AN INVESTIGATION OF GROUP SCHEDULING HEURISTICS IN A FLOW SHOP CELLULAR SYSTEM WITH WORKCENTER SHARING FOR THE FORBIDDEN EARLY SHIPMENT ENVIRONMENT

By

BOR-YUH LEU

Bachelor of Science
National Cheng Kung University
Taiwan, Republic of China
1981

Master of Science
National Cheng Kung University
Taiwan, Republic of China
1983

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1994

OKLAHOMA STATE UNIVERSITY

AN INVESTIGATION OF GROUP SCHEDULING HEURISTICS IN A FLOW SHOP CELLULAR SYSTEM WITH WORKCENTER SHARING FOR THE FORBIDDEN EARLY SHIPMENT ENVIRONMENT

Thesis Approved:

John W. Nazemetz
Thesis Advisor

David J. Weeks

David B. Pratt

Manjunath Kamath

Timothy J. Greene

Thomas C. Collins
Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my gratitude and appreciation to the members of my dissertation committee, each of whom generously rendered their professional assistance: Dr. John W. Nazemetz (chairman), Dr. Timothy J. Greene, Dr. Manjunath Kamath, Dr. David B. Pratt, and Dr. David L. Weeks. To them I attribute all the merit which this work possesses.

Special thanks go to my advisor, Dr. Nazemetz, for his guidance, advice, and encouragement throughout this research and my doctoral study.

I thank the School of Industrial Engineering and Management for providing me the opportunity and support to receive higher education. To those faculty and staff in the School of IE&M, I am sincerely grateful.

Finally, I would like to dedicate this dissertation to my parents, Chi-Chung and Chang-Yen Leu, my wife, Tsao-Jean Leu, and our children, Chia-Luen and Chia-Tien, for their continuous sacrifice, support, encouragement, and love.

TABLE OF CONTENTS

---

* Available from School of Industrial Engineering and Management at the Oklahoma State University

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A_i	Allowance of order i (which refers to the amount of time budgeted to complete order i)
C_i	Completion time of order i
CR_i	Critical ratio of job i
CR_{mqi}	Critical ratio of job i within queue q at workcenter m
D_i	Due date of order i
f	Profit margin
h	Annual out-of-pocket holding cost rate
I_{mq}	Empty queue index for queue q at workcenter m
J_{mq}	Lengthy queue index for queue q at workcenter m
K	Order allowance level
M_i	Number of operations of order i
N	Total number of orders shipped
N_e	Total number of early orders shipped
N_j	Total number of jobs processed
N_o	Total number of on time orders shipped
N_t	Total number of tardy orders shipped
N_{qc}	Number of jobs in queue q used to calculate ACR
N_{qd}	Number of jobs in queue q used to calculate ADD
N_{qs}	Number of jobs in queue q used to calculate ASLK
NB_{mq}	Number of jobs within queue q at workcenter m

NPV_i	Net present value of order i
P_i	Remaining processing time of order i
P_{ij}	Processing time of operation j of order (or job) i
r	Annual interest rate (continuous compounding)
R_i	Undiscounted revenue of order i
RT_i	Release time of order i
S_i	Slack of job i at time t
S_{mqi}	Slack of job i within queue q at workcenter m
t_{ij}	Time when operation j of order i is started
T_i	Shipping time of order i
U_{ij}	Labor processing charge for operation j of order i
V_{ij}	Labor setup charge for operation j of order i
W_i	Material cost of order i
WT_j	Queue waiting time of job j
Y_{ijkmn}	Mean value of performance measure Y with i-th heuristic, j-th level of average annual workload, k-th level of demand pattern variability, m-th level of cell transfer batch, and n-th replication
α	Significance level
β	Absolute error
ε	Random effect
μ	Common effect (or global mean)
π	Tardiness penalty cost rate (percentage of order revenue per year)
ϕ	Set of uncompleted operations
ϕ_1	Set of workcenters ahead
ϕ_2	Set of workcenters behind

CHAPTER I

INTRODUCTION

This research proposes an investigation into group scheduling heuristics in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment. The cellular system consists of two flow shop cells; each of the flow shop cells has five workcenters, with the last workcenter shared between cells. For the forbidden early shipment environment, orders cannot leave the system earlier than the customer has specified. The shop factors impacting the performance of the cellular system have been identified. Five group scheduling heuristics have been developed and then evaluated by computer simulation under different shop conditions. Ten performance measures, which include an economically based measure and other time-based or inventory-based measures, have been selected to collect statistics.

Group scheduling heuristics, also known as group technology scheduling heuristics or family heuristics, attempt to serially process similar jobs and eliminate major setups. Group Technology (GT) can be defined as bringing together and organizing (grouping) common concepts, principles, problems, and tasks (technology) (Greene and Sadowski, 1984). Group technology, which has been gradually adopted as a manufacturing strategy in industry, offers some distinct advantages when compared to a traditional job shop production system. Reduced throughput and material handling times, decreased work-in-process and finished goods inventories, and increased flexibility to handle forecast errors are some of the major advantages mentioned by practicing users (Mosier and Taube, 1985).

