRITE(NO,201) JCUR,XPNT(JCUR,1),XPNT(JCUR,2)
      WRITE(NO,*) 'END CONSX2: NEW COMPLEX:PNT1,PNT2,OBJ.FUN'
      DO 310 I = 1,3
        WRITE(NO,320) I, XPNT(I,1),XPNT(I,2),F(I)
310    CONTINUE
320    FORMAT(1X,I2,2(1X,F8.3),1X,F13.4)
*       PRINT *, 'END OF SUB CONSX2'
      RETURN
      END
```

```
*-------------------------------------------------------
      SUBROUTINE CHECK COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,-
TNOW,TFIN,J COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
C     ALL ARGUMENTS DEFINED IN MAIN LINE AND CONSX
      WRITE(NO,*)'START OF SUBROUTINE CHECK: JCUR =',JCUR
*      PRINT *,'START SUB CHECK'
C     CHECK AGAINST EXPLICIT CONSTRAINTS
      DO 50 IJ=1,N
        IF ((XPNT(JCUR,IJ)-G(IJ)) .LE. 0.0) THEN
20        XPNT(JCUR,IJ) = G(IJ) + DELTA
          GOTO 50
        ENDIF
30       IF ((H(IJ)-XPNT(JCUR,IJ)) .LE. 0.0) THEN
40         XPNT(JCUR,IJ) = H(IJ) - DELTA
        ENDIF
50     CONTINUE
      IF (XPNT(JCUR,1) + XPNT(JCUR,2) .LE. 0.0) THEN
        WRITE(NO,*)'XPNT(1)+XPNT(2) LT 0: CHANGE THEM'
        NUMJ = INT(X(34))
        IF (NUMJ .EQ. 1) NUMJ = 2
        XPNT(JCUR,1) = H(1)*(REAL(JCUR/NUMJ)) - DELTA
        XPNT(JCUR,2) = H(2)*(1/(REAL(JCUR*NUMJ))) - DELTA WRITE(NO,*)'JCUR,-
NUMJ,JCUR*NUMJ',JCUR,NUMJ,(JCUR*NUMJ)
        WRITE(NO,*)'REVISED POINT: X6,X7',XPNT(JCUR,1),XPNT(JCUR,2)
      ENDIF
      WRITE(NO,*) 'END OF SUB CHECK'
110    RETURN
      END

*-------------------------------------------------------
      SUBROUTINE CENTR COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,-
TNOW,TFIN,J,NRUN
COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
     COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      WRITE(NO,*) 'START OF SUBROUTINE CENTR: JCUR =',JCUR
*      PRINT *, 'START OF SUB CENTR'
      DO 20 IJ=1,N
        XC(IJ) = 0.0
        DO 10 IL=1,K
          XC(IJ) = XC(IJ) + XPNT(IL,IJ)
10       CONTINUE
        RK = REAL(K)
        XC(IJ) = (XC(IJ)-XPNT(IEV1,IJ))/(RK-1.0)
20     CONTINUE
*      WRITE(NO,*) 'CENTROID VALUES ARE'
      DO 100 I = 1,N
```

```
*        WRITE(NO,110) XC(I)
100    CONTINUE
110    FORMAT(2X,F9.4)
*      PRINT *,'END OF SUB CENTR'
       RETURN
       END

*-------------------------------------------------------------
       SUBROUTINE FUNC(AVG1,SDEV1,T_TEST) COMMON/SIM/D(50),DL(50),S(50),S-
L(50),X(50),DTNOW,TNOW,TFIN,J,NRUN COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(-
6),G(4),H(4),XC(3),JCUR
       COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ  COMMON/-
START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW COMMON/CRIT/RA-
NK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
       COMMON/ADAT/ALLDAT(100,8),LAB
*      CALLED FROM THE SIMULATION.  CONTROLS THE LENGTH OF
*      TIME BETWEEN CALLS AND THE BATCH SIZE OF THE SAMPLE
*      CHECKS THE CONFIDENCE INTERVAL OF SIMULATION VALUE &
*      WHEN THE CONFIDENCE INTERVAL IS SMALL, ACCEPTS THE
*      MEAN AS VALID.  WHEN ENOUGH POINTS ARE COLLECTED, CALLS
*      CONSX SUB TO EVALUATE THE POINTS
C      JCUR IS CURRENT POINT BEING EVALUATED
*      PRINT *,'START OF SUB FUNC'
       NUMPNT = INT(X(34))
       BETA = X(46)
       IF (X(49) .LE. 10) CL = 0.05
       JGAMMA = JG
       IT = IT + 1
       IF (NUMPNT .LT. 4) IT = 1
55     FORMAT (2X,F12.4)
       AC = X(6)
       AD = X(7)
       AE = X(9)
       AF = X(10)
*      CALCULATE CONFIDENCE INTERVAL AND CHECK IF SATISFACTORY
59     IF ((T_TEST*SDEV1.LT.BETA+DELTA).OR.(NUMPNT.LT.4))THEN WRITE(NO,*)'-
PNT:X6,X7,X9,X10,VALUE,BETA,CL,DF,SDEV1,T_TEST,SIZE' WRITE(NO,69)AC,AD,AE,A-
F,AVG1,BETA,CL,X(48),SDEV1,T_TEST,X(35)
69   FORMAT(1X,4(1X,F7.2),1X,F13.4,3(1X,F6.2),2(1X,F9.4),1X,F6.1)
         IF (NUMPNT .GT. 3) KOUNT = KOUNT + 1
60        F(JCUR) = AVG1
         IF (X(49) .EQ. 1.0) THEN
           ALLDAT(NUMPNT,1) = X(6)
           ALLDAT(NUMPNT,2) = X(7)
         ELSEIF (X(49) .EQ. 3.0) THEN
           ALLDAT(NUMPNT,1) = X(9)
           ALLDAT(NUMPNT,2) = X(10)
         ENDIF
         ALLDAT(NUMPNT,3) = F(JCUR)
```

```
        ALLDAT(NUMPNT,4) = BETA
        ALLDAT(NUMPNT,5) = CL
        ALLDAT(NUMPNT,6) = DF
        ALLDAT(NUMPNT,7) = SDEV1
        ALLDAT(NUMPNT,8) = T_TEST       WRITE(NO,*)'HISTORY:PNT1,PNT2,VALU-
E,BETA,CL,DF,SDEV1,T_TEST'
        DO 20 I = 1, NUMPNT
          WRITE(NO,21) (ALLDAT(I,IJ), IJ=1,8)
20       CONTINUE
21 FORMAT(1X,2(1X,F9.2),1X,F13.4,3(1X,F6.2),1X,F13.4,1X,F13.4)
        WRITE(NO,*)'CURRENT COMPLEX: PNT1,PNT2,F(I)'
        DO 30 I= 1,3
          WRITE(NO,31) XPNT(I,1),XPNT(I,2),F(I)
30       CONTINUE
31       FORMAT(1X,3(2X,F13.4))
        X(34) = X(34) + 1
        X(48) = 1
        X(35) = JBAT
*        STOPPING RULES
        Q = .2
        A = ABS(XPNT(1,1)-XPNT(2,1))
        B = ABS(XPNT(2,1)-XPNT(3,1))
        C = ABS(XPNT(1,1)-XPNT(3,1))
        IF ((A .LT. Q) .AND. (B.LT.Q).AND.(C.LT.Q)) THEN
          KOUNT = JGAMMA
        ENDIF
        IF (IT .GT. ITMAX) THEN
          GOTO 400
        ELSEIF (NUMPNT .LT. 4) THEN
          GOTO 190
        ELSEIF (KOUNT .EQ. JGAMMA) THEN
          GOTO 210
        ELSEIF (NUMPNT .GT. 3) THEN
          CALL CONSX2
        ENDIF
        RETURN
      ELSE
*        CONFIDENCE INTERVAL IS TOO LARGE
        IF (X(48) .LT. X(41)) RETURN
        IF (X(49) .EQ. 1.0) THEN
WRITE(NO,*)
  'CI NOK:X(6),X(7),VALUE,BETA,CL,DF,SDEV1,T_TEST,SIZE' WRITE(NO,69)X(6),X(7),-
AVG1,BETA,CL,X(48),SDEV1,T_TEST,X(35)
        ELSEIF (X(49) .EQ. 3.0) THEN
WRITE(NO,*)
  'CI NOK:X(9),X(10),VALUE,BETA,CL,DF,SDEV1,T_TEST,SIZE' WRITE(NO,69)X(9),X(1-
0),AVG1,BETA,CL,X(48),SDEV1,T_TEST,X(35)
        ENDIF
*        INCREASE NUMBER OF SAMPLES
```

```
      IF (X(35) .LT. 60.0) THEN
*       SECOND, DOUBLE BATCH SIZE
        X(35) = 2*X(35)
        X(48) = REAL(INT(X(48)/2))
        IF (X(48) .LT. 5.0) RETURN
        CALL MOMENT(AVG1,ADEV1,SDEV1,VAR1,SKEW1,CURT1)
        DF = INT(X(48))
        CALL TEST(T_TEST,DF)
        GOTO 59
        RETURN
      ELSEIF (X(41) .LT. 20) THEN
        X(41) = 20
        RETURN
      ELSEIF (X(41) .LT. 30) THEN
        X(41) = 30
        RETURN
      ELSEIF ( X(41) .LT. 40) THEN
        X(41) = 40
   WRITE(NO,*) 'PROBLEM !!!  CONFIDENCE INTERVAL TOO LARGE'
        WRITE(NO,*) 'NEXT TIME SAMPLE WILL BE ACCEPTED'
        RETURN
      ENDIF
139     WRITE (NO,*) 'CONFIDENCE INTERVAL TOO LARGE: STOP'
      IF (NUMPNT .GT. 3) IT = ITMAX + 1
      GOTO 60
    ENDIF
190   IF (NUMPNT .LT. 4) THEN
*       START OF COMPLEX: OBTAINING FIRST THREE POINTS
      JCUR = JCUR + 1
      KOUNT = 1
      CALL CONSX1
      RETURN
    ENDIF
    RETURN
*    HAVE VALUES FOR COMPLEX: NOW MOVE TO OPTIMUM:IMPROVE CL
210   CONTINUE
    WRITE(NO,*) 'IMPROVE C.I. FOR LAST PNT'
    IF (BETA .GT. 10.0) THEN
*  REQUIRE MORE PRECISION FOR CONVERGENCE: CK another POINT
      KOUNT = 1
      IT = ITMAX + 1
      X(35) =  80
      X(41) = 40
      CALL CONSX2
      RETURN
    ENDIF
300   IF (KOUNT .EQ. JGAMMA) THEN
      CALL COUNT(2,1)
*       WRITE(NO,*) 'CONVERGENCE COMPLETED: FINAL COMPLEX'
```

```
        WRITE(NO,*) 'COMPLEX COMPLETED RUN: K1,K2,OBJ.FUNC, NRUN'
        GOTO 500
      ENDIF
400   IF (IT .GT. ITMAX) THEN
*       NUMBER OF ITERATIONS EXCEEDED MAXIMUM ALLOWED: STOP
        CALL COUNT(2,1)
        GOTO 500
      ENDIF
**      PRINT OUT HISTORY OF ALL POINTS EXAMINED
500   CONTINUE
      WRITE(NO,*)'HISTORY:PNT1, PNT2,VALUE,BETA,CL,DF,SDEV1,T_TEST'
      DO 520 I = 1, NUMPNT
        WRITE(NO,21) (ALLDAT(I,IJ), IJ=1,8)
520   CONTINUE
      WRITE(NO,*)' LAST COMPLEX: PNT1, PNT2, F(I)'
      DO 530 I= 1,3
        WRITE(NO,31) XPNT(I,1),XPNT(I,2),F(I)
530   CONTINUE
      RETURN
      END

*----------------
      SUBROUTINE CONST1
       COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
      COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*       DIMENSION X(K,M), G(M), H(M)
*       WRITE(NO,*) 'START SUBROUTINE CONST'
**      PRINT *,'START SUB CONST'
      G(1) = -50.0
      H(1) = 100.0
      G(2) = -50.0
      H(2) =  100.0
      RETURN
      END

*----------------
      SUBROUTINE CONST2
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
      COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*       DIMENSION X(K,M), G(M), H(M)
*       WRITE(NO,*) 'START SUBROUTINE CONST2'
**      PRINT *,'START SUB CONST2'
*     G(1) REFERS TO MIN: H(1) REFERS TO MAX
      G(1) = 0.0
      H(1) = 200
```

```
      G(2) = 0.0
      H(2) = 1.0
      RETURN
      END

*-----------------
      SUBROUTINE PERCNT(TME_TARD,FLOW)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      COMMON/QUART/PER(400,2),NTH,TOT_COST
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*     THIS SECTION COLLECTS THE PERCENTILE STATISTICS
*      WRITE(NO,*) 'START SUBROUTINE PERCNT: TARDY=',TME_TARD
*      WRITE(NO,*) 'FLOW =',FLOW
      JK = 0
      A = TME_TARD
      B = FLOW
      N = NTH
*      WRITE(NO,*) 'PER(N,1)=',PER(N,1), ' N= ',N,'TME_TARD=',A
*      WRITE(NO,*) 'PER(N,2)=', PER(N,2)
      IF (A .LE. PER(N,1)) GOTO 480
      JK = INT(N*.9)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.8)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.75)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.6)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.5)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.4)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.3)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.2)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = INT(N*.1)
      IF (A .LT. PER(JK,1)) GOTO 475
      JK = 1
475    DO 420 I = JK,N
        IF (A .GT. PER(I,1)) THEN
          TEMP = PER(I,1)
          PER(I,1) = A
          DO 460 K = I+1,N
            TMP = PER(K,1)
            PER(K,1) = TEMP
            TEMP = TMP
460         CONTINUE
```

```
          GOTO 480
        ENDIF
420    CONTINUE
480    CONTINUE
*      WRITE(NO,*)'TARDY =',A,' JK=',JK
*      IF (JK .GT. 0) WRITE(NO,*) '       PER(JK,1)=',PER(JK,1)
      IF (B .LE. PER(N,2)) GOTO 540
      JK = INT(N*.9)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.8)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.75)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.7)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.6)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.5)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.4)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.3)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.2)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = INT(N*.1)
      IF (B .LT. PER(JK,2)) GOTO 485
      JK = 1
485    DO 510 I = JK,N
        IF (B .GT. PER(I,2)) THEN
          TEMP = PER(I,2)
          PER(I,2) = B
          DO 520 K = I+1,N
            TMP = PER(K,2)
            PER(K,2) = TEMP
            TEMP = TMP
520        CONTINUE
          GOTO 540
        ENDIF
510    CONTINUE
540    CONTINUE
*      WRITE(NO,*) 'FLOW JK=',JK,'PER(JK,2)=',PER(JK,2)
*      IF (PER(N,2) .GT. 0.0) THEN
*        DO 600 I = 1,N
*          WRITE(NO,*) 'PER(I,2)=', PER(I,2)
*600      CONTINUE
*      ENDIF
*      WRITE(NO,*) 'END SUB: PER(N,1)=',PER(N,1),'PER(N,2)=',PER(N,2)
      RETURN
```

```
      END


*---------------
      SUBROUTINE MOMENT(AVG1,ADEV1,SDEV1,VAR1,SKEW1,CURT1)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*     FROM NUMERICAL RECIPES: THE ART OF SCIENTIFIC COMPUTING
*     PRESS, FLANNERY, TEUKOLSKY, & VETTERING, P.458
*     CAMBRIDGE UNIVERSITY PRESS, 1986
*     GIVEN AN ARRAY, IT RETURNS ITS MEAN, AVERAGE DEVIATION, STANDARD
*     DEVIATION, VARIANCE, SKEWNESS AND KURTOSIS
*
*     FIRST PASS TO OBTAIN THE MEAN: NEED A MINIMUM SIZE SAMPLE
*      WRITE(NO,*) 'SUBROUTINE MOMENT: NUMBER OF SAMPLES=',X(48)
*      PRINT *,'START SUB MOMENT: X(48) =',X(48)
      COUNT = 0
      AVG1 = 0.0
      ADEV1 = 0.0
      VAR1 = 0.0
      SDEV1 = 0.0
      SKEW1 = 0.0
      CURT1 = 0.0
5     FORMAT(1X,'SAMP NUM=',I5,' AVG1=',F9.4,'SAMP(I)=',F9.4)
      DO 10 I = 1,INT(X(48))
        AVG1 = AVG1 + SAMP(I)
        COUNT = COUNT + 1
*        PRINT *,'NUM,AVG,SAMP(NUM),COUNT',I,AVG1,SAMP(I),COUNT
10    CONTINUE
*     PRINT *,'AVG1,COUNT', AVG1, COUNT
11    AVG1 = AVG1/COUNT
*     PRINT *,'AVG1,COUNT', AVG1, COUNT
*     SECOND PASS TO GET FIRST, SECOND, THIRD AND FOURTH MOMENTS
*     OF DEVIATION FROM THE MEAN
      DIFF = 0.0
      SQ = 0.0
      MCOUNT = INT(COUNT)
*      PRINT *,'MCOUNT =',MCOUNT
      DO 20 I = 1, MCOUNT
        DIFF = SAMP(I) - AVG1
        ADEV1 = ADEV1 + ABS(DIFF)
        SQ = DIFF*DIFF
        VAR1 = VAR1 + SQ
        SQ = SQ*DIFF
        SKEW1 = SKEW1 + SQ
        SQ = SQ*DIFF
        CURT1 = CURT1 + SQ
```

```
*      PRINT *,' IN LOOP 20, I =',I
20     CONTINUE
*     PUT THE PIECES TOGETHER ACCORDING TO THE CONVENTIONAL DEFN.
*      PRINT *,'MCOUNT = ',MCOUNT
      ADEV1 = ADEV1/MCOUNT
      VAR1 = VAR1/(MCOUNT - 1)
      SDEV1 = SQRT(VAR1)
*      PRINT *, 'VAR1, AVG1, SDEV1', VAR1, AVG1,SDEV1
*      WRITE(NO,21) VAR1,AVG1,SDEV1
21     FORMAT(1X,'VAR1 =',1X,F9.4,' AVG1=',1X,F9.4,' SDEV1 =',F9.4)
      IF (VAR1 .GT. 0.5) THEN
        SKEW1 = SKEW1/(MCOUNT*(SDEV1**3))
        CURT1 = CURT1/(MCOUNT*(VAR1**2)) - 3.0
      ENDIF
*      PRINT *,'SUBROUTINE MOMENT: PRINTING TO FILES'
**       WRITE(NO,*) 'SUBROUTINE MOMENT: PRINTING TO FILES'
190    FORMAT(1X,'VALUE 1: AVG=',1X,F12.3,1X,'VAR =',1X,F12.3)
*200    WRITE(10,190)AVG1,VAR1
*      WRITE(NO,190)AVG1,VAR1
      A1 = AVG1
      V1 = VAR1
*     WRITE(NO,*) 'END OF SUBROUTINE MOMENT'
*      PRINT *,'END OF SUBROUTINE MOMENT'
      RETURN
      END

*----------------
      SUBROUTINE TEST(T,DF)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
      COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
*     THE PURPOSE OF THIS SUBROUTINE IS TO RETURN CORRECT T VALUE
*     TABLE FOR TWO TAIL T-TEST WITH ALPHA = 0.10 OR 0.05
*     IS FROM STATISTICS FOR MANAGEMENT, RICHARD LEVIN, 3RD ED.
*     TABLE FOR ALPHA = 0.20 IS FROM NETER, WASSERMAN, KUTNER, P.1074
      REAL TAB(121,3)
      DATA TAB/363*0.0/
      IF (X(49) .LE. 10) CL = 0.05
*      PRINT *, 'BEGIN SUBROUTINE TEST: CL=',CL,' DF=',DF
*     CL = 0.10 IS TAB(x,1): CL = 0.05 = TAB(x,2): CL = 0.20 = TAB(x,3)
      TAB(1,1) = 6.314
      TAB(2,1) = 2.920
      TAB(3,1) = 2.353
      TAB(4,1) = 2.132
      TAB(5,1) = 2.015
      TAB(6,1) = 1.943
      TAB(7,1) = 1.895
      TAB(8,1) = 1.860
```

```
TAB(9,1) = 1.833
TAB(10,1) = 1.812
TAB(11,1) = 1.796
TAB(12,1) = 1.782
TAB(13,1) = 1.771
TAB(14,1) = 1.761
TAB(15,1) = 1.753
TAB(16,1) = 1.746
TAB(17,1) = 1.740
TAB(18,1) = 1.734
TAB(19,1) = 1.729
TAB(20,1) = 1.725
TAB(21,1) = 1.721
TAB(22,1) = 1.717
TAB(23,1) = 1.714
TAB(24,1) = 1.711
TAB(25,1) = 1.708
TAB(26,1) = 1.706
TAB(27,1) = 1.703
TAB(28,1) = 1.701
TAB(29,1) = 1.699
TAB(30,1) = 1.697
TAB(40,1) = 1.684
TAB(60,1) = 1.671
TAB(120,1) = 1.658
TAB(121,1) = 1.645
TAB(1,2) = 12.706
TAB(2,2) = 4.303
TAB(3,2) = 3.182
TAB(4,2) = 2.776
TAB(5,2) = 2.571
TAB(6,2) = 2.447
TAB(7,2) = 2.365
TAB(8,2) = 2.306
TAB(9,2) = 2.262
TAB(10,2) = 2.228
TAB(11,2) = 2.201
TAB(12,2) = 2.179
TAB(13,2) = 2.160
TAB(14,2) = 2.145
TAB(15,2) = 2.131
TAB(16,2) = 2.120
TAB(17,2) = 2.110
TAB(18,2) = 2.101
TAB(19,2) = 2.093
TAB(20,2) = 2.093
TAB(21,2) = 2.080
TAB(22,2) = 2.074
TAB(23,2) = 2.069
```

```
      TAB(24,2) = 2.064
      TAB(25,2) = 2.060
      TAB(26,2) = 2.056
      TAB(27,2) = 2.052
      TAB(28,2) = 2.048
      TAB(29,2) = 2.045
      TAB(30,2) = 2.042
      TAB(40,2) = 2.021
      TAB(60,2) = 2.000
      TAB(120,2) = 1.980
      TAB(121,2) = 1.960
      TAB(1,3) = 3.078
      TAB(2,3) = 1.886
      TAB(3,3) = 1.638
      TAB(4,3) = 1.533
      TAB(5,3) = 1.476
      TAB(6,3) = 1.440
      TAB(7,3) = 1.415
      TAB(8,3) = 1.397
      TAB(9,3) = 1.383
      TAB(10,3) = 1.372
      TAB(11,3) = 1.363
      TAB(12,3) = 1.356
      TAB(13,3) = 1.350
      TAB(14,3) = 1.345
      TAB(15,3) = 1.341
      TAB(16,3) = 1.337
      TAB(17,3) = 1.333
      TAB(18,3) = 1.330
      TAB(19,3) = 1.328
      TAB(20,3) = 1.325
      TAB(21,3) = 1.323
      TAB(22,3) = 1.321
      TAB(23,3) = 1.319
      TAB(24,3) = 1.318
      TAB(25,3) = 1.316
      TAB(26,3) = 1.315
      TAB(27,3) = 1.314
      TAB(28,3) = 1.313
      TAB(29,3) = 1.311
      TAB(30,3) = 1.310
      TAB(40,3) = 1.303
      TAB(60,3) = 1.296
      TAB(120,3) = 1.289
      TAB(121,3) = 1.282
*     AT 121 AND ABOVE THIS IS A NORMAL DISTRIBUTION
      IF (DF .GT. 30) THEN
        IF (DF .GT. 120) THEN
          DF = 121
```