Cellular Manufacturing (CM), one of the applications of group technology, is the physical division of the manufacturing facilities machinery into production cells (Burbidge, 1975). Each cell is designed to produce a part family. A part family is defined as a set of parts that require similar machinery, tooling, machine operations, and/or jigs and fixtures (Burbidge, 1971). The advantages associated with cellular manufacturing include reduced material handling, reduced tooling, reduced setup time, reduced expediting, reduced work-in-process inventory, reduced part makespan, improved human relations, improved operator expertise, and better quality. Possible disadvantages are increased capital investment, reduced shop flexibility, and lower machine utilization (Greene and Sadowski, 1984, Wemmerlov and Hyer, 1989). Cellular Manufacturing Systems (CMS) are generally differentiated from flexible manufacturing systems in that they usually involve some manual operations, that is, they are not fully automated.

Scheduling using group technology concepts, regardless of the system's physical layout, is called Group Scheduling (GS). One of the reasons group scheduling studies have received considerable attention recently is that group scheduling heuristics can maximize the advantages of cellular manufacturing by further reducing the overall machine setup time. Another reason is that group scheduling heuristics can reduce the disadvantage of cellular manufacturing (i.e., inflexibility of shop) by employing a diverse range of part subfamilies to increase shop flexibility (Mahmoodi et al., 1990b, Lee, 1985).

Classifications of Group Scheduling

Group scheduling can be classified into four categories based upon the system's physical layout: group scheduling in a single machine layout, group scheduling in a line (or product) layout, group scheduling in a functional (or process) layout, and group scheduling in a cellular (or GT) layout. The classifications of group scheduling are shown in Figure 1.1. Each category is defined and discussed in some detail below.

Group scheduling in a single machine layout assumes specific subfamilies are routed through only one machine. It also assumes that the subfamilies have been formed on