```
      ELSEIF (DF .GT. 60) THEN
        DF = 60
      ELSEIF (DF .GT. 40) THEN
        DF = 40
      ELSEIF (DF .GT. 30) THEN
        DF = 30
      ENDIF
    ENDIF
    JDF = INT(DF)
    IF (CL .EQ. 0.20) THEN
      II = 3
    ELSEIF (CL .EQ. 0.10) THEN
      II = 1
    ELSEIF (CL .EQ. 0.05) THEN
      II = 2
    ENDIF
    T = TAB(JDF,II)
*     PRINT *, 'DF =',JDF, 'T RETURNED =',T
*     PRINT *,'END SUBROUTINE TEST'
    RETURN
    END
```

```
      SUBROUTINE KNOWN
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ
      COMMON/CRIT/RANK(6,2),SHIFTEND,SHIFTBEG,AVGWL,WC_CRIT(6),REL
      COMMON/PLEX/N,M,K,XPNT(6,2),R(6,2),F(6),G(4),H(4),XC(3),JCUR
      COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
      IF (X(45) .LT. 4.0) RETURN
      WRITE(NO,*)'SUBROUTINE KNOWN: X(35) =', X(35)
*      PRINT *,'SUBROUTINE KNOWN: X(35) =',X(35)
*     THIS ROUTINE CREATES A KNOWN DISTRIBUTION OF POINTS
*     CREATE ALL BUT LAST DATA SET COMPILED BY SUB PERIOD
*     DATA IS WEEKLY DATA COMPILED FOR BATCH SIZE X(35)
      CALL CLEAR
      DO 100 IJ = 1,INT(X(41))
*       EACH LOOP HERE REPRESENTS ONE BATCH OF X(35) WEEKS
        DO 110 K = 1, INT(X(35))
*         EACH LOOP HERE IS ONE WEEK
          TRANSFER = UN(9,5)
          IF (NUMPNT .LT. 5) THEN
            TARDY = UN(9,5) - 10*NUMPNT
          ELSEIF ((NUMPNT .GT. 4) .AND. (NUMPNT .LT. 7)) THEN
            TARDY = UN(9,5) - NUMPNT
          ELSEIF (NUMPNT .EQ. 7) THEN
            TARDY = UN(9,5) -20*NUMPNT
          ELSEIF (NUMPNT .GT. 7) THEN
            TARDY = UN(9,5)
          ENDIF
*          WRITE(NO,90) TARDY,TRANSFER
90         FORMAT(1X,'TARDY=',1X,F9.4,' TRANSFER =',F9.4)
          CALL TALLY(9,TARDY)
          CALL COUNT(6,INT(TRANSFER))
          CALL TALLY(12,TRANSFER)
110       CONTINUE
*       THE DATA IS STORED IN ARRAY TO ALLOW VAR CALCULATIONS ETC.
        AVG_INV = 5*UN(9,5)
        TOT_COST = +2*TAVG(9)+.1*AVG_INV +.1*TAVG(12)
        SAMP(IJ) = TOT_COST
        IF(X(49).EQ.1.0) SAMP(IJ)=-1*TOT_COST
        X(48) = X(48) + 1
100   CONTINUE
      X(48) = X(41)
      WRITE(NO,*) 'END OF SUBROUTINE KNOWN'
      RETURN
      END
```

```
*====================
*     SUBROUTINES TO IMPLEMENT THE FIBONACCI
*     BASED ON KUESTER & MIZE , MCGRAW HILL, P.256
*====================
SUBROUTINE FIBON COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,T-
NOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ COMMON/ONCI/-
A,B,IK,JJ,KK,T,U,V,W,FIB(50),NUM,DAT(100,12),
JPOS,ALL COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*     PURPOSE IS TO DEFINE THE ORIGINAL SEARCH
*     CONSTRAINTS AND REQUIRED ACCURACY.  FIRST 2 PTS. PLACED
      ALPHA = 0.01
      WRITE(NO,*) 'START OF SUB FIBON: ALPHA =',ALPHA
      NI = 50
      NUM = 1
      KOUNT = 1
*     ALPHA IS DESIRED ACCURACY AS PCT. OF INTERVAL EXAMINED
      DO 1 I = 1,200
        SAMP(I) = 0.0
1     CONTINUE
      DO 2 I = 1,50
        FIB(I) = 0.0
2     CONTINUE
      DO 5 I = 1, 100
        DO 6 IJ = 1,12
          DAT(I,IJ) = 0.0
6       CONTINUE
5     CONTINUE
*     A IS LOW END OF INTERVAL: B IS HIGH END OF INTERVAL
      A = 0
      B = 1
      DEL = B - A
*      WRITE(NO,10)
10    FORMAT(1X,'FIBONACCI SINGLE-VARIABLE PROCEDURE')
*     DEFINE THE FIRST THREE FIBONACCI NUMBERS
      FIBO = 1.0
      FIB(1) = 1.0
      FIB(2) = 2.0
*      WRITE(NO,11) FIBO,FIB(1),FIB(2)
11    FORMAT(1X,'FIBO, FIB(1), FIB(2)=',3(2X,F9.4))
*     CALCULATE REMAINING FIBONACCI NUMBERS
15    BB = 1.0/ALPHA
      IF ((BB - 2.0) .GT. 0.0) THEN
        JJ = 2
20        JJ = JJ + 1
        FIB(JJ) = FIB(JJ-1) + FIB(JJ-2)
        CC = FIB(JJ)
        IF( (CC - BB) .LT. 0.0) GOTO 20
      ELSE
```

```
        WRITE(NO,25)
25       FORMAT(1X,'ACCURACY SPECIFIED IS NOT SUFFICIENT')
        WRITE(NO,26)
26       FORMAT(1X,'PROGRAM RESET ALPHA, ALPHA = 0.01')
        ALPHA = 0.01
        GOTO 15
      ENDIF
21       FORMAT(1X,'FIB NUMBER JJ EQUALS FIB(JJ)',I5,2X,F9.4)
50    CONTINUE
      KK = JJ - 2
      IK = JJ - 2
      BL = B - A
      ALL = FIB(IK)*BL/FIB(JJ)
      W = A + ALL
      V = B - ALL
*     GATHER DATA FROM SIMULATION
      X(10) = W
*      X(8) = W
*      WRITE(NO,100) W,V,NUM
100    FORMAT(1X,'END SUB
         FIBON:PAR1=',F9.4,'PAR2=',F9.4,'NUM=',I5)
      DAT(NUM,1) = W
      DAT(NUM,7) = V
      JPOS = 0
*     JPOS DETERMINES NUM. POINTS EVALUATED BEFORE START NEW
*     INTERVAL:  JPOS (0) = 2 POINTS:  JPOS(1) = 1 POINT ON
*     LEFT OR W: JPOS(2) = 1 POINT ON THE RIGHT/V: JPOS (3)
*     = 1 POINT EVALUATED AND RETURNED TO SUB FIN:  JPOS(4)=
*     THE LAST POINT BEING EVALUATED
      RETURN
      END

*----------------
      SUBROUTINE CONFID(AVG1,SDEV1,T_TEST) COMMON/SIM/D(50),DL(50),S(50)-
,SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ COMMON/ONCI/-
A,B,IK,JJ,KK,T,U,V,W,FIB(50),NUM,DAT(100,12),
JPOS,ALL COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*     FIRST DETERMINE CONFIDENCE INTERVAL OF SAMPLE
WRITE(NO,*) 'SUB CONFID: HISTORY: PNT,AVG1,CL,DF,SDEV1,T_TEST'
      BETA = X(46)
      DO 20 I = 1,NUM
        WRITE(NO,21) (DAT(I,IJ), IJ = 1,6)
        WRITE(NO,22) (DAT(I,IJ), IJ = 7,12)
20    CONTINUE
21    FORMAT(1X,'PNT 1:',6(1X,F9.4))
22    FORMAT(7X,'PNT 2:',6(1X,F9.4))
*     CALCULATE CONFIDENCE INTERVAL AND CHECK IF SATISFACTORY
25    HALF = SDEV1*T_TEST
```

```
59     FORMAT(1X,6(1X,F9.3))
60     IF (HALF .LT. BETA) THEN
*        CONFIDENCE INTERVAL IS SMALL ENOUGH
         IF (JPOS .GT. 2) THEN
           CALL FIBTAB(AVG1,SDEV1,T_TEST)
           RETURN
         ENDIF
         OLD_BETA = BETA
         CALL FIBTAB(AVG1,SDEV1,T_TEST)
*        CHECK IF BETA WAS REDUCED IN FIBTAB
         IF (BETA .LT. OLD_BETA) THEN
           GOTO 60
         ENDIF
         WRITE(NO,*) 'END SUB CONFID'
         RETURN
       ELSE
*        CONFIDENCE INTERVAL IS TOO LARGE: DOUBLE BATCH SIZE
WRITE(NO,61) AVG1,SDEV1,T_TEST,BETA,X(35),X(41),X(48)
61       FORMAT(1X,2(1X,F10.3),1X,F8.4,4(1X,F6.1))
         IF (X(48) .LT. X(41)) RETURN
*        DOUBLE BATCH SIZE
         IF (X(35) .LT. 60) THEN
           X(35) = 2*X(35)
           IJK = INT(X(48)/2)
80          DO 100 I = 1,IJK
              IJ = 2*I
              SAMP(I) = (SAMP(IJ-1) + SAMP(IJ))/2
100         CONTINUE
           X(48) = REAL(IJK)
           CALL MOMENT(AVG1,ADEV1,SDEV1,VAR1,SKEW1,CURT1)
           DF = INT(X(48))
           CALL TEST(T_TEST,DF)
           GOTO 25
         ENDIF
*        INCREASE NUMBER OF SAMPLES
         IF ( X(41) .LT. 20) THEN
           X(41) = 20
           RETURN
         ELSEIF ( X(41) .LT. 30) THEN
           X(41) = 30
           RETURN
         ELSEIF ( X(41) .LT. 40) THEN
           X(41) = 40
           RETURN
         ELSEIF ( X(41).LT. 50) THEN
          X(41) = 50
           RETURN
         ELSEIF (X(35) .LT. 100) THEN
           X(35) = 2*X(35)
```

```
          IJK = INT(X(48)/2)
          DO 110 I = 1,IJK
            IJ = 2*I
            SAMP(I) = (SAMP(IJ-1) + SAMP(IJ))/2
110         CONTINUE
          X(48) = REAL(IJK)
          CALL MOMENT(AVG1,ADEV1,SDEV1,VAR1,SKEW1,CURT1)
          DF = INT(X(48))
          CALL TEST(T_TEST,DF)
          GOTO 25
        ELSEIF ( X(41) .LT. 60) THEN
          X(41) = 60
          RETURN
        ELSEIF ( X(41) .LT. 70) THEN
          X(41) = 70
          RETURN
        ELSEIF ( X(41) .LT. 80) THEN
          X(41) = 80
*          RETURN
*        ELSEIF ( X(41) .LT. 90) THEN
*          X(41) = 90
*          RETURN
*        ELSEIF ( X(41) .LT. 100) THEN
  WRITE(NO,*) 'PROBLEM !!!  CONFIDENCE INTERVAL TOO LARGE'
          WRITE(NO,*) 'NEXT TIME SAMPLE WILL BE ACCEPTED'
          RETURN
        ENDIF
        WRITE(NO,*)'C.I. TOO LARGE: FINISH INTERVAL & QUIT'
        KOUNT = KK + 1
        CALL FIBTAB(AVG1,SDEV1,T_TEST)
      ENDIF
*      PRINT *, 'END OF SUB CONFID'
*      WRITE(NO,*) 'END OF SUB CONFID'
      RETURN
      END

*---------------
      SUBROUTINE FIBTAB(AVG1,SDEV1,T_TEST) COMMON/SIM/D(50),DL(50),S(50),-
SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ COMMON/ONCI/-
A,B,IK,JJ,KK,T,U,V,W,FIB(50),NUM,DAT(100,12),
JPOS,ALL COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*      FIRST DETERMINE CONFIDENCE INTERVAL OF SAMPLE
*       PRINT *,'START OF SUB FIBTAB'
      WRITE(NO,*) 'START OF SUB FIBTAB: JPOS =',JPOS
*     IF JPOS =0,1,3 OR 4 EVALUATING THE LEFT HAND POINT
      IF ((JPOS .LE. 1.0 ) .OR. (JPOS .EQ. 3)) THEN
        MOVE = 0
        IF (JPOS .NE. 3) T = AVG1
```

```
*       FOUND OBJ FUNCTION FOR 1ST POINT IN INTERVAL
      ELSE
        MOVE = 6
        U = AVG1
      ENDIF
      IF (JPOS .EQ. 4) MOVE = 0
      IF (JPOS .EQ. 5) MOVE = 6
      DF = X(48)
      IF (X(49) .LE. 10) CL = 0.05
      DAT(NUM,2+MOVE) = AVG1
      DAT(NUM,3+MOVE) = CL
      DAT(NUM,4+MOVE) = DF
      DAT(NUM,5+MOVE) = SDEV1
      DAT(NUM,6+MOVE) = T_TEST
      WRITE(NO,*) 'HISTORY: PNT, AVG1, CL, DF, SDEV1, T_TEST'
      WRITE(NO,*) 'HISTORY: PNT, AVG1, CL, DF, SDEV1, T_TEST'
      DO 20 I = 1,NUM
        WRITE(NO,21) (DAT(I,IJ), IJ=1,6)
        WRITE(NO,19) (DAT(I,IJ), IJ=7,12)
20     CONTINUE
21     FORMAT(1X,3(1X,F9.4),3(1X,F9.3))
19     FORMAT(11X,6(1X,F9.3))
      IF (JPOS .GT. 3) THEN
        CALL FIN
        RETURN
      ENDIF
      IF (JPOS .EQ. 3) THEN
        CALL LAST(AVG1)
        RETURN
      ENDIF
      IF (JPOS .EQ. 0) THEN
        JPOS = 2
        X(10) = DAT(NUM,7)
*         X(8) = DAT(NUM,7)
*         WRITE(NO,*) 'LEAVE SUB FIBTAB: JPOS =',JPOS
        RETURN
      ELSEIF (JPOS .GT. 0) THEN
*         WRITE(NO,*) 'CHECK IF BETA ADEQUATE'
40        IF (ABS(DAT(NUM,2) - DAT(NUM,8)) .GT. BETA*2) THEN
*           BETA IS ADEQUATE
            WRITE(NO,*) 'BETA OK: CALL INTER'
            CALL INTER
            RETURN
          ELSE
*           ENDPOINTS ARE NOT DISTINGUISHABLE: CHANGE BETA
75  WRITE(NO,*) 'ENDPOINTS NOT DISTINGUISHABLE: CHANGE BETA'
50          FORMAT (2X,3(2X,F13.4))
            IF (BETA .GT. 90) THEN
              X(46) = 90
```

```
            GOTO 100
          ELSEIF (BETA .GT. 80) THEN
            X(46) = 80
            GOTO 100
          ELSEIF (BETA .GT. 70) THEN
            X(46) = 70
            GOTO 100
          ELSEIF (BETA .GT. 60) THEN
            X(46) = 60
            GOTO 100
          ELSEIF (BETA .GT. 50) THEN
            X(46) = 50
            GOTO 100
          ELSEIF (BETA .GT. 40) THEN
            X(46) = 40
            GOTO 100
          ELSEIF (BETA .GT. 30) THEN
            X(46) = 30
            GOTO 100
          ELSEIF (BETA .GT. 20) THEN
            X(46) = 20
*             GOTO 100
*           ELSEIF (BETA .GT. 5) THEN
*             X(46) = 5
*             GOTO 100
*           ELSEIF (BETA .GT. 3) THEN
*             X(46)= 3
*             GOTO 100
*           ELSEIF (BETA .GT. 1) THEN
*             X(46) = 1
*             GOTO 100
*           ELSEIF (BETA .EQ. 1) THEN
**        IF HERE END POINTS NOT DISTINGUISHABLE: START EXIT
*             KOUNT = KK - 1
            CALL INTER
     WRITE(NO,*) 'END POINT NOT DIFF: TRY ONE LAST POINT'
            RETURN
          ENDIF
        ENDIF
*  ENDPOINT DISTINGUISHABLE WITH THIS BETA.IS CONFID. INT. OK?
100       BETA = X(46)
        IF (ABS(DAT(NUM,2) - DAT(NUM,8)) .GT. 2*BETA) THEN
*           BETA OK: CHECK CONF. INT.
*           WRITE(NO,*) 'BETA OK: CHECK CONFID. INTERVAL'
          RETURN
        ELSE
*           BETA NOT OK: REDUCE BETA
          WRITE(NO,*) 'BETA NOT OK, REDUCE BETA, GOTO 75'
*           GOTO 75
```

```
        ENDIF
      ENDIF
*     WRITE (NO,*) 'LEAVE SUB FIBTAB'
      RETURN
      END

*----------------
      SUBROUTINE INTER COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,T-
NOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ COMMON/ONCI/-
A,B,IK,JJ,KK,T,U,V,W,FIB(50),NUM,DAT(100,12),
JPOS,ALL COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*     PURPOSE OF SUBROUTINE: NARROW SEARCH INTERVAL AND PLACE
*     NEW POINT INTO INTERVAL: USE KNOWN INTERNAL POINTS(')
*     TO SHORTEN INTERVAL FIRST A___A'_____B'___B THEN FIND
*     ANOTHER INTERNAL POINT
*     TRUNCATE FIRST BATCH TO ELIMINATE INTIALIZATION EFFECT
      X(5) = 1
      IF (X(49) .LE. 10) CL = 0.05
      BETA = X(46)
      IF (JPOS .GT. 2) THEN
*  FINISH WORK: DETERMINE WHICH OF CURRENT POINTS IS BEST
        IF (DAT(NUM,2) - DAT(NUM,8) .LT. 0) THEN
          W = DAT(NUM,7)
        ELSE
          W = DAT(NUM,1)
        ENDIF
        CALL FIN
        RETURN
      ENDIF
      NUM = NUM + 1
      KOUNT = KOUNT + 1
      IF (KOUNT .GE. KK) THEN
* THIS MEANS THE INTERVAL IS WITHIN THE CONF. REGION ALPHA
*       CHECK THAT THIS WAS DONE WITH CORRECT CL
        IF (CL .GT. 0.05) THEN
          KOUNT = KK - 2
          IF (KOUNT .LE. 0) KOUNT = 1
          CL = 0.05
          JJ = KK + 2
          IK = JJ - 2
          GOTO 10
        ENDIF
      ENDIF
      IF (KOUNT .GT. KK) THEN
        NUM = NUM - 1
        IF (DAT(NUM,2) - DAT(NUM,8) .LT. 0) THEN
          W = DAT(NUM,7)
        ELSE
```

```
          W = DAT(NUM,1)
        ENDIF
        CALL FIN
        RETURN
      ENDIF
*     ESTABLISH NEW INTERVAL AND NEW POINT
10     IK = IK - 1
      IF (IK - 1 .LT. 0) IK = 1
      JJ = JJ - 1
      IF (DAT(NUM-1,8) - DAT(NUM-1,2) .LE. 0) THEN
* POINT ON LEFT IS LARGER, SHORTEN INTERVAL ON LEFT TO A"
*       A___A"_____B'___B  CENTER SEARCH AROUND B'
*       WRITE(NO,41) DAT(NUM-1,8),DAT(NUM-1,2)
41       FORMAT(1X,'DAT(8) LT DAT(2)',2(1X,F9.4))
        A = A + ALL
        BL = B - A
        ALL = FIB(IK)*BL/FIB(JJ)
        W = V
        DAT(NUM,1) = W
*       STORE VALUES FROM PREVIOUS POINT FOR THIS INTERVAL
        DO 37 I = 2,6
          IF (DAT(NUM-1,7) .EQ. W) THEN
            DAT(NUM,I) = DAT(NUM-1,I+6)
          ELSE
            DAT(NUM,I) = DAT(NUM-1,I)
          ENDIF
37       CONTINUE
        V = B - ALL
        DAT(NUM,7) = V
        X(10) = DAT(NUM,7)
*        X(8) = DAT(NUM,7)
        JPOS = 2
        WRITE(NO,*) 'END SUB INTER: POINT HISTORY: NUM=',NUM
        DO 420 I = 1,NUM
          WRITE(NO,425) (DAT(I,IJ), IJ=1,6)
          WRITE(NO,424) (DAT(I,IJ), IJ=7,12)
420      CONTINUE
424      FORMAT(10X,6(1X,F9.3))
425      FORMAT(1X,6(1X,F9.3))
        RETURN
      ELSEIF (DAT(NUM-1,8) - DAT(NUM-1,2) .GT. 0.0) THEN
*  POINT ON RIGHT LARGER: SHORTEN INTERVAL ON RIGHT TO B"
*       A___A'_____B"___B   CENTER SEARCH AROUND A'
        B = B - ALL
        BL = B - A
        V = W
        DAT(NUM,7) = W
        DO 50 I = 8,12
          IF (DAT(NUM,7) .EQ. W) THEN
```

```
            DAT(NUM,I) = DAT(NUM-1,I-6)
          ELSE
            DAT(NUM,I) = DAT(NUM-1,I)
          ENDIF
50       CONTINUE
        ALL = FIB(IK)*BL/FIB(JJ)
        W = A + ALL
        DAT(NUM,1) = W
        X(10) = DAT(NUM,1)
*        X(8) = DAT(NUM,1)
        JPOS = 1
        WRITE(NO,*) 'END SUB INTER: POINT HISTORY: NUM=',NUM
        DO 520 I = 1,NUM
          WRITE(NO,525) (DAT(I,IJ), IJ=1,6)
          WRITE(NO,524) (DAT(I,IJ), IJ=7,12)
520       CONTINUE
524       FORMAT(10X,6(1X,F9.3))
525       FORMAT(1X,6(1X,F9.3))
        RETURN
      ENDIF
      RETURN
      END

*---------------
      SUBROUTINE FIN COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNO-
W,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ COMMON/ONCI/-
A,B,IK,JJ,KK,T,U,V,W,FIB(50),NUM,DAT(100,12),
JPOS,ALL COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*      PURPOSE OF THIS SUBROUTINE IS TO DO THE
*  CALC OF THE FINAL RANGE OF THE DEPENDENT VARIABLE
      DO 20 I = 1,NUM
        WRITE(NO,25) (DAT(I,IJ), IJ=1,6)
        WRITE(NO,24) (DAT(I,IJ), IJ=7,12)
20     CONTINUE
24     FORMAT(10X,6(1X,F9.3))
25     FORMAT(1X,6(1X,F9.3))
      IF (JPOS .LT. 3) THEN
*       SELECT A POINT EPSILON AWAY FROM BEST POINT
        EPS = 0.001*W
        DAL = W + EPS
        DAT(NUM,1) = DAL
        X(10) = DAT(NUM,1)
*        X(8) = DAT(NUM,1)
*       INSERT INFORMATION ABOUT PREVIOUS POINT INTO DAT
        DO 50 I = 7,12
          DAT(NUM,I) = DAT(NUM-1,I)
50       CONTINUE
        JPOS = 3
```

```
        WRITE(NO,*) 'END SUB FIN: JPOS =',JPOS
        RETURN
      ELSEIF (JPOS .GT. 3) THEN
*       WRITE(NO,*) 'DEL =',DEL
        ACC = (W - A)/DEL
      ENDIF
      WRITE(NO,*) 'THE FINAL FEASIBLE REGION'
      PRINT *,'THE FINAL FEASIBLE REGION'
      WRITE(NO,72) DAT(NUM,1), DAT(NUM,7)
72    FORMAT(1X,2(2X,F13.4))
      WRITE(NO,*) 'WITH FUNCTION VALUES'
      WRITE(NO,72) DAT(NUM,2), DAT(NUM,8)
      WRITE(NO,*) 'LOWER CONFIDENCE INTERVAL IS:'
      HALF = DAT(NUM,5)*DAT(NUM,6)
      UP = DAT(NUM,2) + HALF
      DOWN = DAT(NUM,2) - HALF
      WRITE(NO,73) DAT(NUM,3),DAT(NUM,4),UP,DOWN
      WRITE(NO,*) 'UPPER CONFIDENCE INTERVAL IS:'
      HALF = DAT(NUM,11)*DAT(NUM,12)
      UP = DAT(NUM,8) + HALF
      DOWN = DAT(NUM,8) - HALF
      WRITE(NO,73) DAT(NUM,9),DAT(NUM,10),UP,DOWN
73FORMAT(1X,'α=',F9.3,'DF=',F5.1,'ABOV=',F13.4,'BELOW=',F13.4)
      WRITE(NO,85) ACC
85    FORMAT(1X,'THE ACCURACY IS',1X,F9.4)
      WRITE(NO,86) ALPHA
86    FORMAT(1X,'THE REQUIRED ACCURACY IS ',1X,F9.4)
      WRITE(NO,*) 'HISTORY OF INTERVALS: POINT 1 & POINT 2'
      DO 90 I = 1, NUM
        WRITE(NO,100) (DAT(I,IJ),IJ=1,6)
        WRITE(NO,101) (DAT(I,IJ),IJ=7,12)
90    CONTINUE
100   FORMAT(1X,6(1X,F10.3))
101   FORMAT(5X,6(1X,F10.3))
      CALL COUNT(2,1)
      WRITE(NO,*) 'LEAVE SUB FIN: JPOS =',JPOS
      RETURN
      END

*---------------
      SUBROUTINE LAST(AVG1) COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DT-
NOW,TNOW,TFIN,J,NRUN
COMMON/BATDAT/SAMP(200),JBAT,NUMPNT,NO,DEL,NI,NJ,NL,KJ COMMON/ONCI/-
A,B,IK,JJ,KK,T,U,V,W,FIB(50),NUM,DAT(100,12),
JPOS,ALL COMMON/START/ITMAX,ALPHA,BETA,JG,DELTA,IT,IEV1,IEV2,KOUNT,LOW
*     THIS SUB FINDS A POINT EPSILON DIFF FROM BEST POINT
*     AT THE END OF THE ALGORITHM
      PRINT *,'SUBROUTINE LAST'
      NUM = NUM + 1
```

```
      IF ((AVG1 - T) .LE. 0) THEN
        T = AVG1
        JPOS = 4
        DO 31 I = 1,6
          DAT(NUM,I) = DAT(NUM-1,I)
31       CONTINUE
        DAT(NUM,7) = B
        X(10) = B
*        X(8) = B
        RETURN
      ELSEIF ((AVG1 - T) .GT. 0) THEN
        JPOS = 5
        DO 32 I = 7,12
          DAT(NUM,I) = DAT(NUM-1,I-6)
32       CONTINUE
        WRITE(NO,*) 'DAT(1) = A =',A
        DAT(NUM,1) = A
        X(10) = A
*        X(8) = A
        RETURN
      ENDIF
      RETURN
      END
```

APPENDIX C

APPENDIX C

FORTRAN CODE FOR SETTING OF SAMPLING PROCEDURES

```
*
*-------------------------------------------------------
*     SUBROUTINES TO DETERMINE
*        1) INITIALIZATION BIAS:
*        2) BATCH SIZE NECESSARY FOR INDEPENDENCE:
*        3) BATCH SIZE NECESSARY FOR NORMALITY.
*-------------------------------------------------------
*
      PROGRAM JOUT
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON/RECORD/ORR,WHEN,DISP,PLAN
      INTEGER JOB_BEG,MEAN_SZ,LOOP
      REAL TARD_CST,WIP_CST,TRAN_CST,QUE_TME
      DATA TARD_CST,WIP_CST,TRAN_CST,QUE_TME/2.0,0.1,0.0,0.0/
*     LARRY FREDENDALL, MAY 1990, MICHIGAN STATE UNIVERSITY
*     INITIALIZE VARIABLES
      LOOP = 1
3     MEAN_SZ = 100
      JOB_BEG = 1
      NUM_JOB = 0
      NUM_BAT = 0
      N = 0
      DO 5 I = 1,40000
        FLOW(I,1) = 0.0
        FLOW(I,2) = 0.0
        FLOW(I,3) = 0.0
5     CONTINUE
97    FORMAT(2X,F6.2,F6.2,F6.2,F6.2,F6.2,F6.2,F6.2,F6.2)
98    FORMAT(F5.2,F5.2,F5.2,F5.2,F5.2,F5.2,F5.2,F5.2)
99    FORMAT(F13.2,2X,F13.2,2X,F13.2,2X,F12.4,2X,F12.4)
      IF (LOOP .EQ. 1) THEN
OPEN(UNIT =10,FILE='C:\SUMM',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT =11,FILE='C:\REPORT',FORM='FORMATTED',STATUS='NEW')
OPEN(UNIT=12,FILE='C:\DAT1',FORM='FORMATTED',STATUS='OLD')
        REWIND (UNIT=12)
        READ(12,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
        WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
        WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
        DO 100 I = 1,39999
          READ(12,99,END=50) QUE_TME,POOLTME
          IF ( QUE_TME .LT. 0.0) GOTO 50
          FLOW(I,1) = QUE_TME
          FLOW(I,2) = POOLTME
```

```
          FLOW(I,3) = 0
100       CONTINUE
50        CLOSE(UNIT=12)
OPEN(UNIT=12,FILE='C:\COMPILE',FORM='FORMATTED',STATUS='NEW')
          GOTO 150
        ENDIF
        IF (LOOP .EQ. 2) THEN
OPEN(UNIT=13,FILE='C:\DAT2',FORM='FORMATTED',STATUS='OLD')
          REWIND (UNIT=13)
          READ(13,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          DO 101 I = 1,39000
            READ(13,99,END=150) QUE_TME,POOLTME
            IF (QUE_TME .LT. 0.0) GOTO 150
            FLOW(I,1) = QUE_TME
            FLOW(I,2) = POOLTME
            FLOW(I,3) = 0
101       CONTINUE
        ENDIF
        IF (LOOP .EQ. 3) THEN
OPEN(UNIT=14,FILE='C:\DAT3',FORM='FORMATTED',STATUS='OLD')
          REWIND (UNIT=14)
          READ(14,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          DO 102 I = 1,39000
            READ(14,99,END=150) FLW_JOB, QUE_TME,POOLTME
            IF (FLW_JOB .LT. 0.0) GOTO 150
            FLOW(I,1) = FLW_JOB
            FLOW(I,2) = QUE_TME
            FLOW(I,3) = POOLTME
102       CONTINUE
        ENDIF
        IF (LOOP .EQ. 4) THEN
OPEN(UNIT=15,FILE='C:\DAT4',FORM='FORMATTED',STATUS='OLD')
          REWIND (UNIT=15)
          READ(15,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
          DO 103 I = 1,39000
            READ(15,99,END=150) FLW_JOB, QUE_TME,POOLTME
            IF (FLW_JOB .LT. 0.0) GOTO 150
            FLOW(I,1) = FLW_JOB
            FLOW(I,2) = QUE_TME
            FLOW(I,3) = POOLTME
103       CONTINUE
        ENDIF
        IF (LOOP .EQ. 5) THEN
```

```
OPEN(UNIT=16,FILE='C:\DAT5',FORM='FORMATTED',STATUS='OLD')
       REWIND (UNIT=16)
       READ(16,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       DO 104 I = 1,39000
         READ(16,99,END=150) FLW_JOB, QUE_TME,POOLTME
         IF (FLW_JOB .LT. 0.0) GOTO 150
         FLOW(I,1) = FLW_JOB
         FLOW(I,2) = QUE_TME
         FLOW(I,3) = POOLTME
104     CONTINUE
     ENDIF
     IF (LOOP .EQ. 6) THEN
OPEN(UNIT=17,FILE='C:\DAT6',FORM='FORMATTED',STATUS='OLD')
       REWIND (UNIT=17)
       READ(17,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       DO 105 I = 1,39000
         READ(17,99,END=150) FLW_JOB, QUE_TME,POOLTME
         IF (FLW_JOB .LT. 0.0) GOTO 150
         FLOW(I,1) = FLW_JOB
         FLOW(I,2) = QUE_TME
         FLOW(I,3) = POOLTME
105     CONTINUE
     ENDIF
     IF (LOOP .EQ. 7) THEN
OPEN(UNIT=18,FILE='C:\DAT7',FORM='FORMATTED',STATUS='OLD')
       REWIND (UNIT=18)
       READ(18,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
*      PRINT *,'IN LOOP 7'
       DO 106 I = 1,39000
         READ(18,99,END=150) FLW_JOB, QUE_TME,POOLTME
         IF (FLW_JOB .LT. 0.0) GOTO 150
         FLOW(I,1) = FLW_JOB
         FLOW(I,2) = QUE_TME
         FLOW(I,3) = POOLTME
106     CONTINUE
     ENDIF
     IF (LOOP .EQ. 8) THEN
OPEN(UNIT=19,FILE='C:\DAT8',FORM='FORMATTED',STATUS='OLD')
       REWIND (UNIT=19)
       READ(19,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
       DO 107 I = 1,39000
```

```
          READ(19,99,END=150) FLW_JOB, QUE_TME,POOLTME
          IF (FLW_JOB .LT. 0.0) GOTO 150
          FLOW(I,1) = FLW_JOB
          FLOW(I,2) = QUE_TME
          FLOW(I,3) = POOLTME
107       CONTINUE
      ENDIF
      IF (LOOP .EQ. 9) THEN
OPEN(UNIT=20,FILE='C:\DAT9',FORM='FORMATTED',STATUS='OLD')
        REWIND (UNIT=20)
        READ(20,98) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
        WRITE(11,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
        WRITE(10,97) ORR,WHEN,DISP,X6,X7,X9,X10,PLAN
        DO 108 I = 1,39000
          READ(20,99,END=150) FLW_JOB, QUE_TME,POOLTME
          IF (FLW_JOB .LT. 0.0) GOTO 150
          FLOW(I,1) = FLW_JOB
          FLOW(I,2) = QUE_TME
          FLOW(I,3) = POOLTME
108       CONTINUE
      ENDIF
150   NUM_JOB = I-1
110   FORMAT(5X,'THE NUMBER OF JOBS IS:',2X,I10)
      WRITE(11,110) NUM_JOB
      CALL JBATCH(MEAN_SZ,JOB_BEG)
*     FIRST DO INITIALIZATION BIAS ANALYSIS
      JOB_DROP = JOB_BEG
      CALL SCHRUB(MEAN_SZ,JOB_DROP)
      PRINT *,'RETURN FROM SCHRUB'
      INIT_DROP = JOB_DROP
      IF (INIT_DROP .LT. 500) THEN
        JOB_DROP = 500
WRITE(10,*)'500 JOBS TRUNCATED TO ENSURE BIAS ELIMINATED'
WRITE(11,*)'500 JOBS TRUNCATED TO ENSURE BIAS ELIMINATED'
      ENDIF
*     REBATCH TO ACCOUNT FOR TRUNCATED JOBS
      CALL JBATCH(MEAN_SZ,JOB_DROP)
      CALL BOX(MEAN_SZ,JOB_DROP)
      INDEPEN = MEAN_SZ
*
      CALL FILIB(MEAN_SZ,JOB_DROP)
      NORM_BAT = MEAN_SZ
*
      PRINT *,'PREPARE TO CALL MOMENT'
      AVG1 = 0
      AVG2 = 0
      ADEV1 = 0
      SDEV1 = 0
      VAR1 = 0
```

```
      VAR2 = 0
      CURT1 = 0
      A3 = 0
      V3 = 0
CALL MOMENT(AVG1,ADEV1,SDEV1,VAR1,SKEW1,CURT1,AVG2,VAR2,A3,V3)
*
      SKN = 0
      CALL HIST(VAR1,AVG1,MEAN_SZ,SKEW1,SKN,CURT1)
*
112FORMAT('BATCH SIZE =',2X,I10,2X,'TOTAL NU BATCHES=',2X,I10)
      WRITE(11,112) MEAN_SZ, NUM_BAT
113    FORMAT(1X,I8,8X,F12.4,8X,F12.4,8X,F12.4)
WRITE(11,*)'BATCH NUMBER  BATCH MEAN BATCH MEAN BATCH MEAN'
      DO 275 I = 1,NUM_BAT
        WRITE(11,113)I,B_FLOW(I,1),B_FLOW(I,2),B_FLOW(I,3)
275    CONTINUE
*
*THIS SECTION TAKES INFORMATION FROM THE SUBROUTINES AND *PRINT IT TO
A FILE TO BE READ LATER
      INIT = INIT_DROP
      IND = INDEPEN
      NORM = NORM_BAT
WRITE(12,320)ORR,WHEN,DISP,X6,X7,X9,X10,PLAN,INIT,IND,NORM
      WRITE(12,325) VAR1,AVG1,VAR2,AVG2,V3,A3
320    FORMAT(1X,F3.0,F3.0,F3.0,F4.1,F4.1,F4.1,F4.1,F3.0,I6,I6,I6)
325    FORMAT(5X,F13.2,F8.3,F13.2,F8.3,F13.2,F8.3)
      LOOP = LOOP + 1
      IF (LOOP .LE. 9) GOTO 3
      CLOSE (UNIT=10)
      CLOSE(UNIT=11)
      CLOSE(UNIT=12)
*      PRINT *,'FILES ARE CLOSED'
      RETURN
      END
*
```

```
*-----------------------------------------------------
      SUBROUTINE SCHRUB(MEAN_SZ,JOB_DROP)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
      REAL DIFF(1000), RUN_AVG(1000),STUDENT(31)
      INTEGER JOB_CHK
      DATA DIFF/1000*0.0/,RUN_AVG/1000*0.0/
*-----------------------------------------------------
* DETERMINES WHEN INITIALIZATION BIAS IS ENDED.  BASED ON
* SCHRUBEN, SINGH & TIERNEY. IN OPERATIONS RESEARCH
* 31,NO.6(1983)
* PP.1167-1178.  SEE KLEIJNEN, STATISTICAL TOOLS FOR
* SIMULATION PRACTITIONERS, 1987 P.110.
* PROGRAM BY LARRY FREDENDALL, APRIL 1990, MICHIGAN
* STATE UNIVERSITY
* DISCARD FIRST 100 JOBS: Ho: NO INITIAL BIAS: ONESIDED t
*      ALPHA = .05
*-----------------------------------------------------
*
      PRINT *,'START SCHRUBEN'
22FORMAT('USING',1X,I6,1X,'BATCHES OF SIZE',1X,I6,'TO INIT')
      WRITE(11,22)NUM_BAT,MEAN_SZ
23     FORMAT('INITIALLY DROPS',1X,I6,1X,'JOBS.')
      WRITE(11,23)JOB_DROP
      STUDENT(1) = 6.314
      STUDENT(2) = 2.920
      STUDENT(3) = 2.353
      STUDENT(4) = 2.132
      STUDENT(5) = 2.015
      STUDENT(6) = 1.943
      STUDENT(7) = 1.895
      STUDENT(8) = 1.860
      STUDENT(9) = 1.833
      STUDENT(10) = 1.812
      STUDENT(11) = 1.796
      STUDENT(12) = 1.782
      STUDENT(13) = 1.771
      STUDENT(14) = 1.761
      STUDENT(15) = 1.753
      STUDENT(16) = 1.746
      STUDENT(17) = 1.740
      STUDENT(18) = 1.734
      STUDENT(19) = 1.729
      STUDENT(20) = 1.725
      STUDENT(21) = 1.721
      STUDENT(22) = 1.717
      STUDENT(23) = 1.714
      STUDENT(24) = 1.711
      STUDENT(25) = 1.708
```

```
      STUDENT(26) = 1.706
      STUDENT(27) = 1.703
      STUDENT(28) = 1.701
      STUDENT(29) = 1.699
      STUDENT(30) = 1.697
      STUDENT(31) = 1.658
*     USE ONLY LAST HALF OF TOTAL JOBS TO ESTIMATE MEAN & VAR
      JBEG_BAT = INT((NUM_BAT/2))
      JBEG_BAT = JBEG_BAT + 1
      JNUM_BAT = NUM_BAT - JBEG_BAT
*     CALCULATE VARIANCE
      GRAND = 0.0
      M_COUNT = 0
      DO 500 I = JBEG_BAT, NUM_BAT
        GRAND = GRAND + B_FLOW(I,1)
        M_COUNT = M_COUNT + 1
500   CONTINUE
      PRINT *,'NUMBER BATCH=',NUM_BAT,'GRAND TOTAL=',GRAND
      GRAND = GRAND/M_COUNT
      SQUARE = 0.0
      DO 600 I = JBEG_BAT, NUM_BAT
        SQUARE = SQUARE + (B_FLOW(I,1) - GRAND)**2
600   CONTINUE
      VAR = SQUARE/(M_COUNT - 1)
      JDEG_FRE = M_COUNT - 1
      IF (JDEG_FRE .GT. 30) JDEG_FRE = 31
*     RATIONALE FOR JDEG_FRE=31:APROX NORMAL(31)=120
*
*USE THE FIRST 100 BATCHES TO DETERMINE IF INIT COMPLETE
*BY CALC DIFF BETWEEN BATCHES OF SIZE K=1 TO 100 & GRAND MEAN
*FIRST DROP JOBS KNOWN NOT TO BE INITIALIZED
      NEED_BAT = 100
      JBAT_BEG = 1
620   JBAT_END = JBAT_BEG + NEED_BAT - 1
      IF (JBAT_END .GT. NUM_BAT) THEN
        WRITE(11,*)'UNABLE TO FIND TRUNCATION POINT'
631       FORMAT(1X,'USED',2X,I10,'BATCHES: NEED MORE DATA')
*         WRITE(11,631)JBAT_END
        WRITE(10,631)JBAT_END
        RETURN
      ENDIF
* THIS CALCULATES MOVING AVERAGES FOR FIRST BATCHES
* & COMPARES THESE AVERAGES TO AVERAGES FOR LAST BATCHES
* BATCHES ARE BATCHED AGAIN INTO BATCHES K=1,JNUM_BAT
      DO 725 I = 1,JBAT_END
        RUN_AVG(I) = 0.0
725   CONTINUE
      M_COUNT = 0
      DO 800 K = 1,NEED_BAT
```

```
      IF (K .EQ. 1) then
        RUN_AVG(K) = B_FLOW(JBAT_BEG,1)
      ELSE
RUN_AVG(K) = RUN_AVG(K-1) + B_FLOW(JBAT_BEG +K-1,1)
      ENDIF
      M_COUNT = M_COUNT + 1
800   CONTINUE
     PRINT *,'NUMBER OF BATCHES NEEDED =',M_COUNT
601   DO 900 K = 1, M_COUNT
* REDUCE VALUE IN RUN_AVG(K) TO MEAN OF BATCHES INCLUDED
      RUN_AVG(K) = RUN_AVG(K)/K
      DIFF(K) =  RUN_AVG(K) - GRAND
900   CONTINUE
*     CALCULATE t STATISTIC
     TOT_DIFF = 0.0
     DO 950 K = 1, M_COUNT
       RK = REAL(K)/REAL(M_COUNT)
       TOT_DIFF = TOT_DIFF + DIFF(K)*(RK)*(1-RK)
950   CONTINUE
*
     JDEG_FRE = M_COUNT
     IF (JDEG_FRE .GT. 30) JDEG_FRE = 31
     IF (VAR .LE. 0.000001) VAR =.001
     T_HAT = (SQRT(45/(M_COUNT*VAR)))*TOT_DIFF
     T_HAT = ABS(T_HAT)
*
953   FORMAT('T STAT =',1X,F12.4,1X,'CRIT T VALUE=',1X,F12.4)
     WRITE(11,953)T_HAT,STUDENT(JDEG_FRE)
*
     PRINT *,'CHECKING T STATISTIC: LINE 953'
*     CHECK SIGNIFICANCE
     IF (T_HAT .GE. STUDENT(JDEG_FRE)) THEN
*       T_HAT IS SIGNIFICANT: REJECT Ho: TRUNCATE 1 BATCH
       JBAT_BEG = JBAT_BEG + 1
       JOB_DROP = JOB_DROP + MEAN_SZ
951      FORMAT(1X,'INIT BIAS SIGNIFICANT,DROP:',2X,I10)
       WRITE(11,951)JOB_DROP
       GOTO 620
     ELSE
*       ACCEPT Ho: INITIALIZATION BIAS ENDED AT JOB_STRT
970      FORMAT(1X,'INITIALIZATION BIAS ENDS AT JOB:',2X,I10)
*        WRITE(11,970) JOB_DROP
       WRITE(10,970) JOB_DROP
     ENDIF
     PRINT *, 'END OF SUBROUTINE FIL'
     RETURN
     END
```