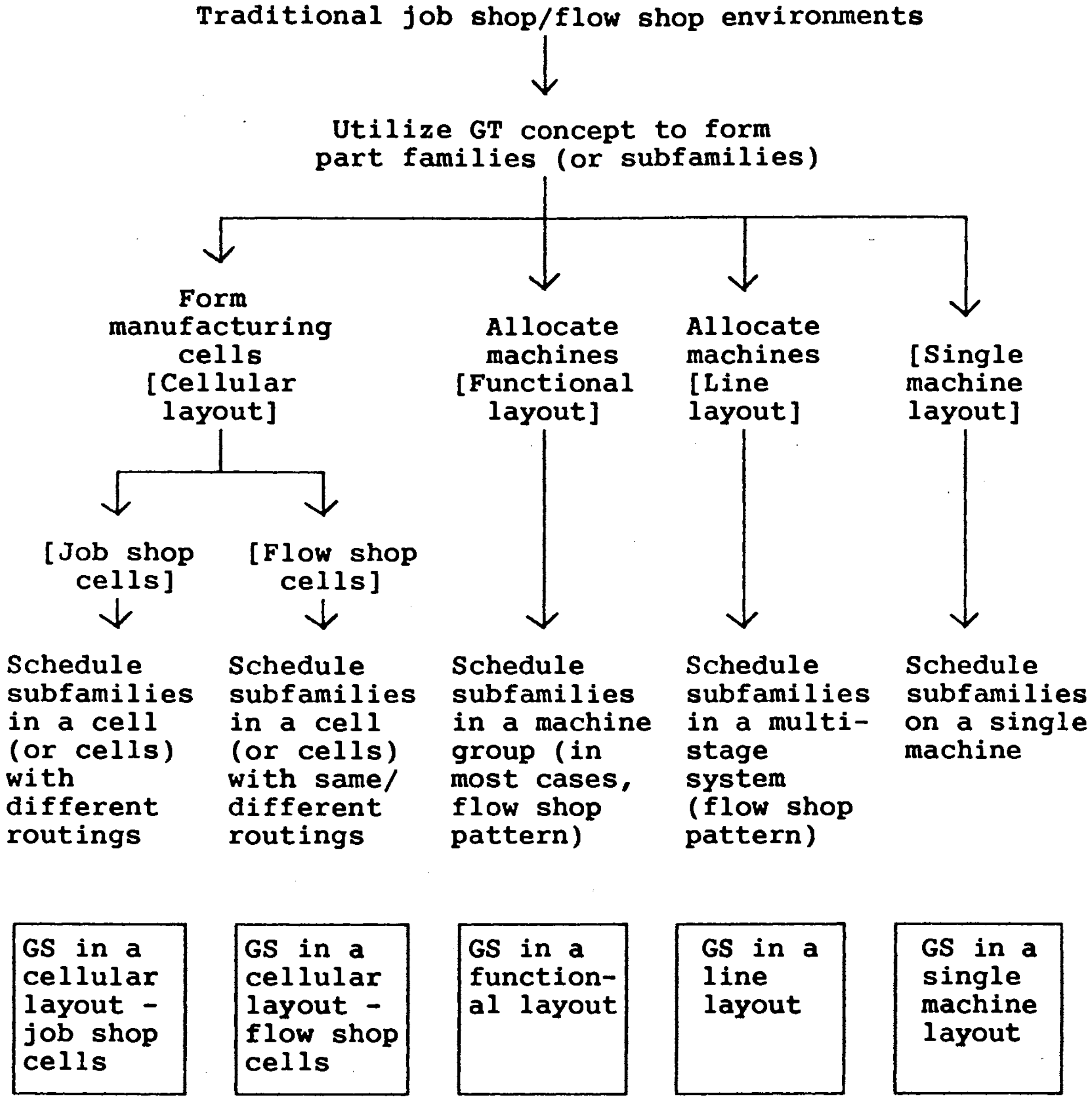


Figure 1.1 Classifications of Group Scheduling

the basis of group technology. A subfamily is a grouping of part types with similar setups (i.e., no significant amount of setup required between part types.) A part family can contain several subfamilies. Another assumption that is often made is that the subfamily setups are sequence-independent. Optimizing (or heuristic) algorithms for determining both the optimal (or near optimal) group (i.e., subfamily) sequence and job sequence in each group have been developed to minimize performance measures such as total tardiness and makespan. Parts to be made are typically called "jobs" and jobs are classified into several "groups" (i.e., subfamilies).

One example is a simulation study conducted by Wemmerlov (1989) to examine the performance of different heuristics in a single machine layout. Two single-stage dispatching rules and two two-stage family heuristics were examined. The dispatching rules used were first-come-first-served (FCFS) and shortest processing time (SPT). The family heuristics used were FCFCFS (both FCFS queue selection and job dispatching rules) and FCSPT (FCFS queue selection rule and SPT job dispatching rule). The results showed that the two-stage family heuristics generally outperformed the single-stage dispatching rules.

Group scheduling in a line layout is equivalent to scheduling specific subfamilies in a multistage manufacturing system (flow shop) assuming that the subfamilies have been formed on the basis of group technology. It may or may not need to allocate (or group logically) machines for specific subfamilies. Other important assumptions made in this category include: fixed workcenter sequences for all jobs, sequence-independent setups, and a machine constrained system.

Optimizing (or heuristic) algorithms for determining both the optimal (or near optimal) group sequence and job sequence within a group have been developed. For optimizing algorithms, branch-and-bound methods can be applied to the problem and minimize performance measures such as total tardiness and makespan. For heuristic algorithms, they can be computerized (or simulated) to determine the near optimal solutions (or performance of heuristics) with respect to specific performance measures. Some examples of the research in this category are Petrov (1968), Hitomi and Ham (1976 and 1977), Ham et al. (1979), Cho (1982), Cho et al. (1982), and Moily and Stinson (1989).

Group scheduling in a functional layout includes two steps. The first step is to form subfamilies on the basis of group technology and then to allocate (or group logically) machines for specific subfamilies based on the parts' machining characteristics in a traditional job shop environment. Due to the limited number of certain machines which are needed by several subfamilies, these machines (shared or key machines) are allocated to more than one machine group. This results in the necessity to determine the group (i.e., subfamily) sequence for processing at the key machines (Radharamanan, 1986).

In the second step, optimizing (or heuristic) algorithms for determining both the optimal (or near optimal) group and job sequence in each group are developed to optimize a performance measure such as total tardiness or makespan. In this category, setups are assumed sequence-independent and the scheduling for all jobs within a subfamily exhibits a flow shop pattern. Some examples of the research in this category are Sundaram (1982 and 1983), and Radharamanan (1986).

Group scheduling in a cellular layout is related to the development of new group scheduling heuristics (or family heuristics), and the scheduling of subfamilies using the developed heuristics in cellular manufacturing. It is assumed that, based on group technology, the formation of part subfamilies and manufacturing cells has been done. In most cases, computer simulation has been utilized to compare the performance of different heuristics including existing and newly developed heuristics.

Group scheduling in a cellular layout can further be divided into two sub-categories: group scheduling in job shop cells and group scheduling in flow shop cells (also known as flow line cells or flow-through cells). In flow shop cells, all parts must follow the workcenter sequences, but the routings may be different within a cell. For the case of simple flow shop cells, all parts have identical routings within a cell. In job shop cells, parts may arrive to and depart from different workcenters and have different routings within a cell.

One example of group scheduling research in single flow shop cells is a recent study done by Mahmoodi et al. (1992). The authors conducted a simulation experiment to compare the performance of two dispatching rules and four group scheduling heuristics in a simple flow shop cell. The flow shop cell consisted of five workcenters each containing one machine. Jobs had to enter the cell from the first workcenter and exit from the last workcenter. The published research on group scheduling in flow shop cells will be reviewed in Chapter II.

One example of group scheduling research in job shop cells is a recent paper written by Ruben et al. (1993). The authors conducted a simulation experiment to compare the performance of three dispatching rules and five group scheduling heuristics in a job shop cell. The job shop cell consisted of five workcenters each containing one machine. Jobs could enter the cell from workcenter 1 or 2 and exit from workcenter 4 or 5. Jobs' routings consisting of between three and five workcenters were dependent on their part types. Other recently published research on group scheduling in job shop cells includes Mahmoodi and Dooley (1991), Mahmoodi et al. (1990a and 1990b), and Sassani (1990), etc.

Forbidden Early Shipment

Kanet and Christy (1984) have identified Forbidden Early Shipment (FES) as a prevalent environment in real-world manufacturing systems. In such environments, orders cannot leave the system earlier than the customer has specified so that the firm has an incentive to complete the order as close to its due date as possible (so as to avoid the unnecessary inventory carrying cost for this finished order).

Various scheduling rules have been examined on a single machine (e.g., Lawrence, 1991) and in job shops or flow shops (e.g., Christy and Kanet, 1990, Scudder et al., 1990, Rohleder and Scudder, 1992) for the forbidden early shipment environment. The results of these studies show that the preferable scheduling rules where early shipments were forbidden were different from those where early shipments were allowed.

For example, Scudder et al. (1990) utilized computer simulation to model a job shop for the forbidden early shipment environment. The results showed that CR (critical ratio) provided higher average net present value than the other three rules, i.e., OPCR (operation critical ratio), PRF/OPT (profit per operation), and VLADRAT (value added ratio). In an earlier study done by Scudder and Smith-Daniels (1989), they evaluated the same rules which were used by Scudder et al. (1990) in a job shop where early shipments were allowed. The results showed that VLADRAT (value added ratio) outperformed the other three rules with respect to the average net present value.

Problem Statement

A survey study done by Wemmerlov and Hyer (1989) showed that most cells implemented in industry were flow shop cells (or close to flow shop cells) and 20% of the companies with manned cells and 14% of those with unmanned cells reported that machines were shared between cells. While some studies in flow shop cells have been done, all studies related to dynamic scheduling in flow shop cell environments only consider a single flow shop cell which had five workcenters and identical routings for all orders. No work has been done to identify the important factors and to investigate the performance of various group scheduling heuristics in flow shop cells with workcenters shared between cells and different routings and order sizes for different part types.

Although just-in-time concepts are gaining popularity in industry, no work has been done to investigate the performance of various group scheduling heuristics in cellular manufacturing systems with the features such as forbidden early shipment and just-in-time delivery of materials.

In the published research related to cellular manufacturing, no monetary performance measure (such as net present value) has been applied to evaluate the performance of various group scheduling heuristics, although profit-maximization is as important as other performance measures such as flow time (measuring efficiency) and percent tardy (measuring effectiveness).

In short, the problem statement for this research can be summarized as:

> To understand the performance of group scheduling heuristics under various shop conditions with respect to different measures in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment.

Organization of the Dissertation

This research is described in detail in the following five chapters. Chapter II reviews previous research on group scheduling in flow shop cellular systems and on manufacturing systems with forbidden early shipment. Chapter III presents the goal, objectives, and scope of the research. Chapter IV discusses the research methodology used in this research effort. Chapter V contains the analysis and interpretation of the simulation results. Finally, this research effort is summarized, the contributions of the research are listed, and the recommendations for further research are offered in Chapter VI.

CHAPTER II

LITERATURE REVIEW

Introduction

This chapter presents a comprehensive review of relevant research. The applicable literature is divided into two categories. The first category is the research on group scheduling in flow shop cellular systems. The second category is the research on manufacturing systems with forbidden early shipment.

Research on Group Scheduling in Flow Shop Cellular Systems*

The research on group scheduling in cellular manufacturing has received considerable attention during the last few years. Because of the limitations of developing and implementing analytical or optimizing techniques, most researchers have proposed heuristics as a

---

* The material in this section is based on a working paper done by Leu, Greene, and Nazemetz (1992).

solution to group scheduling problems (Mahmoodi et al., 1992). Furthermore, most researchers have utilized computer simulation to compare the performance of different heuristics under various shop floor conditions.

This section reviews the major published research efforts concerned with group scheduling in flow shop cellular systems. Table 2.1 shows the summaries of published research on group scheduling in flow shop cellular systems. This table is adopted from a working paper done by Leu et al. (1992). The purpose of this table is to give a comparison of the published research. The key items include the major assumptions, experimental factors, performance measures, shop model, heuristics tested, and study methodology. Each study included in the table is discussed in some detail below.