```
      SUBROUTINE JBATCH(MEAN_SZ,JOB_BEG)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
*-------------------------------------------------------
*     LARRY FREDENDALL, APRIL, 1990, MICHIGAN STATE
*     UNIVERSITY
*     BATCH JOBS TO EITHER TRUNCATE JOBS OR INCREASE
*     BATCHSIZE
*-------------------------------------------------------
*      PRINT *,'BEGIN JBATCH: NUM_JOB =',NUM_JOB
      DO 10 I = 1,1000
        B_FLOW(I,1) = 0.0
        B_FLOW(I,2) = 0.0
        B_FLOW(I,3) = 0.0
10    CONTINUE
*     THIS SUBROUTINE BATCHES JOBS THAT ARE NOT TRUNCATED
*     CALCULATE BATCH MEANS: MEAN COST PER JOB
      I = 1
      JOB_STRT = JOB_BEG
160   JOB_END = JOB_STRT + MEAN_SZ
      MCOUNT = 0
      DO 250 J = JOB_STRT+1, JOB_END
        B_FLOW(I,1) = B_FLOW(I,1) + FLOW(J,1)
        B_FLOW(I,2) = B_FLOW(I,2) + FLOW(J,2)
        B_FLOW(I,3) = B_FLOW(I,3) + FLOW(J,3)
        MCOUNT = MCOUNT + 1
250   CONTINUE
      B_FLOW(I,1) = B_FLOW(I,1)/MCOUNT
      B_FLOW(I,2) = B_FLOW(I,2)/MCOUNT
      B_FLOW(I,3) = B_FLOW(I,3)/MCOUNT
      IF ((NUM_JOB - JOB_END) .GT. MEAN_SZ -1) THEN
        JOB_STRT = JOB_END
        I = I + 1
*        PRINT *,'STARTING ANOTHER BATCH: I =',I
        GOTO 160
      ENDIF
      NUM_BAT = I
      PRINT *,'NUMBER OF JOBS =',NUM_JOB
      PRINT *,'NUMBER JOBS DROPPED =',JOB_BEG
      PRINT *,'END OF JBATCH, NUMBER OF BATCHES =',NUM_BAT
      PRINT *,'BATCH SIZE =',MEAN_SZ
      PRINT *, 'END OF SUBROUTINE JBATCH'
      RETURN
      END
```

```
      SUBROUTINE BOX(MEAN_SZ,JOB_DROP)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
      REAL AUTO(50),CHI_SQ(100)
      DATA AUTO/50*0.0/,CHI_SQ/100*0.0/
*
*-----------------------------------------------------
*     LARRY FREDENDALL, APRIL, 1990, MICHIGAN STATE
*     UNIVERSITY
*     DETERMINE INDEPEDENCE OF BATCH MEANS:  REQUIRES AT
*     LEAST
*     100 BATCHES (KLEIJNEN, 1987, P.67). CALC BOX-PIERCE Q
*     FOR 20 LAGS AND COMPARES Q TO CRITICAL CHI SQUARE.  SEE
*     MAKADAKRIS, WHEELWRIGHT AND MCGEE "FORECASTING: METHODS
*     & APPLICATIONS" 1983, P.390.  SUGGESTED BY PAUL RUBIN.
*     DOUBLE BATCH SIZE UNTIL BOX-PIERCE Q NOT SIGNIFICANT,
*     NO ADDITIONAL DATA AVAILABLE.
*-----------------------------------------------------
*
*      PRINT *,'PRINT START BOX PIERCE SUBROUTINE'
17    FORMAT('STARTING BOX ROUTINE, BATCH SIZE =',1X,I5)
      WRITE(11,17) MEAN_SZ
*     INITIALIZE CHI_SQ TABLE: ALPHA = .05
      CHI_SQ(1) = 3.84
      CHI_SQ(2) = 5.99
      CHI_SQ(3) = 7.81
      CHI_SQ(4) = 9.49
      CHI_SQ(5) = 11.07
      CHI_SQ(6) = 12.59
      CHI_SQ(7) = 14.07
      CHI_SQ(8) = 15.51
      CHI_SQ(9) = 16.92
      CHI_SQ(10) = 18.31
      CHI_SQ(11) = 19.68
      CHI_SQ(12) = 21.03
      CHI_SQ(13) = 22.36
      CHI_SQ(14) = 23.68
      CHI_SQ(15) = 25.00
      CHI_SQ(16) = 26.30
      CHI_SQ(17) = 27.59
      CHI_SQ(18) = 28.87
      CHI_SQ(19) = 30.14
      CHI_SQ(20) = 31.41
      CHI_SQ(21) = 32.67
      CHI_SQ(22) = 33.92
      CHI_SQ(23) = 35.17
      CHI_SQ(24) = 36.42
      CHI_SQ(25) = 37.65
      CHI_SQ(26) = 38.89
```

```
      CHI_SQ(27) = 40.11
      CHI_SQ(28) = 41.34
      CHI_SQ(29) = 42.56
      CHI_SQ(30) = 43.77
      CHI_SQ(40) = 55.76
      CHI_SQ(50) = 67.50
      CHI_SQ(60) = 79.08
      CHI_SQ(70) = 90.53
      CHI_SQ(80) = 101.9
      CHI_SQ(90) = 113.1
      CHI_SQ(100) =124.3
31     GRAND1 = 0.0
      DO 650 L = 1,NUM_BAT
*       CALCULATE GRAND MEAN
        GRAND1 = GRAND1 + B_FLOW(L,1)
650    CONTINUE
      GRAND1 = GRAND1/NUM_BAT
*     CALCULATE AUTOCORRELATION FOR EACH LAG
      DO 700 J = 1, 20
        AUTO(J) = 0.0
        SQ_DIFF = 0.0
        DO 725 K = 1, NUM_BAT - J
AUTO(J)=AUTO(J)+(B_FLOW(K,1)-GRAND1)*(B_FLOW(K+J,1)-GRAND1)
          SQ_DIFF = SQ_DIFF + (B_FLOW(K,1) - GRAND1)**2
725       CONTINUE
        AUTO(J) = (AUTO(J)/SQ_DIFF)
700    CONTINUE
*     COMPUTE BOX-PIERCE Q & CHECK SIGNIF. (CHI SQUARE)
*     DEG OF FREEDOM = NUM_BAT: ALPHA = .05
*     Q = (SAMPLE SIZE)*(SUM(AUTOCORRELATION TERMS SQUARED))
      TOT_AUTO = 0.0
      DO 800 I = 1,20
        TOT_AUTO = TOT_AUTO + (AUTO(I))**2
800    CONTINUE
      BOX_PQ = NUM_BAT*TOT_AUTO
      IF (NUM_BAT .GT. 100) THEN
        CHI = CHI_SQ(100)
      ELSE
        IF (NUM_BAT .GT. 30) THEN
          K = INT(NUM_BAT/10)
          K = K*10
          CHI = CHI_SQ(K)
        ELSE
          CHI = CHI_SQ(NUM_BAT)
        ENDIF
      ENDIF
1000   IF (BOX_PQ .LE. CHI) THEN
*       BATCH SIZE OK
722    FORMAT('BATCH SIZE=',I9,' IS OK: BOX-Q=',F12.4,'CRIT CHI=',F12.4)
```

```
      WRITE(11,722) MEAN_SZ,BOX_PQ,CHI
      WRITE(10,722) MEAN_SZ,BOX_PQ,CHI
    ELSE
*     BATCH SIZE TOO SMALL
727   FORMAT('BATCH SIZE=',I9,'NOT OK: BOX-Q=',F12.4,'CRIT CHI=',F12.4)
      WRITE(11,727) MEAN_SZ,BOX_PQ,CHI
      MEAN_SZ = 2*MEAN_SZ
      CALL JBATCH(MEAN_SZ,JOB_DROP)
      IF (NUM_BAT .LT. 20) THEN
728   FORMAT('INSUFFICIENT DATA NEED TO INCREASE BATCH SIZE TO',1X,I6)
        WRITE(11,728)MEAN_SZ
        WRITE(10,728)MEAN_SZ
729   FORMAT('NEED 100 BATCH, HAVE ONLY',1X,I6,1X,'BATCH')
        WRITE(11,729)NUM_BAT
        WRITE(10,729)NUM_BAT
        GOTO 909
      ENDIF
      GOTO 31
909   ENDIF
    RETURN
    END
```

```
      SUBROUTINE FILIB(MEAN_SZ,JOB_DROP)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
      LOGICAL SORT
      INTEGER LOOP
      REAL BAT_SORT(500,3),COR_COEF(100)
      DATA BAT_SORT/1500*0.0/,COR_COEF/100*0.0/
*-----------------------------------------
*     TEST OF NORMALITY BASED ON RYAN,JOINER
*     & RYAN, MINITAB (1982, P.47-49)
*     & FILIBEN (1975) TECHNOMETRICS
*     Ho IS THAT THE MEANS ARE NORMAL.  DOUBLES BATCH
*     SIZE UNTIL NORMALITY ACHIEVED.  TECHNIQUE SUGGESTED BY
*     PAUL RUBIN.
*     PROGRAMER LARRY FREDENDALL, APRIL 1990, MSU
*     REJECT Ho IF CORR LT CRITICAL VALUE
*-----------------------------------------------
*
      PRINT *,'START FILIBEN SUBROUTINE'
9 FORMAT('INITIAL BATCH SIZE TO DETERMINE NORMALITY =',1X,I6)
      WRITE(11,9)MEAN_SZ
      WRITE(10,9)MEAN_SZ
*     NORMAL PROBABILITY PLOT CORRELATION FOR P = .25
      COR_COEF(3) = .924
      COR_COEF(4) = .931
      COR_COEF(5) = .935
      COR_COEF(6) = .940
      COR_COEF(7) = .944
      COR_COEF(8) = .948
      COR_COEF(9) = .951
      COR_COEF(10) = .954
      COR_COEF(11) = .957
      COR_COEF(12) = .959
      COR_COEF(13) = .962
      COR_COEF(14) = .964
      COR_COEF(15) = .965
      COR_COEF(16) = .967
      COR_COEF(17) = .968
      COR_COEF(18) = .969
      COR_COEF(19) = .971
      COR_COEF(20) = .972
      COR_COEF(21) = .973
      COR_COEF(22) = .974
      COR_COEF(23) = .975
      COR_COEF(24) = .975
      COR_COEF(25) = .976
      COR_COEF(26) = .977
      COR_COEF(27) = .977
      COR_COEF(28) = .978
```

```
      COR_COEF(29) = .979
      COR_COEF(30) = .979
      COR_COEF(31) = .980
      COR_COEF(32) = .980
      COR_COEF(33) = .981
      COR_COEF(34) = .981
      COR_COEF(35) = .982
      COR_COEF(36) = .982
      COR_COEF(37) = .982
      COR_COEF(38) = .983
      COR_COEF(39) = .983
      COR_COEF(40) = .983
      COR_COEF(41) = .984
      COR_COEF(42) = .984
      COR_COEF(43) = .984
      COR_COEF(44) = .984
      COR_COEF(45) = .985
      COR_COEF(46) = .985
      COR_COEF(47) = .985
      COR_COEF(48) = .985
      COR_COEF(49) = .986
      COR_COEF(50) = .986
      COR_COEF(55) = .987
      COR_COEF(60) = .988
      COR_COEF(65) = .989
      COR_COEF(70) = .989
      COR_COEF(75) = .990
      COR_COEF(80) = .991
      COR_COEF(85) = .991
      COR_COEF(90) = .991
      COR_COEF(95) = .992
      COR_COEF(100) = .992
25 FORMAT('USING',I9,2X,'BATCHES: CRITICAL CORR COEF='F12.4)
30     IF (NUM_BAT .GT. 100) THEN
         NUM = 100
       ELSE
         NUM = NUM_BAT
         J_REM = MOD(NUM,5)
         NUM = NUM - J_REM
       ENDIF
       WRITE(11,25)NUM,COR_COEF(NUM)
*      SORT DATA TO DETERMINE MEDIAN: FIRST STORE DATA
       DO 205 I = 1, NUM
         BAT_SORT(I,1) = B_FLOW(I,1)
         BAT_SORT(I,2) = REAL(I)
205    CONTINUE
       SORT = .FALSE.
210    IF (.NOT. SORT) THEN
         SORT = .TRUE.
```

```
      DO 250 I = 1,NUM -1
        IF (BAT_SORT(I,1) .GT. BAT_SORT(I+1,1)) THEN
          TEMP_ONE = BAT_SORT(I,1)
          TEMP_TWO = BAT_SORT(I,2)
          BAT_SORT(I,1) = BAT_SORT(I+1,1)
          BAT_SORT(I,2) = BAT_SORT(I+1,2)
          BAT_SORT(I+1,1) = TEMP_ONE
          BAT_SORT(I+1,2) = TEMP_TWO
          SORT = .FALSE.
        ENDIF
250     CONTINUE
      GOTO 210
    ENDIF
*   CALC UNIFORM ORDER STATISTIC MEDIANS (FILIBEN, P.116)
    EXPOWER = (1.0/REAL(NUM_BAT))
    BAT_SORT(NUM,2) = (.5**EXPOWER)
    BAT_SORT(1,2) = 1 - BAT_SORT(NUM,2)
    DO 10 I = 2, NUM - 1
      BAT_SORT(I,2) = (I - .3175)/(REAL(NUM) + .365)
10   CONTINUE
*   CALCULATE NORMAL ORDER STATISTIC MEDIANS
266  FORMAT(1X,F12.4,5X,F12.4)
    DO 300 I = 1, NUM
      PHI = BAT_SORT(I,2)**.14
      THETA = (1 - BAT_SORT(I,2))**.14
      BAT_SORT(I,3) = (4.91)*(PHI - THETA)
300  CONTINUE
*   CALCULATE CORRELATION: USE FILLIBEN'S FORMULA P.112
    RMEAN = 0.0
    DO 100, I = 1, NUM
      RMEAN = RMEAN + BAT_SORT(I,1)
100  CONTINUE
    RMEAN = RMEAN/(REAL(NUM))
    TOTAL = 0.0
    SQ_M = 0.0
    DO 450 I = INT(NUM/2) +1, NUM
TOTAL=TOTAL+BAT_SORT(I,3)*(BAT_SORT(I,1)-BAT_SORT(NUM-I+1,1))
      SQ_M = SQ_M + BAT_SORT(I,3)*BAT_SORT(I,3)
450  CONTINUE
    SQ_M = 2*SQ_M
    SQ_DIFF = 0.0
    DO 480 I = 1,NUM
      SQ_DIFF = SQ_DIFF + (BAT_SORT(I,1) - RMEAN)**2
480  CONTINUE
    COR = TOTAL/(SQRT((SQ_M*SQ_DIFF)))
    PRINT *,'FILIBEN, CRITICAL CORRELATION =',COR_COEF(NUM)
    PRINT *,'ACTUAL CORRELATION =',COR
    IF (COR .GE. COR_COEF(NUM)) THEN
*     SCORES ARE NORMAL
```

```
437FORMAT('NORMAL: CORRELATION OF MEANS & NSCORES =',1X,F12.4)
      WRITE(11,437)COR
      WRITE(10,437)COR
438    FORMAT('BATCH SIZE REQUIRED FOR NORMALITY =',1X,I5)
      WRITE(11,438)MEAN_SZ
      WRITE(10,438)MEAN_SZ
    ELSE
*     SCORES ARE NOT NORMAL
440    FORMAT('NOT NORMAL: REJECT Ho: CRIT COEF=',1X,F12.4)
      WRITE(11,440)COR_COEF(NUM)
439    FORMAT('CUR BAT_SZ=',1X,I5,1X,'ACT cORR=',1X,F12.4)
      WRITE(11,439)MEAN_SZ,COR
*     INCREASE SIZE OF EACH BATCH
      MEAN_OLD = MEAN_SZ
      IF (MEAN_SZ .LT. 1000) THEN
        MEAN_SZ = 2*MEAN_SZ
      ELSE
        IF (MEAN_SZ .LT. 3000) THEN
          MEAN_SZ = INT(1.5*MEAN_SZ)
        ELSE
          MEAN_SZ = MEAN_SZ + 1000
        ENDIF
      ENDIF
      CALL JBATCH(MEAN_SZ, JOB_DROP)
      IF (NUM_BAT .LT. 5) THEN
491      FORMAT('INSUFFICIENT SAMPLE SIZE: BATCH=',2X,I9)
        WRITE(11,491)MEAN_OLD
        WRITE(10,491)MEAN_OLD
        RETURN
      ELSE
        GOTO 30
      ENDIF
    ENDIF
    RETURN
    END
```

```
      SUBROUTINE HIST(V,AVG,MEAN_SZ,SK,SKN,CURT)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
*-------------------------------------------------
*     PLOTS A HISTOGRAM: VERTICAL SCALE: 1 LINE = 5%
*     PROGRAM BASED ON BRAINEAD, GOLDBERG & GROSS IN
*     FORTRAN 77 PROGRAMMING, HARPER AND ROW, NEW YORK,
*     1978, P.237
*-------------------------------------------------
      REAL CLAS_M(20),EDGE(20,2)
INTEGER JROW, JCOL,CLASS(55),NUM_CLAS,MAX,MIN,MEMBER(20)
      INTEGER MARG,JCUT
      CHARACTER*1 HGRAM(30,80)
      LOGICAL STR
      DATA NUM_CLAS/0/,CLAS_M/20*0.0/,EDGE/40*0.0/
      DATA PERCENT/0.0/,MARG/20/,JCUT/1/
*
*      PRINT *,'START HIST: NUMBER OF BATCHES =',NUM_BAT
*     TOTAL NUMBER OF BATCHES IS NUM_BAT
*     NUM OF CLASSES TO DIVIDE THE DATA IS BETWEEN 6 AND 15
*     FEWER THAN 10 BATCHES IS DIVIDED INTO 6 CLASSES:
*       AN ADDITIONAL
*     CLASS IS ADDED FOR EVERY 5 BATCHES IN DATA EX. 15 BATCH
*     DIVIDED INTO 7 CLASSES: 55 BATCH DIVIDED 15 CLASSES
      IF (NUM_BAT .LT. 5) THEN
        WRITE(10,4)NUM_BAT
4FORMAT('TOO FEW BATCHES FOR HISTOGRAM: NUMBER=',2X,I6)
3       FORMAT('CURRENT BATCH SIZE =',2X,I6)
        WRITE(10,3) MEAN_SZ
        WRITE(10,*)'          ++++++++++      '
        RETURN
      ENDIF
      CLASS(1) = 6
      CLASS(2) = 6
      CLASS(3) = 6
      CLASS(4) = 6
      CLASS(5) = 6
      CLASS(6) = 6
      CLASS(7) = 6
      CLASS(8) = 6
      CLASS(9) = 6
      CLASS(10) = 7
      CLASS(11) = 7
      CLASS(12) = 7
      CLASS(13) = 7
      CLASS(14) = 7
      CLASS(15) = 8
      CLASS(16) = 8
      CLASS(17) = 8
```

```
      CLASS(18) = 8
      CLASS(19) = 8
      CLASS(20) = 9
      CLASS(21) = 9
      CLASS(22) = 9
      CLASS(23) = 9
      CLASS(24) = 9
      CLASS(25) = 9
      CLASS(26) = 10
      CLASS(27) = 10
      CLASS(28) = 10
      CLASS(29) = 10
      CLASS(30) = 10
      CLASS(31) = 11
      CLASS(32) = 11
      CLASS(33) = 11
      CLASS(34) = 11
      CLASS(35) = 11
      CLASS(36) = 12
      CLASS(37) = 12
      CLASS(38) = 12
      CLASS(39) = 12
      CLASS(40) = 12
      CLASS(41) = 13
      CLASS(42) = 13
      CLASS(43) = 13
      CLASS(44) = 13
      CLASS(45) = 13
      CLASS(46) = 14
      CLASS(47) = 14
      CLASS(48) = 14
      CLASS(49) = 14
      CLASS(50) = 14
      CLASS(51) = 15
      CLASS(52) = 15
      CLASS(53) = 15
      CLASS(54) = 15
      CLASS(55) = 15
      IF (NUM_BAT .GT. 55) THEN
        NUM_CLAS = 16
      ELSE
        NUM_CLAS = CLASS(NUM_BAT)
      ENDIF
5     DO 10,JCOL = 1,80
        DO 11, JROW = 1,30
          HGRAM(JROW,JCOL) = ' '
11      CONTINUE
10    CONTINUE
*     FIND MAXIMUM AND MINIMUM BATCH MEAN
```

```
      MAX = 1
      MIN = 1
      DO 50 I = 1,NUM_BAT
        IF (B_FLOW(I,1) .GT. B_FLOW(MAX,1)) MAX = I
        IF (B_FLOW(I,1) .LT. B_FLOW(MIN,1)) MIN = I
50     CONTINUE
*      CALCULATE WIDTH OF CLASS INTERVALS
WIDTH = (INT(B_FLOW(MAX,1) + 1.0) - B_FLOW(MIN,1))/NUM_CLAS
*      DETERMINE MIDDLE OF EACH CLASS
      CLAS_M(1) = (2*B_FLOW(MIN,1) + WIDTH)/2
      DO 75 I = 2,NUM_CLAS
        CLAS_M(I) = CLAS_M(I-1) + WIDTH
75     CONTINUE
*      DETERMINE BEGINNING AND ENDING OF EACH CLASS
      EDGE(1,1) = B_FLOW(MIN,1)
      DO 100 I = 2,NUM_CLAS
        EDGE(I-1,2) = EDGE(I-1,1) + WIDTH
        EDGE(I,1) = EDGE(I-1,2)
100    CONTINUE
      EDGE(NUM_CLAS,2) = EDGE(NUM_CLAS,1) + WIDTH
*      COUNT NUMBER OF BATCH MEANS IN EACH CLASS
      DO 120 I = 1, NUM_CLAS
        MEMBER(I) = 0
        DO 140 K = 1,NUM_BAT
IF((B_FLOW(K,1).GE.EDGE(I,1)).AND.(B_FLOW(K,1).LT.EDGE(I,2)))THEN
MEMBER(I) = MEMBER(I) +1
ENDIF
140       CONTINUE
        IF ((MEMBER(I) .EQ. 0).AND.(JCUT .LE. 3)) THEN
*          REDUCE NUMBER OF CLASSES
          NUM_CLAS = NUM_CLAS - 1
          JCUT = JCUT + 1
          GOTO 5
        ENDIF
120    CONTINUE
*      CALCULATE PERCENT OF BATCH MEANS IN EACH CLASS
*       PRINT *,'NUMBER OF BATCHES =',NUM_BAT
174    DO 175 I = 1, NUM_CLAS
*       PRINT *,'MEMBER(I) =',MEMBER(I)
        PERCENT = (100*MEMBER(I)/NUM_BAT)
*       PRINT *,'PERCENT =',PERCENT
IF((PERCENT.GT.0.0).AND.(PERCENT.LT.1.0))PERCENT =1.0
*       PRINT *,'PERCENT =',PERCENT
        LENGTH = NINT(PERCENT/1.0)
*       PRINT *,'LENGTH =',LENGTH
*       PRINT *,'MARG =',MARG
        DO 200 JCOL = MARG,MARG+LENGTH
          HGRAM(I,JCOL) = '*'
200       CONTINUE
```