Vakharia and Chang (1990) proposed two family heuristics based on simulated annealing in flow shop environments. Simulated annealing is a randomized local search method that was used to derive near-optimal solutions for computationally complex optimization problems. These two family heuristics were compared to a branch and bound procedure (developed by Hitomi and Ham in 1976) and two other family heuristics, i.e., CDS-F (the family version of the CDS procedure developed by Campbell, Dudek, and Smith in 1970) and NEH-F (the family version of the NEH procedure developed by Nawaz, Enscore, and Ham in 1983). The CDS procedure is a static scheduling procedure

TABLE 2.1

SUMMARY OF PUBLISHED RESEARCH ON GROUP SCHEDULING IN FLOW SHOP CELLULAR SYSTEMS

Research	Major Assumptions	Experimental Factors (Levels)	Performance Measures	Shop Model	Heuristics Tested
1. Vakharia and Chang (1990) [COMP]	A5,B2 C1,M1 P4,S1 T1	Problem size (5) Parameters for setup time distribution (3)	Relative error rate Average computation time	A flow shop cell - 3 - 10 WC, 3 - 10 SF, 3 - 10 PT per SF	2-stage: CDS NEH IPF SAH SAH1
2. Wemmerlov and Vakharia (1991) [SIMU]	A1,B2 C1,M1 P3,S1 T1	Shop load (2) Setup time to run time ratio (2) # of subfamilies (2)	Flow time Ratio of early to late jobs	A flow shop cell - 5 WC, 3/6 SF, 4/5/6 PT per SF	1-stage: FCFS,SLK CDS,NEH 2-stage: FCFS-F SLK/PT-F CDS-F NEH-F
3. Russell and Philipoom (1991) [SIMU]	A1,B2 C1,M1 P4,S1 T1	Due date rule (4) Setup time to run time ratio (2)	Flow time Tardiness Root mean square of tardiness	A flow shop cell - 5 WC, 5 SF, 5 PT per SF	(Phase 2) 2-stage: FE-FCFS/SLK FE-APT/SPT SAW-FCFS/SLK SAW-APT/SPT EDD/T SLK/T
4. Mahmoodi, Tierney, and Mosier (1992) [SIMU]	A1/A4 B1,C1 M1,P3 S2,T1	Shop load (2) Setup time to run time ratio (2) Due date tightness (2) Interarrival time distribution (2)	Flow time Tardiness Percent tardy	A flow shop cell - 5 WC, 3 SF, 5 PT per SF	1-stage: FCFS SPT 2-stage: FCFCFS MSSPT DDSI ECSI

TABLE 2.1 (Continued)

Notations for major assumptions:

A1 (A2,A3,A4,A5): Exponential (fifth order Erlang, normal, uniform, deterministic) interarrival times
B1 (B2): Without (with) key machines
C1 (C2): Machine (machine and worker) constrained
M1 (M2): Without (with) machine breakdowns
P1 (P2,P3,P4,P5): Exponential (third order Erlang, normal, uniform, deterministic) processing times
S1 (S2): Sequence independent (dependent) setups
T1 (T2): Transportation times or costs neglected (considered)

Notations for research methodology:

[SIMU]: Simulation
[COMP]: Computation

Notations for shop model:

WC: Workcenters
MA: Machines
SF: Subfamilies
PT: Part Types

that collapses a K-stage flow shop into (K-1) 2-stage problems. While, the NEH procedure is a static scheduling procedure which builds on the creation of successively larger job sequences by entering a new job in all possible positions without disturbing the job order in the previous, partial sequence.

In Vakharia and Chang's study, the shop model was a set of flow shop cells with different number of machines (ranging from three to ten) for different problem sizes. The results revealed that all the family heuristics provided comparable solutions to the optimal procedure for small problems. However, when the problem size increased, their proposed family heuristics outperformed the CDS-F and the NEH-F family heuristics in both solution quality and computation time.

Wemmerlov and Vakharia (1991) conducted an experimental design to compare the performance of four single-stage dispatching rules and four two-stage family heuristics. The dispatching rules used were FCFS (first-come-first-served), SLK (slack), CDS (which collapses a K-stage flow shop into (K-1) 2-stage problems), and NEH (which builds on the creation of successively larger job sequences by entering a new job in all possible positions without disturbing the job order in the previous, partial sequence). The family heuristics used were FCFS-F (both first-come-first-served queue selection and job dispatching rules), SLK/PT-F (which uses either slack or processing time information depending on job status), CDS-F (the family version of the CDS procedure), and NEH-F (the family version of the NEH procedure). Among the four family heuristics tested, SLK/PT-F was the only non-exhaustive heuristic.

Wemmerlov and Vakharia's study utilized computer simulation to model a flow shop cell which consisted of five workcenters. It could be concluded that, for the scheduling heuristics and conditions used in this study, two-stage family heuristics can generate marked improvements with respect to flow time and lateness-oriented measures. Among these, FCFS-F (both first-come-first-served queue selection and job dispatching rules) was the best overall family heuristic. Also, the static scheduling procedures, as a group, were not always better than dispatching rules in the context of stochastic, intermittent scheduling.

Russell and Philipoom (1991) conducted an experimental design to investigate due date setting procedures and dispatching decisions in a 5-workcenter flow shop cell. This study included two phases. The first phase was to decide which next family and next job rules perform well for the family exhaustion and Sawicki truncation rules (Sawicki, 1973). In the second phase, the best next family rule and best next job rule for family exhaustion and truncation rules were combined into four two-stage family heuristics. The four family heuristics were compared with two other truncation rules, i.e., EDD/T (the truncated version of the EDD/CE developed by Ragatz and Carter in 1988) and SLK/T (which uses either slack or processing time information depending on job status).