```
*      PRINT *,'STOPPING VALUE OF JCOL=',JCOL
175    CONTINUE
*
      WRITE(10,*)' '
      WRITE(10,*)'            HISTOGRAM OF BATCH MEANS'
      WRITE(10,301)ORR,WHEN
301    FORMAT('ORR RULE =',1X,F5.0,1X,'WHEN RULE =',1X,F5.0)
      WRITE(10,302)DISP,PLAN
302    FORMAT('DISP RULE =',1X,F5.0,1X,'PLAN SYS =',1X,F5.0)
WRITE(10,*)'-------------------------------------------------0    FORMAT(10X,'BATCH SIZE
=',1X,I5,5X,'NUMBER OF BATCHE
,1X,I5)       WRITE(10,300)MEAN_SZ ,NUM_BAT
      WRITE(1
0,321)V,AVG
321    FORMAT(10X,'VARIANCE =',1X,F8.3,1X,'AVERAGE =',1X,F8.3)
      WRITE(10,303)SK,SKN
303    FORMAT('SKEWNESS =',1X,F12.4,1X,'EXPECTED SKEW=',1X,F12.4)
      WRITE(10,304)CURT
304    FORMAT('KURTOSIS =',1X,F12.4)
      DO 310 I = 1,2
        WRITE(10,*)' '
310    CONTINUE
      WRITE(10,*)'MIDPOINT   COUNT'
      DO 325 I = 1,NUM_CLAS
        IF (MEMBER(I) .EQ. 0) THEN
          WRITE(10,354)CLAS_M(I),MEMBER(I)
          GOTO 324
        ENDIF
324       N = I
325    CONTINUE
352    FORMAT(F8.2,1X,I8,4X,55(A))
354    FORMAT(F8.2,1X,I8)
      WRITE(10,*)'HORIZONTAL SCALE:  1 ASTERISK = 1%'
      print *,'end of histogram'
      RETURN
      END
```

```
      SUBROUTINE MOMENT(A1,ADEV1,SDEV1,V1,SKEW1,CURT1,A2,V2,A3,V3)
      COMMON/BATCH/NUM_BAT,B_FLOW(1000,3),NUM_JOB
      COMMON /INITL/TAR,WIP,TR,FLOW(40000,3)
      REAL AVG1,AVG2,AVG3,VAR1,VAR2,VAR3
DATA AVG1,AVG2,AVG3,VAR1,VAR2,VAR3/0.0,0.0,0.0,0.0,0.0,0.0/
*
*     FROM NUMERICAL RECIPES: THE ART OF SCIENTIFIC COMPUTING
*     PRESS, FLANNERY, TEUKOLSKY, & VETTERING, P.458
*     CAMBRIDGE UNIVERSITY PRESS, 1986
*     GIVEN AN ARRAY, IT REYTURNS ITS MEAN, AVERAGE
*       DEVIATION, STANDARD
*     DEVIATION, VARIANCE, SKEWNESS AND KURTOSIS
*
*     FIRST PASS TO OBTAIN THE MEAN: NEED A MINIMUM SIZE *      SAMPLE
* PRINT *,'SUBROUTINE MOMENT: NUMBER OF BATCHES =',NUM_BAT
      IF (NUM_BAT .LT. 2) RETURN
      DO 10 I = 1,NUM_BAT
        AVG1 = AVG1 + B_FLOW(I,1)
        AVG2 = AVG2 + B_FLOW(I,2)
        AVG3 = AVG3 + B_FLOW(I,3)
*         AVG4 = AVG4 + BAT_JOB(I,4)
*         AVG5 = AVG5 + BAT_JOB(I,5)
*         AVG6 = AVG6 + BAT_JOB(I,6)
*         AVG7 = AVG7 + BAT_JOB(I,7)
10     CONTINUE
      AVG1 = AVG1/NUM_BAT
      AVG2 = AVG2/NUM_BAT
      AVG3 = AVG3/NUM_BAT
*      AVG4 = AVG4/NUM_BAT
*      AVG5 = AVG5/NUM_BAT
*      AVG6 = AVG6/NUM_BAT
*      AVG7 = AVG7/NUM_BAT
* SECOND PASS TO GET FIRST, SECOND, THIRD AND FOURTH MOMENTS
*     OF DEVIATION FROM THE MEAN
      DIFF = 0.0
      SQ = 0.0
      DO 20 I = 1, NUM_BAT
        DIFF = B_FLOW(I,1) - AVG1
        ADEV1 = ADEV1 + ABS(DIFF)
        SQ = DIFF*DIFF
        VAR1 = VAR1 + SQ
        SQ = SQ*DIFF
        SKEW1 = SKEW1 + SQ
        SQ = SQ*DIFF
        CURT1 = CURT1 + SQ
        DIFF = B_FLOW(I,2) - AVG2
        ADEV2 = ADEV2 + ABS(DIFF)
        SQ= DIFF*DIFF
        VAR2 = VAR2 + SQ
```

```
        SQ= SQ*DIFF
        SKEW2 = SKEW2 + SQ
        SQ= SQ*DIFF
        CURT2 = CURT2 + SQ
        DIFF= B_FLOW(I,3) - AVG3
        ADEV3 = ADEV3 + ABS(DIFF)
        SQ= DIFF*DIFF
        VAR3 = VAR3 + SQ
        SQ= SQ*DIFF
        SKEW3 = SKEW3 + SQ
        SQ= SQ*DIFF
        CURT3 = CURT3 + SQ
*        DIFF= BAT_JOB(I,4) - AVG4
*        ADEV4 = ADEV4 + ABS(DIFF)
*        SQ= DIFF*DIFF
*        VAR4 = VAR4 + SQ
*        SQ= SQ*DIFF
*        SKEW4 = SKEW4 + SQ
*        SQ= SQ*DIFF
*        CURT4 = CURT4 + SQ
*        DIFF= BAT_JOB(I,5) - AVG5
*        ADEV5 = ADEV5 + ABS(DIFF)
*        SQ= DIFF*DIFF
*        VAR5 = VAR5 + SQ
*        SQ= SQ*DIFF
*        SKEW5 = SKEW5 + SQ
*        SQ= SQ*DIFF
*        CURT6 = CURT6 + SQ
*        DIFF= BAT_JOB(I,7) - AVG7
*        ADEV7 = ADEV7 + ABS(DIFF)
*        SQ= DIFF*DIFF
*        VAR7 = VAR7 + SQ
*        SQ= SQ*DIFF
*        SKEW7 = SKEW7 + SQ
*        SQ= SQ*DIFF
*        CURT7 = CURT7 + SQ
20    CONTINUE
* PUT THE PIECES TOGETHER ACCORDING TO THE CONVENTIONAL DEFN.
     ADEV1 = ADEV1/NUM_BAT
     ADEV2 = ADEV2/NUM_BAT
     ADEV3 = ADEV3/NUM_BAT
*     ADEV4 = ADEV4/NUM_BAT
*     ADEV5 = ADEV5/NUM_BAT
*     ADEV6 = ADEV6/NUM_BAT
*     ADEV7 = ADEV7/NUM_BAT
     VAR1 = VAR1/(NUM_BAT-1)
     VAR2 = VAR2/(NUM_BAT-1)
     VAR3 = VAR3/(NUM_BAT-1)
*     VAR4 = VAR4/(NUM_BAT-1)
```

```
*      VAR5 = VAR5/(NUM_BAT-1)
*      VAR6 = VAR6/(NUM_BAT-1)
*      VAR7 = VAR7/(NUM_BAT-1)
      SDEV1 = SQRT(VAR1)
      SDEV2 = SQRT(VAR2)
      SDEV3 = SQRT(VAR3)
*      SDEV4 = SQRT(VAR4)
*      SDEV5 = SQRT(VAR5)
*      SDEV6 = SQRT(VAR6)
*      SDEV7 = SQRT(VAR7)
*      PRINT *,'VARIANCE ONE =',VAR1,'   AVERAGE ONE=',AVG1
* PRINT *,'VARIANCE TWO =',VAR2,'     AVERAGE TWO =',AVG2
*      PRINT *,'VARIANCE THREE =',VAR3,'   AVERAGE 3 =',AVG3
      IF (VAR1 .GT. 0.1) THEN
        SKEW1 = SKEW1/(NUM_BAT*(SDEV1**3))
        CURT1 = CURT1/(NUM_BAT*(VAR1**2)) - 3.0
*      PRINT *,'KURTOSIS 1 =',CURT1
      ENDIF
      IF (VAR2 .GT. 0.1) THEN
        SKEW2 = SKEW2/(NUM_BAT*(SDEV2**3))
        CURT2 = CURT2/(NUM_BAT*(VAR2**2)) - 3.0
*      PRINT *,'KURTOSIS 2 =',CURT2
      ENDIF
      IF (VAR3 .GT. 0.1) THEN
        SKEW3 = SKEW3/(NUM_BAT*(SDEV3**3))
        CURT3 = CURT3/(NUM_BAT*(VAR3**2)) - 3.0
*      PRINT *,'KURTOSIS 3 =',CURT3
      ENDIF
*      IF (VAR4 .NE. 0.0) THEN
*        SKEW4 = SKEW4/(NUM_BAT*(SDEV4**3))
*        CURT4 = CURT4/(NUM_BAT*(VAR4**2)) - 3.0
*      ELSE
*        GOTO 200
*      ENDIF
*      IF (VAR5 .NE. 0.0) THEN
*        SKEW5 = SKEW5/(NUM_BAT*(SDEV5**3))
*        CURT5 = CURT5/(NUM_BAT*(VAR5**2)) - 3.0
*      ELSE
*        GOTO 200
*      ENDIF
*      IF (VAR6 .NE. 0.0) THEN
*        SKEW6 = SKEW6/(NUM_BAT*(SDEV6**3))
*        CURT6 = CURT6/(NUM_BAT*(VAR6**2)) - 3.0
*      ELSE
*        GOTO 200
*      ENDIF
*      IF (VAR7 .NE. 0.0) THEN
*        SKEW7 = SKEW7/(NUM_BAT*(SDEV7**3))
*        CURT7 = CURT7/(NUM_BAT*(VAR7**2)) - 3.0
```

```
*      ELSE
*        GOTO 200
*      ENDIF
*      PRINT *,'SUBROUTINE MOMENT: PRINTING TO FILES'
190    FORMAT(1X,'VALUE 1: AVG=',1X,F12.3,1X,'VAR =',1X,F12.3)
191    FORMAT(1X,'VALUE 2: AVG=',1X,F12.3,1X,'VAR =',1X,F12.3)
192    FORMAT(1X,'VALUE 3: AVG=',1X,F12.3,1X,'VAR =',1X,F12.3)
200    WRITE(10,190)AVG1,VAR1
*      PRINT *,'AVG1=',AVG1,' AVG2=',AVG2,' AVG3=',AVG3
*      PRINT *,'VAR1=',VAR1,'  VAR2=',VAR2,'  VAR3=',VAR3
       WRITE(11,190)AVG1,VAR1
       WRITE(10,191)AVG2,VAR2
       WRITE(11,191)AVG2,VAR2
       WRITE(10,192)AVG3,VAR3
       WRITE(11,192)AVG3,VAR3
       A1 = AVG1
       V1 = VAR1
       A2 = AVG2
       V2 = VAR2
       A3 = AVG3
       V3 = VAR3
*      PRINT *,'END OF SUBROUTINE MOMENT'
       RETURN
       END
```

APPENDIX D

APPENDIX D

ILLUSTRATION OF MLC JOB SELECTION MECHANISM

This is an illustration of the MLC job selection mechanism. This heuristic uses two different parameters: X(9) and X(10). These were selected using the Complex response surface methodology algorithm described in chapter 4. This example uses two arbitrary values for X(9) and three arbitrary values for X(10).

The example is for the calculation of release dates for all the jobs in the pool at the beginning of one day. It is assumed that the shop is empty, so only jobs in the pool enter into the calculations. There are four work centers numbered 1,2,3, and 4. Each work center has one machine. There are four operators, one assigned to each work center.

JOBS IN THE POOL

JOB NUM	OPERATION 1		OPERATION 2		OPERATION 3		OPERATION 4		TOTAL	DUE
	WC	PROC	WC	PROC	WC	PROC	WC	PROC	PROC	DATE
1	1	1	2	2	3	1	4	2	6	8
2	2	2	3	2	4	1	1	1	6	7
3	3	1	4	1	1	2	2	1	5	6
4	4	2	1	1	2	2	3	2	7	5

STEP 1: CALCULATE DISCOUNT FACTOR FOR PERIODS 1 TO 4

$DF(I) = X(10)^{I-1}$

where I = periods or stage in routing 1 to 4.

PERIOD	X(10) = 1	X(10) = 0.5	X(10) = 0
1	1	1	1
2	1	.5	0
3	1	.25	0
4	1	.125	0

STEP 2: CALCULATE DISCOUNTED LOAD AT EACH WORK CENTER. THIS IS DONE IN TWO STAGES. FIRST DISCOUNT PROCESSING TIME AT EACH JOB. SECOND CALCULATE DISCOUNTED LOAD FOR WORK CENTER.

STAGE 1: DF(I)*(OPERATION PROCESSING TIME)$_I$

FOR X(10) = 1, THERE IS NO DISCOUNTING.

FOR X(10) = 0.5, DISCOUNTING IS BELOW.

JOB NUM	OPERATION 1 JOBS IN THE POOL WC	DF(1)	OPERATION 2 WC	DF(2)	OPERATION 3 WC	DF(3)	OPERATION 4 WC	DF(4)
1	1	1	2	1	3	.25	4	.25
2	2	2	3	1	4	.25	1	.125
3	3	1	4	.5	1	.5	2	.125
4	4	2	1	.5	2	.5	3	.25

FOR X(10) = 0, DISCOUNTING IS BELOW.

JOB NUM	OPERATION 1 JOBS IN THE POOL WC	DF(1)	OPERATION 2 WC	DF(2)	OPERATION 3 WC	DF(3)	OPERATION 4 WC	DF(4)
1	1	1	2	0	3	0	4	0
2	2	2	3	0	4	0	1	0
3	3	1	4	0	1	0	2	0
4	4	2	1	0	2	0	3	0

STAGE 2: CALCULATE DISCOUNTED LOAD AT WORK CENTERS

$$DLOAD(WC) = \sum_{jobs} \sum_{i=1}^{4} DF(I)*OPN$$

where i= operation number 1 to 4
and OPN = operation processing time

DISCOUNTED LOAD AT WC	X(10) = 1	X(10) = 0.5	X(10) = 0
DLOAD(1)	5	2.125	1
DLOAD(2)	7	3.625	2
DLOAD(3)	6	2.50	1
DLOAD(4)	6	3.0	2

STEP 3: CALCULATE CONGESTION FOR EACH JOBS

$$CONG(JOB) = DLOAD(WC_1) + \sum_{i=2}^{4} DLOAD(WC_i)/X(9)$$

CONGESTION FOR X(9) = 1

JOB	X(10) = 1 CONG	X(10) = .5 CONG	X(10) = 0 CONG
1	24	11.250	6
2	24	11.250	6
3	24	11.250	6
4	24	11.250	6

CONGESTION FOR X(9) = 10

JOB	X(10) = 1 CONG	X(10) = .5 CONG	X(10) = 0 CONG
1	6.9	3.04	1.5
2	8.7	4.29	2.4
3	7.8	3.38	1.5
4	7.8	3.83	2.4

STEP 4: CALCULATE RELEASE DATE FOR EACH JOB

RD = DD - PROC - CONG

where RD = RELEASE DATE; DD= DUE DATE;
PROC = TOTAL PROCESSING TIME OF JOB;
CONG = CONGESTION CALCULATED ABOVE.

CONGESTION FOR X(9) = 1

JOB	X(10) = 1 RD	X(10) = .5 RD	X(10) = 0 RD
1	-22	-9.25	-4
2	-23	-10.25	-5
3	-23	-10.25	-5
4	-26	-13.25	-8

CONGESTION FOR X(9) = 10

JOB	X(10) = 1 RD	X(10) = .5 RD	X(10) = 0 RD
1	-4.9	-1.038	.5
2	-7.7	-3.29	-1.4
3	-6.8	-2.28	-.5
4	-9.8	-5.83	-4.4

The release dates shown here are all negative except for one. That means that all the jobs would be eligible for release immediately. To avoid this, its necessary to pick a combination of parameters X(9) and X(10) so that the resulting release dates are not lumped together.

APPENDIX E

APPENDIX E

RATIO OF VARIANCES BY CELLS
ARRANGED IN ASCENDING ORDER OF VARIANCE OF RECIPRICOL COST*

CELLS

		3311	3331	3211	3211	3111	3231	3131	3321	3121	2211	2231	4331	2311
		4111	4324	2321	4221	2331	4211	4311	1331	1231	4131	1321	1221	1131
		2131	2111	1211	1121	4231	1311	2121	4121	2221	1111	3232	3322	3332
		3312	3232	3212	3132	3122	3112	1332	1232	1322	1212	2232	1312	4132
		1112	2212	2312	2222	4212	2132	1122	4122	2322	4112	4232	4332	2112
		4322	1222	4222	2122									
CELL	VARIANCE (10^{-7})													
3311	3.89	1.000	1.185	1.221	1.244	1.262	1.267	1.269	1.406	1.434	2.843	2.874	2.897	3.089
		3.195	3.470	3.622	3.712	3.874	3.899	4.097	4.161	4.290	4.508	4.688	4.706	4.879
		4.933	5.264	5.555	5.740	5.776	6.007	6.478	6.709	7.107	7.449	7.537	7.735	7.763
		7.917	8.079	8.082	8.308	8.339	8.501	11.100	11.287	11.444	12.048	12.496	12.717	13.336
		13.357	13.470	14.038	14.174	14.365	15.339	15.755	16.197	16.753	16.922	16.984	16.994	17.108
		17.529	18.126	19.444	19.557	19.868	20.367	21.563						
3331	4.61	1.185	1.000	1.030	1.049	1.065	1.069	1.071	1.186	1.210	2.399	2.425	2.444	2.607
		2.696	2.928	3.056	3.132	3.268	3.290	3.457	3.511	3.620	3.804	3.956	3.971	4.117
		4.162	4.442	4.687	4.843	4.874	5.069	5.466	5.661	5.997	6.286	6.360	6.527	6.550
		6.681	6.817	6.819	7.010	7.036	7.173	9.366	9.524	9.657	10.167	10.544	10.731	11.253
		11.271	11.366	11.846	11.961	12.121	12.943	13.295	13.668	14.136	14.279	14.331	14.340	14.436
		14.791	15.295	16.407	16.503	16.765	17.186	18.195						
3211	4.75	1.221	1.030	1.000	1.018	1.033	1.037	1.040	1.157	1.174	2.328	2.353	2.372	2.530
		2.616	2.842	2.966	3.040	3.172	3.193	3.355	3.408	3.513	3.692	3.840	3.854	3.995
		4.040	4.311	4.549	4.701	4.730	4.920	5.305	5.494	5.821	6.101	6.172	6.334	6.357
		6.484	6.616	6.618	6.804	6.824	6.962	9.090	9.244	9.372	9.867	10.233	10.414	10.922
		10.938	11.031	11.496	11.608	11.764	12.562	12.903	13.265	13.720	13.858	13.909	13.917	14.010
		14.355	14.844	15.924	16.016	16.271	16.680	17.658						
3221	4.84	1.244	1.049	1.018	1.000	1.014	1.018	1.020	1.130	1.152	2.285	2.309	2.328	2.483
		2.568	2.789	2.911	2.983	3.113	3.134	3.293	3.345	3.448	3.623	3.768	3.783	3.921
		3.964	4.231	4.464	4.613	4.642	4.828	5.206	5.392	5.712	5.987	6.057	6.216	6.239
		6.363	6.493	6.495	6.677	6.709	6.832	8.921	9.072	9.198	9.683	10.043	10.221	10.719
		10.735	10.826	11.283	11.392	11.545	12.328	12.663	13.018	13.464	13.601	13.650	13.659	13.750
		14.088	14.568	15.628	15.719	15.969	16.369	17.330						

(cont'd)

3111	4.91	1.262	1.065	1.033	1.014	1.0	1.004	1.006	1.114	1.136	2.252	2.276	2.295	2.448
		2.531	2.749	2.869	2.940	3.9	3.089	3.246	3.297	3.399	3.572	3.714	3.729	3.865
		3.908	4.171	4.401	4.547	4.6	4.759	5.132	5.315	5.631	5.902	5.971	6.128	6.150
		6.272	6.401	6.403	6.582	6.6	6.735	8.794	8.942	9.067	9.545	9.900	10.075	10.566
		10.582	10.672	11.122	11.230	11.0	12.152	12.482	12.833	13.272	13.407	13.456	13.464	13.554
		13.888	14.360	15.405	15.494	15.1	16.136	17.083						
3231	4.93	1.267	1.069	1.037	1.018	1.4	1.000	1.002	1.109	1.131	2.243	2.267	2.286	2.438
		2.521	2.738	2.858	2.929	3.6	3.077	3.233	3.283	3.385	3.557	3.699	3.713	3.849
		3.892	4.154	4.383	4.529	4.7	4.740	5.111	5.294	5.608	5.878	5.947	6.103	6.125
		6.247	6.375	6.377	6.555	6.0	6.707	8.758	8.906	9.030	9.507	9.860	10.034	10.523
		10.539	10.628	11.077	11.184	11.4	12.103	12.432	12.780	13.219	13.352	13.401	13.409	13.499
		13.831	14.302	15.342	15.432	15.7	16.071	17.014						
3131	4.94	1.269	1.071	1.040	1.020	1.006	1.002	1.000	1.109	1.129	2.238	2.263	2.281	2.433
		2.516	2.732	2.852	2.923	3.0	3.070	3.226	3.277	3.378	3.550	3.692	3.706	3.842
		3.884	4.145	4.374	4.520	4.548	4.730	5.101	5.283	5.597	5.866	5.935	6.091	6.113
		6.234	6.362	6.364	6.542	6.566	6.694	8.740	8.888	9.012	9.487	9.840	10.014	10.502
		10.518	10.607	11.054	11.161	11.311	12.078	12.406	12.755	13.192	13.325	13.374	13.382	13.471
		13.803	14.273	15.311	15.400	15.645	16.038	16.979						
3321	5.47	1.406	1.186	1.151	1.130	1.114	1.109	1.107	1.000	1.020	2.021	2.043	2.060	2.197
		2.272	2.468	2.575	2.639	2.755	2.773	2.914	2.959	3.051	3.206	3.334	3.347	3.469
		3.508	3.744	3.950	4.082	4.107	4.272	4.606	4.771	5.054	5.297	5.360	5.500	5.521
		5.630	5.745	5.747	5.908	5.930	6.045	7.893	8.027	8.138	8.568	8.886	9.043	9.484
		9.499	9.579	9.983	10.080	10.215	10.908	11.204	11.519	11.914	12.034	12.078	12.085	12.166
		12.466	12.890	13.828	13.908	14.129	14.484	15.334						
3121	5.58	1.434	1.210	1.174	1.152	1.136	1.131	1.129	1.029	1.000	1.982	2.003	2.019	2.154
		2.227	2.419	2.525	2.587	2.700	2.718	2.856	2.901	2.991	3.143	3.268	3.281	3.401
		3.439	3.670	3.872	4.001	4.026	4.188	4.516	4.677	4.955	5.193	5.254	5.392	5.412
		5.519	5.632	5.634	5.792	5.813	5.926	7.738	7.869	7.978	8.399	8.711	8.865	9.297
		9.311	9.390	9.786	9.881	10.010	10.693	10.983	11.292	11.679	11.797	11.840	11.847	11.926
		12.220	12.636	13.555	13.634	13.851	14.198	15.032						
2211	11.06	2.843	2.399	2.328	2.285	2.252	2.243	2.238	2.023	1.982	1.000	1.010	1.018	1.086
		1.123	1.220	1.273	1.305	1.362	1.371	1.441	1.463	1.509	1.585	1.649	1.655	1.716
		1.735	1.851	1.953	2.018	2.031	2.113	2.278	2.359	2.500	2.620	2.650	2.720	2.730
		2.784	2.841	2.842	2.922	2.933	2.990	3.904	3.970	4.025	4.237	4.395	4.472	4.690
		4.698	4.737	4.937	4.985	5.052	5.395	5.541	5.697	.8.924	.9.520	.9.737	5.977	6.017
		6.165	6.375	6.839	6.878	6.988	7.163	7.584						

(cont'd)

2231	11.18	2.874	2.425	2.353	2.309	2.276	2.267	2.263	2.043	2.003	1.010	1.000	1.008	1.075
		1.111	1.207	1.260	1.291	1.347	1.356	1.425	1.448	1.492	1.568	1.631	1.637	1.697
		1.716	1.831	1.932	1.997	2.009	2.090	2.254	2.334	2.473	2.592	2.622	2.691	2.701
		2.754	2.811	2.812	2.890	2.901	2.957	3.862	3.927	3.982	4.192	4.347	4.424	4.640
		4.647	4.686	4.884	4.932	4.998	5.337	5.482	5.635	5.829	5.888	5.909	5.913	5.952
		6.099	6.306	6.765	6.805	6.913	7.08677	7.502						
4331	11.27	2.897	2.444	2.372	2.328	2.295	2.286	2.281	2.060	2.019	1.018	1.008	1.000	1.066
		1.102	1.197	1.250	1.281	1.337	1.346	1.414	1.436	1.480	1.556	1.618	1.624	1.684
		1.702	1.817	1.917	1.981	1.993	2.073	2.236	2.315	2.453	2.571	2.601	2.669	2.679
		2.732	2.788	2.789	2.867	2.878	2.934	3.831	3.896	3.950	4.158	4.313	4.389	4.603
		4.610	4.649	4.845	4.892	4.958	5.294	5.438	5.590	5.782	5.841	5.862	5.866	5.905
		6.050	6.256	6.711	6.750	6.858	7.030	7.442						
2311	12.02	3.089	2.607	2.530	2.483	2.448	2.438	2.433	2.195	2.154	1.086	1.075	1.066	1.000
		1.034	1.123	1.172	1.201	1.253	1.262	1.326	1.346	1.388	1.459	1.517	1.523	1.579
		1.596	1.703	1.797	1.857	1.869	1.944	2.096	2.171	2.300	2.410	2.439	2.503	2.512
		2.562	2.614	2.615	2.688	2.698	2.751	3.592	3.653	3.703	3.899	4.044	4.115	4.316
		4.322	4.359	4.543	4.587	4.648	4.964	5.099	5.242	5.421	5.476	5.496	5.500	5.536
		5.673	5.866	6.292	6.329	6.430	6.591	6.978						
4111	12.43	3.195	2.696	2.616	2.568	2.531	2.521	2.516	2.271	2.227	1.123	1.111	1.102	1.034
		1.000	1.086	1.133	1.161	1.212	1.220	1.282	1.302	1.342	1.411	1.467	1.473	1.526
		1.543	1.647	1.738	1.796	1.807	1.880	2.027	2.099	2.224	2.331	2.358	2.420	2.429
		2.477	2.528	2.529	2.600	2.609	2.660	3.473	3.532	3.581	3.770	3.910	3.979	4.173
		4.180	4.215	4.393	4.436	4.495	4.800	4.930	5.069	5.242	5.296	5.315	5.318	5.353
		5.485	5.672	6.085	6.120	6.218	6.374	6.748						
4324	13.50	3.470	2.928	2.842	2.789	2.749	2.738	2.732	2.463	2.419	1.220	1.207	1.197	1.123
		1.086	1.000	1.043	1.069	1.116	1.123	1.180	1.199	1.236	1.299	1.351	1.356	1.405
		1.421	1.517	1.600	1.654	1.664	1.731	1.866	1.933	2.048	2.146	2.171	2.228	2.237
		2.281	2.328	2.328	2.394	2.402	2.449	3.198	3.252	3.297	3.471	3.600	3.664	3.842
		3.848	3.881	4.045	4.084	4.139	4.420	4.540	4.667	4.827	4.876	4.894	4.897	4.929
		5.051	5.222	5.602	5.635	5.725	5.868	6.213						

(cont'd)

2321	14.09	3.622	3.056	2.966	2.911	2.869	2.858	2.852	2.573	2.525	1.273	1.260	1.250	1.172
		1.133	1.043	1.000	1.024	1.069	1.076	1.131	1.149	1.184	1.244	1.294	1.299	1.347
		1.361	1.453	1.533	1.584	1.594	1.658	1.788	1.852	1.962	2.056	2.080	2.135	2.143
		2.185	2.230	2.231	2.293	2.302	2.347	3.064	3.116	3.159	3.326	3.449	3.511	3.682
		3.687	3.718	3.875	3.913	3.965	4.234	4.349	4.471	4.625	4.672	4.689	4.691	4.723
		4.839	5.004	5.368	5.399	5.485	5.623	5.953						
4221	14.44	3.712	3.132	3.040	2.983	2.940	2.929	2.923	2.638	2.587	1.305	1.291	1.281	1.201
		1.161	1.069	1.024	1.000	1.043	1.050	1.103	1.121	1.155	1.214	1.263	1.268	1.314
		1.328	1.418	1.496	1.546	1.556	1.618	1.745	1.807	1.914	2.006	2.030	2.083	2.091
		2.132	2.176	2.177	2.238	2.246	2.290	2.990	3.040	3.083	3.245	3.366	3.425	3.592
		3.598	3.628	3.781	3.818	3.869	4.132	4.244	4.363	4.513	4.558	4.575	4.578	4.608
		4.722	4.882	5.238	5.268	5.352	5.486	5.808						
2331	15.07	3.874	3.268	3.172	3.113	3.069	3.056	3.050	2.755	2.700	1.362	1.347	1.337	1.253
		1.212	1.116	1.069	1.043	1.000	1.006	1.057	1.074	1.107	1.163	1.210	1.214	1.259
		1.273	1.358	1.433	1.481	1.491	1.550	1.672	1.731	1.834	1.923	1.945	1.996	2.003
		2.043	2.085	2.086	2.144	2.152	2.194	2.865	2.913	2.954	3.110	3.225	3.282	3.442
		3.447	3.477	3.623	3.658	3.708	3.959	4.067	4.181	4.324	4.368	4.384	4.386	4.416
		4.524	4.678	5.019	5.048	5.128	5.257	5.566						
4211	15.17	3.899	3.290	3.193	3.134	3.089	3.077	3.070	2.773	2.718	1.371	1.356	1.346	1.262
		1.220	1.123	1.076	1.050	1.006	1.000	1.050	1.067	1.100	1.156	1.202	1.206	1.251
		1.264	1.350	1.424	1.471	1.481	1.540	1.661	1.720	1.822	1.910	1.932	1.983	1.990
		2.030	2.071	2.072	2.130	2.138	2.179	2.846	2.894	2.934	3.089	3.204	3.261	3.419
		3.425	3.454	3.599	3.634	3.683	3.933	4.040	4.153	4.295	4.339	4.355	4.357	4.386
		4.495	4.647	4.986	5.015	5.094	5.222	5.529						
4311	15.94	4.097	3.457	3.355	3.293	3.246	3.233	3.226	2.914	2.856	1.441	1.425	1.414	1.326
		1.282	1.180	1.131	1.103	1.057	1.050	1.000	1.015	1.047	1.100	1.144	1.148	1.190
		1.203	1.284	1.355	1.400	1.409	1.466	1.580	1.637	1.734	1.818	1.839	1.887	1.894
		1.932	1.971	1.972	2.027	2.035	2.074	2.708	2.754	2.792	2.940	3.049	3.103	3.254
		3.259	3.287	3.425	3.459	3.505	3.743	3.845	3.952	4.088	4.129	4.144	4.147	4.175
		4.277	4.423	4.745	4.772	4.848	4.970	5.262						
1331	16.19	4.161	3.511	3.408	3.345	3.297	3.283	3.277	2.959	2.901	1.463	1.448	1.436	1.346
		1.302	1.199	1.149	1.121	1.074	1.067	1.015	1.000	1.030	1.083	1.126	1.130	1.172
		1.185	1.264	1.334	1.379	1.387	1.443	1.556	1.612	1.707	1.789	1.810	1.858	1.865
		1.902	1.941	1.941	1.996	2.003	2.042	2.667	2.712	2.749	2.894	3.002	3.055	3.204
		3.209	3.236	3.373	3.405	3.451	3.685	3.785	3.891	4.025	4.066	4.080	4.083	4.110
		4.211	4.355	4.672	4.699	4.773	4.893	5.180						
1231	16.69	4.290	3.620	3.513	3.448	3.399	3.385	3.378	3.051	2.991	1.509	1.492	1.480	1.388
		1.342	1.236	1.184	1.155	1.107	1.100	1.047	1.030	1.000	1.050	1.092	1.097	1.137
		1.149	1.227	1.294	1.337	1.346	1.400	1.509	1.563	1.656	1.736	1.756	1.802	1.809
		1.845	1.883	1.883	1.936	1.943	1.981	2.587	2.630	2.667	2.808	2.912	2.964	3.108
		3.113	3.139	3.272	3.303	3.348	3.575	3.672	3.775	3.904	3.944	3.958	3.961	3.987
		4.085	4.224	4.532	4.558	4.630	4.747	5.025						

(cont'd)

4131	17.54	4.508	3.804	3.692	3.623	3.572	3.557	3.550	3.206	3.143	1.585	1.568	1.556	1.459
		1.411	1.299	1.244	1.214	1.163	1.156	1.100	1.083	1.050	1.000	1.039	1.043	1.082
		1.094	1.167	1.232	1.273	1.281	1.332	1.436	1.488	1.576	1.652	1.671	1.715	1.721
		1.755	1.791	1.792	1.842	1.849	1.885	2.461	2.503	2.538	2.672	2.771	2.820	2.957
		2.962	2.987	3.113	3.143	3.185	3.401	3.494	3.592	3.715	3.753	3.766	3.769	3.794
		3.887	4.019	4.312	4.337	4.406	4.517	4.782						
1321	18.24	4.688	3.956	3.840	3.768	3.714	3.699	3.692	3.334	3.268	1.649	1.631	1.618	1.517
		1.467	1.351	1.294	1.263	1.210	1.202	1.144	1.126	1.092	1.039	1.000	1.003	1.040
		1.052	1.122	1.184	1.224	1.231	1.281	1.381	1.430	1.515	1.588	1.607	1.649	1.655
		1.688	1.723	1.723	1.771	1.778	1.813	2.367	2.407	2.440	2.569	2.665	2.712	2.844
		2.848	2.872	2.993	3.023	3.063	3.271	3.360	3.454	3.572	3.609	3.622	3.624	3.648
		3.738	3.865	4.146	4.171	4.237	4.343	4.598						
1221	18.31	4.704	3.974	3.852	3.781	3.726	3.713	3.703	3.347	3.281	1.655	1.637	1.624	1.523
		1.473	1.356	1.299	1.268	1.214	1.206	1.148	1.130	1.097	1.043	1.003	1.000	1.036
		1.048	1.118	1.180	1.219	1.227	1.276	1.376	1.425	1.510	1.582	1.601	1.643	1.649
		1.682	1.716	1.717	1.765	1.771	1.806	2.358	2.398	2.431	2.559	2.654	2.701	2.833
		2.837	2.861	2.982	3.011	3.051	3.258	3.347	3.441	3.559	3.595	3.608	3.610	3.634
		3.724	3.850	4.131	4.155	4.221	4.327	4.581						
1131	18.98	4.879	4.117	3.995	3.921	3.865	3.849	3.842	3.469	3.401	1.716	1.697	1.684	1.579
		1.526	1.405	1.347	1.314	1.259	1.251	1.190	1.172	1.137	1.082	1.040	1.036	1.000
		1.011	1.079	1.138	1.176	1.183	1.231	1.327	1.375	1.456	1.526	1.544	1.585	1.591
		1.622	1.655	1.656	1.702	1.709	1.742	2.275	2.313	2.345	2.469	2.561	2.606	2.733
		2.737	2.760	2.877	2.905	2.944	3.143	3.229	3.319	3.433	3.468	3.481	3.483	3.506
		3.592	3.714	3.985	4.008	4.072	4.174	4.419						
2131	19.19	4.933	4.162	4.040	3.964	3.908	3.892	3.884	3.508	3.439	1.735	1.716	1.702	1.596
		1.543	1.421	1.361	1.328	1.273	1.264	1.203	1.185	1.149	1.094	1.052	1.048	1.011
		1.000	1.067	1.126	1.163	1.170	1.217	1.313	1.360	1.440	1.510	1.527	1.568	1.573
		1.605	1.637	1.638	1.684	1.690	1.723	2.250	2.288	2.319	2.442	2.533	2.577	2.703
		2.707	2.730	2.845	2.873	2.911	3.109	3.193	3.283	3.396	3.430	3.442	3.445	3.467
		3.553	3.674	3.941	3.964	4.027	4.128	4.371						
2111	20.48	5.264	4.442	4.311	4.231	4.171	4.154	4.145	3.744	3.670	1.851	1.831	1.817	1.703
		1.647	1.517	1.453	1.418	1.358	1.350	1.284	1.264	1.227	1.167	1.122	1.118	1.079
		1.067	1.000	1.055	1.090	1.097	1.141	1.230	1.274	1.350	1.415	1.431	1.469	1.474
		1.503	1.534	1.535	1.578	1.583	1.614	2.108	2.144	2.173	2.288	2.373	2.415	2.533
		2.537	2.558	2.666	2.692	2.728	2.913	2.992	3.076	3.182	3.214	3.226	3.228	3.249
		3.329	3.442	3.693	3.714	3.773	3.868	4.095						

(cont'd)

1211	21.61	5.555	4.687	4.549	4.464	4.401	4.383	4.374	3.950	3.872	1.953	1.932	1.917	1.797
		1.738	1.600	1.533	1.496	1.433	1.424	1.355	1.334	1.294	1.232	1.184	1.180	1.138
		1.126	1.055	1.000	1.033	1.039	1.081	1.166	1.207	1.279	1.341	1.356	1.392	1.397
		1.425	1.454	1.454	1.495	1.501	1.530	1.998	2.031	2.060	2.168	2.249	2.289	2.400
		2.404	2.424	2.527	2.551	2.585	2.761	2.836	2.915	3.015	3.046	3.057	3.059	3.079
		3.155	3.262	3.500	3.520	3.576	3.666	3.881						
1121	22.33	5.740	4.843	4.701	4.613	4.547	4.529	4.520	4.082	4.001	2.018	1.997	1.981	1.857
		1.796	1.654	1.584	1.546	1.481	1.471	1.400	1.379	1.337	1.273	1.224	1.219	1.176
		1.163	1.090	1.033	1.000	1.006	1.046	1.128	1.168	1.238	1.297	1.313	1.347	1.352
		1.379	1.407	1.407	1.447	1.452	1.480	1.933	1.966	1.993	2.098	2.176	2.215	2.323
		2.326	2.346	2.445	2.469	2.502	2.672	2.744	2.821	2.918	2.948	2.958	2.960	2.980
		3.053	3.157	3.387	3.407	3.461	3.548	3.756						
4231	22.47	5.776	4.874	4.730	4.642	4.576	4.557	4.548	4.107	4.026	2.031	2.009	1.993	1.869
		1.807	1.664	1.594	1.556	1.491	1.481	1.409	1.387	1.346	1.281	1.231	1.227	1.183
		1.170	1.097	1.039	1.006	1.000	1.040	1.121	1.161	1.230	1.289	1.304	1.339	1.344
		1.370	1.398	1.399	1.438	1.443	1.471	1.921	1.954	1.981	2.085	2.163	2.201	2.308
		2.312	2.331	2.430	2.453	2.486	2.655	2.727	2.804	2.900	2.929	2.940	2.942	2.961
		3.034	3.137	3.366	3.385	3.439	3.526	3.732						
1311	23.37	6.007	5.069	4.920	4.828	4.759	4.740	4.730	4.272	4.188	2.113	2.090	2.073	1.944
		1.880	1.731	1.658	1.618	1.550	1.540	1.466	1.443	1.400	1.332	1.281	1.276	1.231
		1.217	1.141	1.081	1.046	1.040	1.000	1.078	1.116	1.183	1.240	1.254	1.287	1.292
		1.317	1.344	1.345	1.382	1.388	1.415	1.847	1.878	1.905	2.005	2.080	2.116	2.219