In Russell and Philipoom's study, five next family rules, four next job rules, and four due date rules were considered. The next family rules used were TWK (total work content), FCFS (first-come-first-served), EDD (earliest due date), CYC (cyclical), and APT (average processing time). The next job rules used were FCFS (first-come-first-served), EDD (earliest due date), SLK (slack), and SPT (shortest processing time). The due date rules used were TWK (total work content), CON (constant), RAN (random), and SEQ (number of switches). Using TWK due date rule, due date offsets are determined by multiplying the work content of a job by a constant allowance factor. The CON due date rule assigns each job entering shop the same due date offset of 600 hours. The RAN due date rule assigns a due date offset from a uniform distribution between 300 and 900 hours. Using SEQ, due date offsets are determined on the basis of how many family switches will take place before the arriving job will be processed.

The results of Russell and Philipoom's study showed that the due date setting procedure had a major impact on how dispatching should be performed in the shop. The APT/SPT family heuristic (APT next family rule and SPT next job rule) was the best performer for mean flow time. When setup times were long, the SEQ due date rule using the family exhaustion procedure with FCFS/EDD (FCFS next family rule and EDD next job rule) family heuristic performed well, while the EDD/T (the truncated version of the EDD/CE developed by Ragatz and Carter in 1988) performed well for short setup times.

Mahmoodi, Tierney, and Mosier (1992) conducted an experiment to compare the performance of two single-stage dispatching rules and four two-stage exhaustive, family heuristics that had exhibited superior performance in previous studies in a flow shop cell environment. The dispatching rules used were first-come-first-served (FCFS) and shortest processing time (SPT). The family heuristics used were MSSPT (MSFAM queue selection rule and SPT job dispatching rule), DDSI (DDFAM queue selection rule and SI* job dispatching rule), ECSI (ECON queue selection rule and SI* job dispatching rule), and FCFCFS (both FCFS queue selection and job dispatching rules). This comparative study utilized computer simulation to model a flow shop cell.

The flow shop cell used in Mahmoodi, Tierney, and Mosier's study consisted of five workcenters each containing one machine. The results indicated that, in general, two-stage heuristics outperformed single-stage rules under all shop floor conditions, as well as being relatively insensitive to changing shop floor conditions. Among the two-stage heuristics, DDSI (DDFAM queue selection rule and SI* job dispatching rule) was the best performing heuristic. Also, the results showed that interarrival time distributions had a major impact on the performance of scheduling heuristics.

Summary

From the published research on group scheduling in flow shop cellular systems as reviewed above, some important conclusions can be drawn:

(1) Few studies related to group scheduling in flow shop cellular systems have been done, although many studies have been published studying job shop cellular systems.

(2) Among the four studies reviewed above, one (Vakharia and Chang, 1990) used static scheduling in several shop configurations and the other three considered dynamic scheduling in a single flow shop cell which had five workcenters and identical routings for all jobs.

(3) Most researchers have proposed heuristics as a solution to group scheduling problems. Overall, (two-stage) group scheduling heuristics performed better than (single-stage) dispatching rules. The choice of dispatching rules or group scheduling heuristics depended on the system definition (including shop model, job characteristics, and assumptions made), shop floor conditions, and performance measures used, and no

heuristic/rule performed better than others in all cases (i.e., in different systems, shop floor conditions, or performance measures, etc.)

(4) Most researchers utilized computer simulation as the tool to evaluate relative performance of different heuristics under various shop floor conditions.

(5) In the previous studies, the most popular performance measure used was average flow time, followed by average percent tardy and average tardiness. No study considered economically based measures, although cost-minimization (or value-maximization) is as important as other performance measures such as flow time (measuring efficiency) and percent tardy (measuring effectiveness).

(6) In the previous studies, the most common assumptions made were sequence independent setups, only machine availability constrained (i.e., labor is not a constraint), no machine breakdown, transportation time (or cost) neglected, and no cycling allowed.

(7) Most studies assumed that jobs arrived according to a Poisson process and processing times were either normally distributed or uniformly distributed.

Research on Manufacturing Systems with Forbidden Early Shipment

This section reviews the major published research efforts concerned with manufacturing systems with Forbidden Early Shipment (FES). Table 2.2 shows the summary of published research on manufacturing systems with forbidden early shipment. This table is made by the author during this research effort. The purpose of this table is to give a comparison of the published research. The key items include the major assumptions, experimental factors, performance measures, shop model, scheduling rules tested, and study methodology. Each study included in the table is discussed in some detail below.

Kanet and Christy (1984) argued that the inclusion of the requirement that orders should not be shipped prior to their due dates was representative of many real systems. They illustrated that for such forbidden early shipment systems, shortest processing time (SPT) scheduling rule no longer minimized the average number of jobs in the system. They additionally argued that forbidden early shipments changed the way in which managers would run these systems. For the case where order allowances are held constant (i.e., system lead times were identical for each order), they showed that mean system inventory, mean order flow time, and mean order tardiness were directly correlated, i.e., optimizing one automatically optimized the other two.

TABLE 2.2

SUMMARY OF PUBLISHED RESEARCH ON MANUFACTURING SYSTEMS WITH FORBIDDEN EARLY SHIPMENT

Research	Major Assumptions	Experimental Factors (Levels)	Performance Measures	Shop Model	Scheduling Rules Tested
1. Kanet and Christy (1984) [FRAM]			Flow time Tardiness Mean inventory value (MIV)	Any Mfg systems with FES An example: M/M/1 queue	FCFS
2. Kanet and Christy (1988) [SIMU]	A1,B1 C1,M1 P1,S1 T1	Rescheduling policy (4) Allowance level (4) Updating interval (3) Demand skewness (3)	Tardiness Percent tardy MIV	A job shop with 8 machines	ODD SOPT
3. Morton, Lawrence, Rajagopolan, and Kekre (1988) [SIMU]	Ar,B1 C1,M1 Pr,S1 T1	Slack factor (2)	Net present value (NPV)	A single machine A job shop A flow shop A bottleneck job shop A proportional flow shop	SCHED-STAR (2 vers.) Early/tardy COV/AQT COV/QLR COV/IR CR/AQT CR/QLR CR/IR
4. Kanet and Christy (1989) [SIMU]	A1,B1 C1,M1 P1,S1 T1	Allowance method (2) Allowance level (5)	Flow time Tardiness Percent tardy Inventory MIV	A job shop with 8 machines	ODD
5. Scudder and Hoffmann (1989) [SIMU]	A1,B1 C1,M1 P3,S1 T1	Utilization (4) Delta (2)	Earliness Tardiness Inventory value	A job shop A flow shop each with 9 machines	CR OPCR PRF/OPT VLADRAT

TABLE 2.2 (Continued)

Research	Major Assumptions	Experimental Factors (Levels)	Performance Measures	Shop Model	Scheduling Rules Tested
6. Christy and Kanet (1990) [SIMU]	A1,B1 C1,M1 P1,S1 T1	Allowance method (2) Allowance level (2)	Tardiness Percent tardy Inventory MIV	A job shop with 8 machines	ODD MOD SOPT OPCR TSLK
7. Scudder, Smith-Daniels, and Rohleder (1990) [SIMU]	A1,B1 C1,M1 P3,S1 T1	Utilization (2) Order release (2) Raw material delivery (2) Due date multiplier (2) Tardiness penalty (2) Job length (3) # of operations per job (2) Interest rate (3)	NPV	A job shop with 9 machines	CR OPCR PRF/OPT VLADRAT
8. Lawrence (1991) [COMP]	A5,B1 C1,M1 P5,S1 T1	Machine load (11) Tardy/early cost ratio (6)	NPV, WIP Early/tardy costs Flow time Earliness Tardiness Flow time Makespan # tardy Tardiness Maximum tardiness	A single machine	MTP/RDE ET/ETR COV/AQT COV/QLR CR/AQT CR/QLR
9. Rohleder and Scudder (1992) [SIMU]	A1,B1 C1,M1 P3,S1 T1	Allowance level (4)	NPV, MIV Tardiness Percent tardy Inventory	A job shop with 9 machines	ODD,MOD SOPT,OPCR TSLK,OPSLK LWKR,CR EDD,MDD

TABLE 2.2 (Continued)

Notations for major assumptions:

A1 (A2,A3,A4,A5): Exponential (fifth order Erlang, normal, uniform, deterministic) interarrival times
B1 (B2): Without (with) bottlenecked machines
C1 (C2): Machine (machine and worker) constrained
M1 (M2): Without (with) machine breakdowns
P1 (P2,P3,P4,P5): Exponential (third order Erlang, normal, uniform, deterministic) processing times
S1 (S2): Sequence independent (dependent) setups
T1 (T2): Transportation times or costs neglected (considered)
Ar (Pr): Interarrival time (or processing time) which is a random variable and depends on other variables

Notations for research methodology:

[SIMU]: Simulation
[COMP]: Computation
[FRAM]: Framework

The precise analytical relationship between these measures was derived.

Christy and Kanet (1988) studied the performance of open order rescheduling policies in a job shop with forbidden early shipment. This paper was based on a Ph.D. dissertation done by Christy (1984) at the University of Georgia. Open order rescheduling is the act of changing the due date of a job (order) that has been released previously to the productive system. Four different order updating policies were examined in this study. For the EL (earlier or later) order updating policy, open order can be revised to either an earlier or a later date. For the EO (earlier only) policy, open orders can only be revised to an earlier date. For The OL (later only) policy, open orders can only be revised to an later date. For the OO (no earlier or later update) policy, no changes can be made to open order due date.

Christy and Kanet's study utilized computer simulation to model a job shop which consisted of eight machines. Two scheduling rules were used in this study, i.e., EODD (earliest operation due date) and SOPT (shortest operation processing time). The results showed that open order rescheduling was beneficial only when allowances were loosely set based on tardiness measure. The results also indicated that inventory performance was improved by order rescheduling, particularly in cases when due dates were revised to earlier times than originally forecast.