		2.223	2.242	2.336	2.359	2.391	2.553	2.622	2.696	2.788	2.816	2.827	2.828	2.847
		2.917	3.017	3.236	3.255	3.307	3.390	3.589						
2121	25.20	6.478	5.466	5.305	5.206	5.132	5.111	5.101	4.606	4.516	2.278	2.254	2.236	2.096
		2.027	1.866	1.788	1.745	1.672	1.661	1.580	1.556	1.509	1.436	1.381	1.376	1.327
		1.313	1.230	1.166	1.128	1.121	1.078	1.000	1.035	1.097	1.150	1.163	1.194	1.198
		1.222	1.247	1.247	1.282	1.287	1.312	1.713	1.742	1.766	1.859	1.928	1.963	2.058
		2.061	2.079	2.167	2.188	2.217	2.367	2.432	2.500	2.586	2.612	2.621	2.623	2.640
		2.705	2.798	3.001	3.019	3.067	3.144	3.328						
4121	26.10	6.709	5.661	5.494	5.392	5.315	5.294	5.283	4.771	4.677	2.359	2.334	2.315	2.171
		2.099	1.933	1.852	1.807	1.731	1.720	1.637	1.612	1.563	1.488	1.430	1.425	1.375
		1.360	1.274	1.207	1.168	1.161	1.116	1.035	1.000	1.059	1.110	1.123	1.152	1.157
		1.180	1.204	1.204	1.238	1.242	1.267	1.654	1.682	1.705	1.795	1.862	1.895	1.987
		1.990	2.007	2.092	2.112	2.140	2.286	2.348	2.414	2.496	2.522	2.531	2.532	2.549
		2.612	2.701	2.898	2.914	2.961	3.035	3.213						

1 (cont'd)

2221	27.65	7.107	5.997	5.821	5.712	5.631	5.608	5.597	5.054	4.955	2.500	2.473	2.453	2.300
		2.224	2.048	1.962	1.914	1.834	1.822	1.734	1.707	1.656	1.576	1.515	1.510	1.456
		1.440	1.350	1.279	1.238	1.230	1.183	1.097	1.059	1.000	1.048	1.060	1.088	1.092
		1.113	1.136	1.137	1.168	1.173	1.196	1.561	1.588	1.610	1.695	1.758	1.789	1.876
		1.879	1.895	1.975	1.994	2.020	2.158	2.216	2.278	2.356	2.380	2.389	2.390	2.406
		2.466	2.550	2.735	2.751	2.795	2.865	3.033						
1111	28.98	7.449	6.286	6.101	5.987	5.902	5.878	5.866	5.297	5.193	2.620	2.592	2.571	2.410
		2.331	2.146	2.056	2.006	1.923	1.910	1.818	1.789	1.736	1.652	1.588	1.582	1.526
		1.510	1.415	1.341	1.297	1.289	1.240	1.150	1.110	1.048	1.000	1.011	1.038	1.042
		1.062	1.084	1.084	1.115	1.119	1.141	1.489	1.515	1.536	1.617	1.677	1.707	1.790
		1.792	1.808	1.884	1.902	1.928	2.059	2.114	2.174	2.248	2.271	2.279	2.281	2.296
		2.353	2.433	2.610	2.625	2.667	2.733	2.894						
3222	29.32	7.537	6.360	6.172	6.057	5.971	5.947	5.935	5.360	5.254	2.650	2.622	2.601	2.439
		2.358	2.171	2.080	2.030	1.945	1.932	1.839	1.810	1.756	1.671	1.607	1.601	1.544
		1.527	1.431	1.356	1.313	1.304	1.254	1.163	1.123	1.060	1.011	1.000	1.026	1.030
		1.050	1.071	1.072	1.102	1.106	1.127	1.472	1.497	1.518	1.598	1.657	1.687	1.769
		1.772	1.787	1.862	1.880	1.905	2.035	2.090	2.149	2.222	2.245	2.253	2.254	2.269
		2.325	2.404	2.579	2.594	2.636	2.702	2.860						
3322	30.09	7.735	6.527	6.334	6.216	6.128	6.103	6.091	5.500	5.392	2.720	2.691	2.669	2.503
		2.420	2.228	2.135	2.083	1.996	1.983	1.887	1.858	1.802	1.715	1.649	1.643	1.585
		1.568	1.469	1.392	1.347	1.339	1.287	1.194	1.152	1.088	1.038	1.026	1.000	1.003
		1.023	1.044	1.044	1.074	1.078	1.099	1.435	1.459	1.479	1.557	1.615	1.644	1.724
		1.726	1.741	1.814	1.832	1.857	1.983	2.036	2.094	2.165	2.187	2.195	2.197	2.211
		2.266	2.343	2.513	2.528	2.568	2.633	2.787						
3332	30.20	7.763	6.550	6.357	6.239	6.150	6.125	6.113	5.521	5.412	2.730	2.701	2.679	2.512
		2.429	2.237	2.143	2.091	2.003	1.990	1.894	1.865	1.809	1.721	1.655	1.649	1.591
		1.573	1.474	1.397	1.352	1.344	1.292	1.198	1.157	1.092	1.042	1.030	1.003	1.000
		1.019	1.040	1.041	1.070	1.074	1.095	1.429	1.453	1.474	1.551	1.609	1.638	1.717
		1.720	1.735	1.808	1.825	1.850	1.975	2.029	2.086	2.157	2.179	2.187	2.189	2.203
		2.257	2.334	2.504	2.519	2.559	2.623	2.777						
3312	30.80	7.917	6.681	6.484	6.363	6.272	6.247	6.234	5.630	5.519	2.784	2.754	2.732	2.562
		2.477	2.281	2.185	2.132	2.043	2.030	1.932	1.902	1.845	1.755	1.688	1.682	1.622
		1.605	1.503	1.425	1.379	1.370	1.317	1.222	1.180	1.113	1.062	1.050	1.023	1.019
		1.000	1.020	1.020	1.049	1.053	1.073	1.401	1.425	1.445	1.521	1.578	1.606	1.684
		1.687	1.701	1.773	1.790	1.814	1.937	1.989	2.045	2.115	2.137	2.145	2.146	2.160
		2.213	2.289	2.455	2.470	2.509	2.572	2.723						

(cont'd)

3232	31.43	8.079	6.817	6.616	6.493	6.401	6.375	6.362	5.745	5.632	2.841	2.811	2.788	2.614
		2.528	2.328	2.230	2.176	2.085	2.071	1.971	1.941	1.883	1.791	1.723	1.716	1.655
		1.637	1.534	1.454	1.407	1.398	1.344	1.247	1.204	1.136	1.084	1.071	1.044	1.040
		1.020	1.000	1.000	1.028	1.032	1.052	1.373	1.397	1.416	1.491	1.546	1.573	1.650
		1.653	1.667	1.737	1.754	1.777	1.898	1.950	2.004	2.073	2.094	2.102	2.103	2.117
		2.169	2.243	2.406	2.420	2.459	2.520	2.668						
3212	31.44	8.082	6.819	6.618	6.495	6.403	6.377	6.364	5.747	5.634	2.842	2.812	2.789	2.615
		2.529	2.328	2.231	2.177	2.086	2.072	1.972	1.941	1.883	1.792	1.723	1.717	1.656
		1.638	1.535	1.454	1.407	1.399	1.345	1.247	1.204	1.137	1.084	1.072	1.044	1.041
		1.020	1.000	1.000	1.027	1.031	1.051	1.373	1.396	1.416	1.490	1.546	1.573	1.650
		1.652	1.666	1.736	1.753	1.777	1.897	1.949	2.004	2.072	2.093	2.101	2.102	2.116
		2.168	2.242	2.405	2.419	2.458	2.520	2.667						
3132	32.32	8.308	7.010	6.804	6.677	6.582	6.555	6.542	5.908	5.792	2.922	2.890	2.867	2.688
		2.600	2.394	2.293	2.238	2.144	2.130	2.027	1.996	1.936	1.842	1.771	1.765	1.702
		1.684	1.578	1.495	1.447	1.438	1.382	1.282	1.238	1.168	1.115	1.102	1.074	1.070
		1.049	1.028	1.027	1.000	1.003	1.023	1.336	1.358	1.377	1.450	1.504	1.530	1.605
		1.607	1.621	1.689	1.706	1.728	1.846	1.896	1.949	2.016	2.036	2.044	2.045	2.059
		2.109	2.181	2.340	2.353	2.391	2.451	2.595						
3122	32.44	8.339	7.036	6.829	6.702	6.606	6.580	6.566	5.930	5.813	2.933	2.901	2.878	2.698
		2.609	2.402	2.302	2.246	2.152	2.138	2.035	2.003	1.943	1.849	1.778	1.771	1.709
		1.690	1.583	1.501	1.452	1.443	1.388	1.287	1.242	1.173	1.119	1.106	1.078	1.074
		1.053	1.032	1.031	1.003	1.000	1.019	1.331	1.353	1.372	1.444	1.498	1.524	1.599
		1.601	1.615	1.683	1.699	1.722	1.839	1.889	1.942	2.008	2.029	2.036	2.037	2.051
		2.102	2.173	2.331	2.345	2.382	2.442	2.585						
3112	33.07	8.501	7.173	6.962	6.832	6.735	6.707	6.694	6.045	5.926	2.990	2.957	2.934	2.751
		2.660	2.449	2.347	2.290	2.194	2.179	2.074	2.042	1.981	1.885	1.813	1.806	1.742
		1.723	1.614	1.530	1.480	1.471	1.415	1.312	1.267	1.196	1.141	1.127	1.099	1.095
		1.073	1.052	1.051	1.023	1.019	1.000	1.305	1.327	1.346	1.417	1.469	1.495	1.568
		1.571	1.584	1.651	1.667	1.689	1.804	1.853	1.905	1.970	1.990	1.997	1.999	2.012
		2.061	2.132	2.287	2.300	2.337	2.395	2.536						
1332	43.18	11.10	9.366	9.090	8.921	8.794	8.758	8.740	7.893	7.738	3.904	3.862	3.831	3.592
		3.473	3.198	3.064	2.990	2.865	2.846	2.708	2.667	2.587	2.461	2.367	2.358	2.275
		2.250	2.108	1.998	1.933	1.921	1.847	1.713	1.654	1.561	1.489	1.472	1.435	1.429
		1.401	1.373	1.373	1.336	1.331	1.305	1.000	1.016	1.031	1.085	1.125	1.145	1.201
		1.203	1.213	1.264	1.276	1.294	1.381	1.419	1.459	1.509	1.524	1.530	1.531	1.541
		1.579	1.632	1.751	1.761	1.789	1.834	1.942						

(cont'd)

1232	43.91	11.28	9.524	9.244	9.072	8.942	8.906	8.888	8.027	7.869	3.970	3.927	3.896	3.653
		3.532	3.252	3.116	3.040	2.913	2.894	2.754	2.712	2.630	2.503	2.407	2.398	2.313
		2.288	2.144	2.031	1.966	1.954	1.878	1.742	1.682	1.588	1.515	1.497	1.459	1.453
		1.425	1.397	1.396	1.358	1.353	1.327	1.016	1.000	1.013	1.067	1.107	1.126	1.181
		1.183	1.193	1.243	1.255	1.272	1.358	1.395	1.434	1.484	1.499	1.504	1.505	1.515
		1.552	1.605	1.722	1.732	1.760	1.804	1.910						
1322	44.52	11.44	9.657	9.372	9.198	9.067	9.030	9.012	8.138	7.978	4.025	3.982	3.950	3.703
		3.581	3.297	3.159	3.083	2.954	2.934	2.792	2.749	2.667	2.538	2.440	2.431	2.345
		2.319	2.173	2.060	1.993	1.981	1.905	1.766	1.705	1.610	1.536	1.518	1.479	1.474
		1.445	1.416	1.416	1.377	1.372	1.346	1.031	1.013	1.000	1.052	1.091	1.111	1.165
		1.167	1.176	1.226	1.238	1.255	1.340	1.376	1.415	1.463	1.478	1.484	1.484	1.494
		1.531	1.583	1.699	1.708	1.736	1.779	1.884						
1212	46.87	12.04	10.16	9.867	9.683	9.545	9.507	9.487	8.568	8.399	4.237	4.192	4.158	3.899
		3.770	3.471	3.326	3.245	3.110	3.089	2.940	2.894	2.808	2.672	2.569	2.559	2.469
		2.442	2.288	2.168	2.098	2.085	2.005	1.859	1.795	1.695	1.617	1.598	1.557	1.551
		1.521	1.491	1.490	1.450	1.444	1.417	1.085	1.067	1.052	1.000	1.037	1.055	1.106
		1.108	1.117	1.165	1.176	1.192	1.273	1.307	1.344	1.390	1.404	1.409	1.410	1.419
		1.454	1.504	1.613	1.623	1.649	1.690	1.789						
2232	48.61	12.49	10.54	10.23	10.04	9.900	9.860	9.840	8.886	8.711	4.395	4.347	4.313	4.044
		3.910	3.600	3.449	3.366	3.225	3.204	3.049	3.002	2.912	2.771	2.665	2.654	2.561
		2.533	2.373	2.249	2.176	2.163	2.080	1.928	1.862	1.758	1.677	1.657	1.615	1.609
		1.578	1.546	1.546	1.504	1.498	1.469	1.125	1.107	1.091	1.037	1.000	1.017	1.067
		1.068	1.077	1.123	1.134	1.149	1.227	1.260	1.296	1.340	1.354	1.359	1.360	1.369
		1.402	1.450	1.556	1.565	1.590	1.629	1.725						
1312	49.47	12.71	10.73	10.41	10.22	10.07	10.03	10.01	9.043	8.865	4.472	4.424	4.389	4.115
		3.979	3.664	3.511	3.425	3.282	3.261	3.103	3.055	2.964	2.820	2.712	2.701	2.606
		2.577	2.415	2.289	2.215	2.201	2.116	1.963	1.895	1.789	1.707	1.687	1.644	1.638
		1.606	1.573	1.573	1.530	1.524	1.495	1.145	1.126	1.111	1.055	1.017	1.000	1.048
		1.050	1.059	1.103	1.114	1.129	1.206	1.238	1.273	1.317	1.330	1.335	1.336	1.345
		1.378	1.425	1.529	1.537	1.562	1.601	1.695						
4132	51.88	13.33	11.25	10.92	10.71	10.56	10.52	10.50	9.484	9.297	4.690	4.640	4.603	4.316
		4.173	3.842	3.682	3.592	3.442	3.419	3.254	3.204	3.108	2.957	2.844	2.833	2.733
		2.703	2.533	2.400	2.323	2.308	2.219	2.058	1.987	1.876	1.790	1.769	1.724	1.717
		1.684	1.650	1.650	1.605	1.599	1.568	1.201	1.181	1.165	1.106	1.067	1.048	1.000
		1.001	1.010	1.052	1.062	1.077	1.150	1.181	1.214	1.256	1.268	1.273	1.274	1.282
		1.314	1.359	1.457	1.466	1.489	1.527	1.616						

(cont'd)

1132	51.96	13.35	11.27	10.93	10.73	10.58	10.53	10.51	9.499	9.311	4.698	4.647	4.610	4.322
		4.180	3.848	3.687	3.598	3.447	3.425	3.259	3.209	3.113	2.962	2.848	2.837	2.737
		2.707	2.537	2.404	2.326	2.312	2.223	2.061	1.990	1.879	1.792	1.772	1.726	1.720
		1.687	1.653	1.652	1.607	1.601	1.571	1.203	1.183	1.167	1.108	1.068	1.050	1.001
		1.000	1.008	1.051	1.061	1.075	1.148	1.179	1.212	1.254	1.266	1.271	1.272	1.280
		1.312	1.357	1.455	1.464	1.487	1.524	1.614						
2332	52.40	13.47	11.36	11.03	10.82	10.67	10.62	10.60	9.579	9.390	4.737	4.686	4.649	4.359
		4.215	3.881	3.718	3.628	3.477	3.454	3.287	3.236	3.139	2.987	2.872	2.861	2.760
		2.730	2.558	2.424	2.346	2.331	2.242	2.079	2.007	1.895	1.808	1.787	1.741	1.735
		1.701	1.667	1.666	1.621	1.615	1.584	1.213	1.193	1.176	1.117	1.077	1.059	1.010
		1.008	1.000	1.042	1.052	1.066	1.138	1.169	1.202	1.243	1.256	1.260	1.261	1.270
		1.301	1.345	1.443	1.451	1.475	1.512	1.600						
4312	54.61	14.03	11.84	11.49	11.28	11.12	11.07	11.05	9.983	9.786	4.937	4.884	4.845	4.543
		4.393	4.045	3.875	3.781	3.623	3.599	3.425	3.373	3.272	3.113	2.993	2.982	2.877
		2.845	2.666	2.527	2.445	2.430	2.336	2.167	2.092	1.975	1.884	1.862	1.814	1.808
		1.773	1.737	1.736	1.689	1.683	1.651	1.264	1.243	1.226	1.165	1.123	1.103	1.052
		1.051	1.042	1.000	1.009	1.023	1.092	1.122	1.153	1.193	1.205	1.209	1.210	1.218
		1.248	1.291	1.385	1.393	1.415	1.450	1.535						
1112	55.14	14.17	11.96	11.60	11.39	11.23	11.18	11.16	10.08	9.881	4.985	4.932	4.892	4.587
		4.436	4.084	3.913	3.818	3.658	3.634	3.459	3.405	3.303	3.143	3.023	3.011	2.905
		2.873	2.692	2.551	2.469	2.453	2.359	2.188	2.112	1.994	1.902	1.880	1.832	1.825
		1.790	1.754	1.753	1.706	1.699	1.667	1.276	1.255	1.238	1.176	1.134	1.114	1.062
		1.061	1.052	1.009	1.000	1.013	1.082	1.111	1.142	1.181	1.193	1.198	1.198	1.206
		1.236	1.278	1.371	1.379	1.401	1.436	1.521						
2212	55.88	14.36	12.12	11.76	11.54	11.38	11.33	11.31	10.21	10.01	5.052	4.998	4.958	4.648
		4.495	4.139	3.965	3.869	3.708	3.683	3.505	3.451	3.348	3.185	3.063	3.051	2.944
		2.911	2.728	2.585	2.502	2.486	2.391	2.217	2.140	2.020	1.928	1.905	1.857	1.850
		1.814	1.777	1.777	1.728	1.722	1.689	1.294	1.272	1.255	1.192	1.149	1.129	1.077
		1.075	1.066	1.023	1.013	1.000	1.067	1.096	1.127	1.166	1.178	1.182	1.183	1.190
		1.220	1.261	1.353	1.361	1.383	1.417	1.501						
2312	59.67	15.33	12.94	12.56	12.32	12.15	12.10	12.07	10.90	10.69	5.395	5.337	5.294	4.964
		4.800	4.420	4.234	4.132	3.959	3.933	3.743	3.685	3.575	3.401	3.271	3.258	3.143
		3.109	2.913	2.761	2.672	2.655	2.553	2.367	2.286	2.158	2.059	2.035	1.983	1.975
		1.937	1.898	1.897	1.846	1.839	1.804	1.381	1.358	1.340	1.273	1.227	1.206	1.150
		1.148	1.138	1.092	1.082	1.067	1.000	1.027	1.055	1.092	1.103	1.107	1.107	1.115
		1.142	1.181	1.267	1.275	1.295	1.327	1.405						

(cont'd)

2222	61.29	15.75	13.29	12.90	12.66	12.48	12.43	12.40	11.20	10.98	5.541	5.482	5.438	5.099
		4.930	4.540	4.349	4.244	4.067	4.040	3.845	3.785	3.672	3.494	3.360	3.347	3.229
		3.193	2.992	2.836	2.744	2.727	2.622	2.432	2.348	2.216	2.114	2.090	2.036	2.029
		1.989	1.950	1.949	1.896	1.889	1.853	1.419	1.395	1.376	1.307	1.260	1.238	1.181
		1.179	1.169	1.122	1.111	1.096	1.027	1.000	1.028	1.063	1.074	1.077	1.078	1.085
		1.112	1.150	1.234	1.241	1.261	1.292	1.368						
4212	63.01	16.19	13.66	13.26	13.01	12.83	12.78	12.75	11.51	11.29	5.697	5.635	5.590	5.242
		5.069	4.667	4.471	4.363	4.181	4.153	3.952	3.891	3.775	3.592	3.454	3.441	3.319
		3.283	3.076	2.915	2.821	2.804	2.696	2.500	2.414	2.278	2.174	2.149	2.094	2.086
		2.045	2.004	2.004	1.949	1.942	1.905	1.459	1.434	1.415	1.344	1.296	1.273	1.214
		1.212	1.202	1.153	1.142	1.127	1.055	1.028	1.000	1.034	1.044	1.048	1.049	1.056
		1.082	1.119	1.200	1.207	1.226	1.257	1.331						
2132	65.17	16.75	14.13	13.72	13.46	13.27	13.21	13.19	11.91	11.67	5.892	5.829	5.782	5.421
		5.242	4.827	4.625	4.513	4.324	4.295	4.088	4.025	3.904	3.715	3.572	3.559	3.433
		3.396	3.182	3.015	2.918	2.900	2.788	2.586	2.496	2.356	2.248	2.222	2.165	2.157
		2.115	2.073	2.072	2.016	2.008	1.970	1.509	1.484	1.463	1.390	1.340	1.317	1.256
		1.254	1.243	1.193	1.181	1.166	1.092	1.063	1.034	1.000	1.010	1.013	1.014	1.021
		1.046	1.081	1.160	1.167	1.185	1.215	1.287						
1122	65.83	16.92	14.27	13.85	13.60	13.40	13.35	13.32	12.03	11.79	5.952	5.888	5.841	5.476
		5.296	4.876	4.672	4.558	4.368	4.339	4.129	4.066	3.944	3.753	3.609	3.595	3.468
		3.430	3.214	3.046	2.948	2.929	2.816	2.612	2.522	2.380	2.271	2.245	2.187	2.179
		2.137	2.094	2.093	2.036	2.029	1.990	1.524	1.499	1.478	1.404	1.354	1.330	1.268
		1.266	1.256	1.205	1.193	1.178	1.103	1.074	1.044	1.010	1.000	1.003	1.004	1.010
		1.035	1.071	1.149	1.155	1.174	1.203	1.274						
4122	66.07	16.98	14.33	13.90	13.65	13.45	13.40	13.37	12.07	11.84	5.973	5.909	5.862	5.496
		5.315	4.894	4.689	4.575	4.384	4.355	4.144	4.080	3.958	3.766	3.622	3.608	3.481
		3.442	3.226	3.057	2.958	2.940	2.827	2.621	2.531	2.389	2.279	2.253	2.195	2.187
		2.145	2.102	2.101	2.044	2.036	1.997	1.530	1.504	1.484	1.409	1.359	1.335	1.273
		1.271	1.260	1.209	1.198	1.182	1.107	1.077	1.048	1.013	1.003	1.000	1.000	1.007
		1.032	1.067	1.144	1.151	1.169	1.199	1.269						
2322	66.11	16.99	14.34	13.91	13.65	13.46	13.40	13.38	12.08	11.84	5.977	5.913	5.866	5.500
		5.318	4.897	4.691	4.578	4.386	4.357	4.147	4.083	3.961	3.769	3.624	3.610	3.483
		3.445	3.228	3.059	2.960	2.942	2.828	2.623	2.532	2.390	2.281	2.254	2.197	2.189
		2.146	2.103	2.102	2.045	2.037	1.999	1.531	1.505	1.484	1.410	1.360	1.336	1.274
		1.272	1.261	1.210	1.198	1.183	1.107	1.078	1.049	1.014	1.004	1.000	1.000	1.006
		1.031	1.066	1.144	1.150	1.169	1.198	1.268						

(cont'd)

4112	66.55	17.10	14.43	14.01	13.75	13.55	13.49	13.47	12.16	11.92	6.017	5.952	5.905	5.536
		5.353	4.929	4.723	4.608	4.416	4.386	4.175	4.110	3.987	3.794	3.648	3.634	3.506
		3.467	3.249	3.079	2.980	2.961	2.847	2.640	2.549	2.406	2.296	2.269	2.211	2.203
		2.160	2.117	2.116	2.059	2.051	2.012	1.541	1.515	1.494	1.419	1.369	1.345	1.282
		1.280	1.270	1.218	1.206	1.190	1.115	1.085	1.056	1.021	1.010	1.007	1.006	1.000
		1.024	1.059	1.136	1.143	1.161	1.190	1.260						
4232	68.19	17.52	14.79	14.35	14.08	13.88	13.83	13.80	12.46	12.22	6.165	6.099	6.050	5.673
		5.485	5.051	4.839	4.722	4.524	4.495	4.277	4.211	4.085	3.887	3.738	3.724	3.592
		3.553	3.329	3.155	3.053	3.034	2.917	2.705	2.612	2.466	2.353	2.325	2.266	2.257
		2.213	2.169	2.168	2.109	2.102	2.061	1.579	1.552	1.531	1.454	1.402	1.378	1.314
		1.312	1.301	1.248	1.236	1.220	1.142	1.112	1.082	1.046	1.035	1.032	1.031	1.024
		1.000	1.034	1.109	1.115	1.133	1.161	1.230						
4332	70.51	18.12	15.29	14.84	14.56	14.36	14.30	14.27	12.89	12.63	6.375	6.306	6.256	5.866
		5.672	5.222	5.004	4.882	4.678	4.647	4.423	4.355	4.224	4.019	3.865	3.850	3.714
		3.674	3.442	3.262	3.157	3.137	3.017	2.798	2.701	2.550	2.433	2.404	2.343	2.334
		2.289	2.243	2.242	2.181	2.173	2.132	1.632	1.605	1.583	1.504	1.450	1.425	1.359
		1.357	1.345	1.291	1.278	1.261	1.181	1.150	1.119	1.081	1.071	1.067	1.066	1.059
		1.034	1.000	1.072	1.078	1.096	1.123	1.189						
2112	75.64	19.44	16.40	15.92	15.62	15.40	15.34	15.31	13.82	13.55	6.839	6.765	6.711	6.292
		6.085	5.602	5.368	5.238	5.019	4.986	4.745	4.672	4.532	4.312	4.146	4.131	3.985
		3.941	3.693	3.500	3.387	3.366	3.236	3.001	2.898	2.735	2.610	2.579	2.513	2.504
		2.455	2.406	2.405	2.340	2.331	2.287	1.751	1.722	1.699	1.613	1.556	1.529	1.457
		1.455	1.443	1.385	1.371	1.353	1.267	1.234	1.200	1.160	1.149	1.144	1.144	1.136
		1.109	1.072	1.000	1.005	1.021	1.047	1.108						
4322	76.08	19.55	16.50	16.01	15.71	15.49	15.43	15.40	13.90	13.63	6.878	6.805	6.750	6.329
		6.120	5.635	5.399	5.268	5.048	5.015	4.772	4.699	4.558	4.337	4.171	4.155	4.008
		3.964	3.714	3.520	3.407	3.385	3.255	3.019	2.914	2.751	2.625	2.594	2.528	2.519
		2.470	2.420	2.419	2.353	2.345	2.300	1.761	1.732	1.708	1.623	1.565	1.537	1.466
		1.464	1.451	1.393	1.379	1.361	1.275	1.241	1.207	1.167	1.155	1.151	1.150	1.143
		1.115	1.078	1.005	1.000	1.015	1.041	1.102						
1222	77.29	19.86	16.76	16.27	15.96	15.74	15.67	15.64	14.12	13.85	6.988	6.913	6.858	6.430
		6.218	5.725	5.485	5.352	5.128	5.094	4.848	4.773	4.630	4.406	4.237	4.221	4.072
		4.027	3.773	3.576	3.461	3.439	3.307	3.067	2.961	2.795	2.667	2.636	2.568	2.559
		2.509	2.459	2.458	2.391	2.382	2.337	1.789	1.760	1.736	1.649	1.590	1.562	1.489
		1.487	1.475	1.415	1.401	1.383	1.295	1.261	1.226	1.185	1.174	1.169	1.169	1.161
		1.133	1.096	1.021	1.015	1.000	1.025	1.085						

(cont'd)

4222	79.23	20.36	17.18	16.68	16.36	16.13	16.07	16.03	14.48	14.19	7.163	7.086	7.030	6.591
		6.374	5.868	5.623	5.486	5.257	5.222	4.970	4.893	4.747	4.517	4.343	4.327	4.174
		4.128	3.868	3.666	3.548	3.526	3.390	3.144	3.035	2.865	2.733	2.702	2.633	2.623
		2.572	2.520	2.520	2.451	2.442	2.395	1.834	1.804	1.779	1.690	1.629	1.601	1.527
		1.524	1.512	1.450	1.436	1.417	1.327	1.292	1.257	1.215	1.203	1.199	1.198	1.190
		1.161	1.123	1.047	1.041	1.025	1.000	1.058						
2122	83.88	21.56	18.19	17.65	17.33	17.08	17.01	16.97	15.33	15.03	7.584	7.502	7.442	6.978
		6.748	6.213	5.953	5.808	5.566	5.529	5.262	5.180	5.025	4.782	4.598	4.581	4.419
		4.371	4.095	3.881	3.756	3.732	3.589	3.328	3.213	3.033	2.894	2.860	2.787	2.777
		2.723	2.668	2.667	2.595	2.585	2.536	1.942	1.910	1.884	1.789	1.725	1.695	1.616
		1.614	1.600	1.535	1.521	1.501	1.405	1.368	1.331	1.287	1.274	1.269	1.268	1.260
		1.230	1.189	1.108	1.102	1.085	1.058	1.000						

* ALL VARIANCES $\sigma^2_{max}/\sigma^2_{min}$

LIST OF REFERENCES

LIST OF REFERENCES

Ackerman, Sanford S., "Even-Flow, A Scheduling Method for Reducing Lateness in Job Shops", Management Technology, Volume 3, No. 1(1963), pp.20-32.

Ashton, James, A., Cook,F.X., "Time To Reform Job Shop Manufacturing", Harvard Business Review, Volume 67, No. 2, (1989), pp. 106-111.

Allen, Morton, "The Efficient Utilization of Labor Under Conditions of Fluctuating Demand", in Industrial Scheduling, eds. John F. Muth, Gerald L. Thompson and Peter R. Winters, Prentice-Hall, Englewood Cliffs, New Jersey, 1963, pp.252-276.

Avi-Itzhak, B., Maxwell, W.L., Miller, L.W., "Queuing With Alternating Priorities", Operations Research, Volume 13, No. 2, (1965), pp.306-318.

Bain, Lee J., Engelhardt, M., Introduction to Probability and Mathematical Statistics, PWS Publishers, Boston Massachusetts, 1987.

Baker, Kenneth R., Introduction to Sequencing and Scheduling, John Wiley & Sons, New York, New York, 1974.

Baker, Kenneth R., "The Effects of Input Control In a Simple Scheduling Model", Journal of Operations Management, Volume 4, No. 2 (1984), pp.99-112.

Baker, Kenneth R., Kanet, J.J., "Job Shop Scheduling With Modified Due Dates", Journal of Operations Management, Volume 4, No.1, (1983), pp.11-22.

Baker, C.T., Dzielinski, B.P., "Simulation of a Simplified Job Shop", Management Science, Volume 6, (1960), pp.311-323.

Bechte, Wolfgang, "Controlling Manufacturing Lead Time and Work-in-Process Inventory by Means of Load-Oriented Order Release", APICS Conference Proceedings, 1982, pp.67-72.

Bechte, Wolfgang, "Load-Oriented Manufacturing Control", APICS Conference Proceedings, 1988a, pp.148-152.

Bechte, Wolfgang, "Theory and Practice of load-oriented Manufacturing Control", International Journal of Production Research, Volume 26, No.3 (1988b), pp.375-395.

Begley, Thomas, Kainen, T.L., Maggard, M.J., "The Impact of Alternative Staffing Configurations on Workunit Performance in the Service Sector", Human Resource Planning, Volume 6, No. 4, (1983)

Belt, Bill, "Integrating Capacity Planning and Capacity Control", Production and Inventory Management, (1976), pp.9-25.

Berry, William, L., Schmitt, T.G., Vollmann, T.E., "Capacity Planning Techniques for Manufacturing Control Systems: Information Requirements and Operational Features", Journal of Operations Management, Volume 3, No. 1, (1982), pp. 13-25.

Bertrand, J.W.M., "The Use of Workload Information to Control Job Lateness in Controlled and Uncontrolled Release Production Systems", Journal of Operations Management, volume 3, No.2 (1983), pp.79-92.

Blackstone, John H., Phillips, D.T., Hogg, G.L., "A state-of-the-art survey of dispatching rules for manufcturing job shop opertions," International journal of production research, Volume 20, No.1, (1982), p.27-45.

Bobrowski, P.M., "Implementing a Loading Heuristic in a Discrete Release Job Shop", International Journal of Production Research, Volume 27, No.11 (1989), pp.1135-1148.

Bobrowski, Paul M., Mabert, V.A., "Alternate Routing Strategies in Batch Manufacturing: An Evaluation", Decision Sciences, Volume 19, No. 4, (1988), pp. 713-733.

Bobrowski, P.M., Park, P.S., "Work Release Strategies in a Dual Resource Constrained Job Shop", OMEGA International Journal of Management Science, Volume 17, NO. 2 (1989), pp.177-188.

Browne, J., Boon, J.E., Davies, B.J., "Job Shop Control", International Journal of Production Research, Volume 19, No. 6, (1981), pp. 633-643.

Browne, J., Davies, B.J., "The Phased Release of Batches Into a Job Shop", International Journal of Operations and Production Management, Volume 4, No. 4, (1984), pp. 16-27.

Bulkin, Michael H., Colley, J.L., Steinhoff, H.W., "Load Forecasting, Priority Sequencing, and Simulation in a Job Shop Control System", Management Science, Volume 13, No.2, (1966), pp.B-29 to B-51.

Christy, David P., Kanet, J.J., "Open Order Rescheduling in Job Shops with Demand Uncertainty: A Simulation Study", Decisions Sciences, Volume 19, No.4, (1988), pp.801-818.

Christy, David P., Kanet, J.J., "Manufacturing Systems with forbidden early shipment: implications for choice of scheduling rules", International Journal of Production Research, Volume 28, No.1, (1990), pp. 91-100.

Conover, W.J., Practical Nonparametric Statistics, John Wiley & Sons Inc., New York, New York, 1971.

Conway, R.W., Maxwwell, W.L., Miller, L.W., Theory of Scheduling, Reading, Mass., Addison-Wesley Publishing, 1967.

Cook, Thomas D., Campbell, D.T., Quasi-Experimentation: Design & Analysis Issues for Field Settings, Houghton Mifflin Company, Boston, Mass., 1979.

Day, James E., Hottenstein, M.P., "Review of Sequencing Research", Naval Research Logistics Quarterly, Volume 17, No. 1, (1970), pp.11-39.

Eilon, Samuel, Chowdhury, I.G., Serghious, S.S., "Experiments with the SI^X Rule in Job-Shop Scheduling", Simulation, Volume 24, (1975), pp. 45-48.

Elvers, Douglas A., Treleven, M.D., "Job-Shop vs. Hybrid Flow-Shop Routing in a Dual Resource Constrained System", Decision Sciences, Volume 16, (1985), pp. 213-222.

Fry, Timothy, Smith, A.E., "A Procedure for Implementing Input/Output Control: A Case Study", Production and Inventory Managment Journal, (1987), pp. 50-52.

Fryer, John S., "Operating Policies in Multiechelon Dual-Constraint Job Shops", Management Science, Volume 19, No.9, (1973), pp.1001-1012.

Fryer, John S., "Labor Flexibility in Multiechelon Dual-Constraint Job Shops", Management Science, Volume 20, No. 7, (1974a), pp. 1073-1080.

Fryer, John S., "Organizational Structure of Dual-constraint Job Shops", Decision Sciences, Volume 5, No. 1, (1974b), pp. 45-57.

Fryer, John S., "Effects of Shop Size and Labor Flexibility in Labor and Machine Limited Production Systems", Management Science, Volume 21, No. 5, (1975), pp. 507-515.

Fryer, John S., "Organizational Segmentation and Labor Transfer Policies in Labor and Machine Limited Production Systems", Decision Sciences, Volume 7, No. 4, (1976), pp.725-738.

Filliben, James, J., "The Probability Plot Correlation Test for Normality", Technometrics, Volume 17, No. 1 (1975), pp.111-117.

Graves, Stephen C., "A Tactical Planning Model for a Job Shop", Operations Research, Volume 34, No. 4, (1986), pp. 522-533.

Gue, F.S., "Input/Output Control In a Job Shop", APICS 1975 Conference Proceedings, (1975), pp. 58-72.

Gunther, Richard E., "Server Transfer Delays in a Dual Resource Constrained Parallel Queing System", Management Science, Volume 25, No.12, (1979), pp.1245-1257.

Gunther, Richard E., "Dual-Resource Parallel Queues with Server Transfer and Information Access Delays", Decision Sciences, Volume 12, No. 1, (1981), pp.97-111.

Harris, Roy D., "An Empirical Investigation of a Jobshop As a Network of Queueing Systems" (Ph.D. dissertation, University of California - Los Angeles, 1965).

Harty, James D., "Controlling Production Capacity", APICS 12th Annual Conference Proceedings, (1969), pp. 60-64.

Hax, Arnoldo C., Candea,D. Production and Inventory Management, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (1984).

Heilweil, Norman L., "Do You Actually Control Your Capacity", APICS Conference Proceedings, (1989), pp. 95-108.

Hochberg,Yosef, Tamhane, A.C., Multiple Comparison Procedures, John Wiley & Sons, Inc. 1987.

Hogg, G.L., Phillips, D.T., Maggard, M.J., "Parallel-Channel, Dual-Resource Constrained Queueing Systems with Heterogeneous Resources", AIIE Transactions, Volume 9, No.4, (1977), pp.352-362.

Hogg, Gary L., Phillips, D.T., Maggard, M.J., Lesso, W.G., "Gerts QR: A Model for Multi-Resource Constrained Queueing Systems Part I: Concepts, Notation, and Examples", AIIE Transactions, Volume 7, No. 2, (1975a), pp. 89-99.

Hogg, Gary L., Phillips, D.T., Maggard, M.J., Lesso, W.G., "Gerts QR: A Model for Multi-Resource Constrained Queueing Systems Part II: An Analysis of Parallel-channel, Dual-Resource Constrained Queueing Systems with Homogeneous Resources", AIIE Transactions, Volume 7, No. 2, (1975b), pp. 100-109.

Holloway, Charles A., Nelson, R.T., "Alternative Formulation of the Job Shop Problem with Due dates", Management Science, Volume 20, No. 1, (1973), pp.65-75.

Holstein, William K., Berry, W.L., "The Labor Assignment Decision: An Application of Work Flow Structure Information", Management Science, Volume 18, No. 7, (1972), pp. 390-400.

Huge, Ernest C., "Managing Manufacturing Lead Times", Harvard Business Review, (1979), pp. 116-123.

Igel,M., "Experiences in Jobshop Planning", International Journal of Production Research, Volume 19, No.2 (1981), pp.153-164.

Irastorza, Joseph C., Deane, R.H., "A Loading and Balancing Methodology for Job Shop Control", AIIE Transactions, Volume 6, No.4 (1974), pp.302-307.

Irastorza, Joseph C., Deane, R.H., "Starve the Shop - Reduce Work-in-Process", Production & Inventory Management, 2nd Quarter, 1976, pp.20-25.

Jackson, James R., "Networks of Waiting Lines", Operations Research, Volume 5, No. 4, (1957), pp. 518-521.

Jackson, James R., "Jobshop-Like Queueing Systems", Management Science, Volume 10, No. 1, (1963), pp. 131-142.

Jones, C.H., "An Economic Evaluation of Job Shop Dispathcing Rules," Management Science, Volume 20, No.3 (1973), pp.293-307.

Kanet, J.J, Christy, D.P., "Manufacturing Systems with Forbidden Early Order Departure", International Journal of Production Research, Volume 22, No.1, (1984), pp. 41-50.

Kanet, John J., "Load-Limited Order Release In Job Shop Scheduling Systems", Journal of Operations Mangement, Volume 7, No. 3, (1988), pp.44-58

Karni, Reuven, "Capacity Requirements Planning - A Systematization", International Journal Production Research, Volume 20, No. 6, (1982), pp. 715-739.

Kerlinger, Fred N., Foundations of Behavioral Research, 3rd edition, Holt, Rinehart and Winston, Inc., 1986.

Kingsman,B.G., Tatsiopoulos,I.P., Hendry,L.C., "A Structural Methodology for Managing Manufacturing Lead Times in Make-to-Order Companies", European Journal of Operational Research, Volume 40, (1989), pp.196-209.

Kirk,Roger E., Experimental Design: Procedures for the Behavioral Sciences, 2nd edition, Wadsworth, Inc. Belmont CA, 1972.

Kleijnen, Jack P.C., Statistical Techniques in Simulation: Part 1, Marcel Dekker, Inc., New York, New York, 1974.

Kleijnen, Jack,P.C., Statistical Tools for Simulation Practitioners, Marcel Dekker, Inc., New York and Basel 1987.

Klockars, Alan J., Sax, G., Multiple Comparisons, Sage Publications, Beverly Hills, 1986.

Krajewski, Lee, J., Ritzman, L.P., Operations Management, Addison-Wesley Publishing Co., Reading, Mass., 1990.

Kuester, James L., Mize, J.H., Optimization Techniques with Fortran, McGraw-Hill Book Company, New York (1973).

Law, Averill,M., Carson, J.S., "A Sequential Procedure for Determining the Length of a Steady-State Simulation", Operations Research, Volume 27, No.5, (1979), pp.1011-1025.

Law, Averill, M., Kelton, W.D., Simulation Modeling and Analysis, McGraw-Hill Book Company, New York, New York, 1982.

LeGrande, Earl, "The Development of A Factory Simulation System Using Actual Operating Data", Readings in Production and Operations Management, ed. Elwood S. Buffa, John Wiley and Sons, New York, New York, 1966, pp.132-154.

Lewis, P.A.W., Orav, E.J., Simulation Methodology for Statisticians, Operations Analysts, and Engineers, Volume 1, Wadsworth & Brooks, Pacific Grove, CA, 1989.

Lipsey, Mark W., Design Sensitivity: Statistical Power for Experimental Research, Sage Publictions, Beverly Hills, 1990.

Maggard, Michael J., Lesso, W.G., Hogg, G.L., Phillips, D.T., "GERTS III QR: A Multiple Resource Constrained Network Simulation Model", Management Datamatics, Volume 5, No.1, (1976), pp.5-14.

Maggard, Michael J., Lesso, W.G., Keating, R.J., Wexler, M.J., "Network Analysis with GERTS III QR", Industrial Engineering, Volume 6, No.5, (1974), pp.24-29.

Maggard, Michael J., Hogg, G.L., Phillips, D.T., "The Efficiency and Economics of Small Heterogeneous labour Limited Queueing Systems", International Journal of Production Research, No. 5, (1980), pp.619-636.

Makridakis,Spyros, Wheelwright, S.C., McGee,V.E., Forecasting: Methods and Applications, John Wiley & Sons, New York, 1983.

McKay, Kenneth N., Safayeni, F.R., Buzacott, J.A., "Job-Shop Scheduling Theory: What is Relevant?", Interfaces, Volume 18, No. 4, (1988), pp. 84-90.

Melnyk, Steven A., Carter, P.L., Production Activity Control, Dow Jones-Irwin, Homewood, Illinois, 1987.

Melnyk, Steven A., Carter, P.L., Dilts, D.M., Lyth, D.M., Shop Floor Control, Dow Jones-Irwin, Homewood, Illinois, 1985.

Melnyk, Steven A., Ragatz, G.L., "Order Review/Release and its Impact on the Shop Floor", Production and Inventory Management Journal, Volume 29, No. 2, (1988), pp. 13-17.

Melnyk, Steven A., Ragatz, G.L., "Order review/release: research issues and perspectives", International Journal of Production Research, Volume 27, No.7 (1989), pp.1081-1096.

Melnyk, Steven A., Ragatz, G.L., Fredendall, L.D., "Load Smoothing By the Planning and Order Review/Release Systems: A Simulation Experiment", submitted 1991 to Journal of Operations Management.

Melnyk, Steven A., Vickery, S.K., Carter, P.L., "Scheduling, Sequencing, And Dispatching: Alternative Perspectives", Production and Inventory Management, (1986), pp. 58-68.

Mihram, G.Arthur, "Blocking in Simular Experimental Designs", Journal Statistical Computing and Simulation, voluume 3 (1974), pp.29-32.

Nelson, Rosser T., "Labor Assignment As a Dynamic Control Problem", Operations Research, Volume 14, No. 3, (1966), pp.369-376.

Nelson, Rosser T., "Labor and Machine Limited Production Systems", Management Science, Volume 13, No. 9, (1967), pp. 648-671.

Nelson, Rosser T., "Dual-Resource Constrained Series Service Systems", Operations Research, Volume 16, No. 2, (1968), pp.324-341.

Nelson, Rosser T., "A Simulation of Labor Efficiency and Centralized Assignment in a Production Model", Management Science, Volume 17, No. 2, (1970), pp. B-97-B106.

Neter, John, Wasserman, W., Kutner, M.H., Applied Linear Statistical Models, 2nd edition, Irwin, Homewood, Illinois, 1985.

Nicholson, T.A.J., Pullen, R.D., "A Practical Control System for Optimizing Production Schedules", International Journal Production Research, Volume 9, No. 2, (1971), pp. 219-227.

O'Grady, P.J., Azoza, M.A., "An Adaptive Approach to Shop Loading", Omega International Journal of Management Science, Volume 15, No. 2, (1987), pp. 121-128.

O'Grady, P.J., Byrne, M.D., "A Combined Switching Algorithm and Linear Decision Rule Approach to Production Planning", International Journal of Production Research, Volume 23, No. 2, (1985), pp.285-296).

Onur, Levent, Fabrycky, W.J., "An Input/Output Control system for the Dynamic Job Shop", IIE Transactions, (1987), pp. 88-97.

Orlicky, Joseph, Material Requirements Planning, McGraw-Hill Book Company, New York, New York 1975.

Panico, Joseph A., Queuing Theory, Prentice-Hall, Inc., Englewood Cliffs, N.J. 1969.

Panwalker,S.S., Iskander, W., "A Survey of Scheduling Rules", Operations Research, Volume 25, No. 1, (1977), pp.45-61.

Park, Sungchil, The Effects of Input Control In A Dual Resource Constrained Job Shop, University Microfilms International, Ann Arbor, Michigan (1987).

Park, Paul S., Bobrowski, P.M., "Job Release and Labor Flexibility in a Dual Resource Constrained Job Shop", Accepted for publication in the Journal of Operations Management, 1989

Pedhazur, Elazar J., Multiple Regression in Behavioral Research, CBS College Publishing, 1982.

Pegden,C.,Dennis, Introduction to SIMAN, Systems Modeling Corporation, State College, Pennsylvania, 1987.

Pegden, C. Dennis, Shannon, Robert, E., Sadowski, Randall P., Introduction to Simulation Using SIMAN, McGraw-Hill, New York, New York (1990)

Plossl, George W., Wight, O.W., "Capacity Planning and Control", Production & Inventory Management, Volume 14, No. 3 (1973), pp.31-67.

Ragatz, Gary L., An Evaluation of Order Release Mechanisms for Job Shops, University Microfilms International, Ann Arbor Michigan (1985).

Ragatz, Gary L., Mabert, V.A., "An Evaluation of Order Release Mechanisms in a Job-Shop Environment", Decision Sciences, Volume 19, No.1, (1988), pp. 167-189.

Reiter, Stanley, "A System for Managing Job-Shop Production", The Journal of Business, Volume 39, No. 3, (1966), pp.371-393.

Rochette, Rene, Sadowski, R.P., "A Statistical Comparison of the Performance of Simple Dispatching Rules for a Particular Set of Job Shops", International Journal of Production Research, Volume 14, No. 1, (1976), pp. 63-75.

Rowe, A.J., "Toward A Theory of Scheduling", The Journal of Industrial Engineering, Volume 11, No. 2, (1960), pp.125-136.

Russell, Roberta, S., Taylor, B.W., "An Evaluation of Sequencing Rules For an Assembly Shop", Decision Sciences, Volume 16, No. 2 (1985), pp. 196-212.

Ryan, Thomas A. Jr., Joiner, B.L., Ryan, B.F., Minitab, Minitab Project, The Pennsylvania State University, University Park, PA (1982).

Sandman, Wm.,E., Hayes, J.P., How to Win Productivity in Manufacturing, AMACOM, A Division of American Management Associations, pp.65-77, 1982.

SAS Institute Inc., SAS/STAT User's Guide, Release 6.03 Edition, SAS Institute Inc., Cary North Carolina, 1988.

SAS Institute Inc., SAS Procedures Guide, Release 6.03 Edition, SAS Institute Inc., Cary North Carolina, 1988.

Scheffe, Henry, The Analysis of Variance, John Wiley & Sons, Inc., New York, New York, 1959.

Schruben, L., Singh, H., Tierney, L., "Optimal Tests for Initialization Bias in Simulation Output", Operations Research, Volume 31, No. 6 (1983),pp.1167-1178.

Shimoyashiro, Sadao, Isoda, K., Awane, H., "Input Scheduling and Load Balance Control for a Job Shop", International Journal of Production Research, Volume 22, No.4 (1984), pp.597-605.

Steinhoff, Harry W., "Daily System for Sequencing Orders in a Large Scale Job Shop", Readings in Production Management and Operations, ed. Elwood S. Buffa, New York, New York, 1966.

Takacs, Lajos "Two Queues Attended by a Single Server", Operations Research, Volume 16, No. 3 (1968), pp. 639-650.

Tatsiopoulos, I.P., "Some Aspects of the Input/output Methods For Managing Work-in-Process Inventories", Engineering Costs and Production Economics, Volume 15, (1988), 235-239.

Tatsiopoulos, I.P., Kingsman, B.G., "Lead Time Management", European Journal of Operational Research, Volume 14, (1983), pp. 351-358.

Treleven, Mark D., Elvers, D.A., "An Investigation of Labor Assignment Rules in a Dual-Constrained Job Shop", Journal of Operations Management, Volume 6, No. 1, (1985), pp.51-68.

Treleven, Mark, "A Review of the Dual Resource Constrained System Research", IIE Transactions, Volume 21, No. 3, (1989), pp.279-287.

Treleven, Mark, "The Timing of Labor Transfers in Dual Resource-constrained Systems: "Push" vs. "Pull" Rules", Decision Sciences, Volume 18, (1987), pp. 73-88.

Turban, Efraim, Meredith, J.R., Fundamentals of Management Science, Business Publications Inc., Dallas, Texas, 1977.

Umble, M. Michael, Srikanth, M.L., Synchronous Manufacturing: Principles for World Class Excellence, South-Western Publishing Co, Cincinnati, Ohio, 1990.

Vollmann, Thomas E., Berry, W.L., Whybark, D.C., Manufacturing Planning and Control Systems, Dow Jones-Irwin, Homewood, Ilinois, 1988.

Weeks, James K., "A Simulation Study of Predictable Due-Dates", Management Science, Volume 25, No.4, April 1979, pp. 363-373.

Weeks, James, K., Fryer, J.S., "A Simulation Study of Operating Policies in a Hypothetical Dual-Constrained Job Shop", Management Science, Volume 22, No. 12, (1976), pp. 1362-1371.

Weeks, James, K., Fryer, J.S., "A Methodology for Assigning Minimum Cost Due-Dates", Managment Science, Volume 23, No. 8, (1977), pp.872-881.

Weeks, James K., Wingler, T.R., "A Stochastic Dominance Ordering of Scheduling Rules", Decision Sciences, Volume 10, (1979), pp. 245-257.

Wight, Oliver, "Input/Output Control A Real Handle On Lead Time", Production and Inventory Management, No.3, (1970), pp.9-30.

Wilkinson, Leland, SYGRAPH: The System for Graphics for the PC, SYSTAT Inc., 1987.