Morton et al. (1988) described another approach to shop scheduling. They developed a price-based shop scheduling module, entitled SCHED-STAR, that used cost-benefit analysis to make job release and priority decision. Iterative internal simulations were used to derive prices and lead times for cost-benefit calculations. They tested the module over several different shop configurations using the net present value criterion that includes explicit earliness and tardiness penalties under the assumption of forbidden early shipment. The shop configurations that were used included a single machine, flow shops, and job shops.

The priority rules tested in Morton et al.'s study included critical ratio (CR), a weighted version of COVERT (Vepsalainen and Morton, 1987), and an early/tardy heuristics. The release policies used included IR (immediate release), AQT (average queue time), and QLR (queue-length release). The SCHED-STAR module has been coded and tested with artificial data. The results showed that the SCHED-STAR heuristic dominated the other rules and release policies.

Kanet and Christy (1989) compared two well-known methods for setting order allowances in a job shop with forbidden early shipment. One method for setting a job's allowance is to make it proportional to the total processing time for the job (i.e., TWK). The other method is PPW that a job's allowance is obtained by adding to the total job processing time an allowance for waiting that is proportional to the number of operations that the job requires.

Kanet and Christy's study utilized computer simulation to model a job shop which consisted of eight machines. The only scheduling rule used in this study was ODD (earliest operation due date). The results of computer simulations over a wide range of average due date difficulty suggested that TWK (total work content) was the dominant procedure by virtue of providing both lower tardiness and lower inventory.

Scudder and Hoffmann (1989) examined the performance of four priority scheduling rules in a job shop and a flow shop both with forbidden early shipment. The four scheduling rules used were the same as a earlier study (Scudder and Hoffmann, 1987). These scheduling rules were CR (critical ratio), OPCR (operation critical ratio), PRF/OPT (which refers to the ratio of total profitability of a job to the setup and run time at the current workcenter), and VLADRAT (which refers to the ratio value added so far to a job to the total value it will have upon completion).

Scudder and Hoffmann's study utilized computer simulation to model a job shop and a flow shop which consisted of nine machines per shop. Little difference was found between the job shop and flow shop configurations for this environment. The major finding was that creating two-class queues (i.e., active and inactive queues) was very effective in reducing the amount of finished-goods inventory, while at the same time causing only a slight increase in work-in-process. In addition, CR (critical ratio) and OPCR (operation critical ratio) performed well on all measures. In earlier research (Scudder and Hoffmann, 1987), when early shipments were allowed, VLADRAT, which refers to the ratio value added so far to a job to the total value it will have upon completion, was an excellent alternative to CR (critical ratio) at most utilization levels.

Christy and Kanet (1990) examined the performance of five priority scheduling rules (all time-based rules) in a job shop with forbidden early shipment. The five scheduling rules used were ODD (earliest operation due date), MOD (modified operation due date), SOPT (shortest operation processing time), OPCR (operation critical ratio), and TSLK (based on a job's slack time per remaining operation). Two methods for setting order allowances, i.e., TWK (total work content) and PPW (processing plus waiting), were used in this study.

Christy and Kanet's study utilized computer simulation to model a job shop which consisted of eight machines. The results suggested that the MOD (modified operation due date) scheduling rule in conjunction with TWK (total work content) allowance setting would achieve the concurrent objectives of timely order completion with controlled inventory investment. The SOPT (shortest operation processing time) rule, which has been found best for minimizing inventory when early shipments were allowed, was not only a poor rule but was clearly the worst rule among the rules examined.

Scudder et al. (1990) examined the performance of four priority scheduling rules in a job shop with forbidden early shipment. The four scheduling rules used were the same as earlier studies (Scudder and Hoffmann, 1987 and 1989, Scudder and Smith-Daniels, 1989). The only performance measure used was the net present value. The net present value measure provides a means of balancing a variety of performance criteria that have been treated as separate objectives previously, including work-in-process inventory, finished goods inventory, mean flow time, and mean tardiness, while also providing a means of measuring monetarily the value of various shop scheduling approaches.

Scudder et al.'s study utilized computer simulation to model a job shop which consisted of nine machines. The results showed that the critical ratio (CR) rule provided higher average net present value than the three other rules in the study. However, in some situations that were consistent with just-in-time practice, value-based rules also performed well. In earlier research (Scudder and Smith-Daniels, 1989), where early shipments were allowed, VLADRAT (which refers to the ratio value added so far to a job to the total value it will have upon completion) was superior (i.e., higher net present value) to the other rules examined.

Lawrence (1991) investigated a static single-machine scheduling problem in forbidden early shipment environments using the net present value objective. A job dispatching rule MTP (marginal tardiness penalty) and a job release rule RDE (release at delay equilibrium) were developed through a marginal cost analysis of the net present value objective, and the composite MTP/RDE scheduling policy (i.e., MTP job dispatching rule and RDE job release rule) was extensively tested against several other benchmark heuristics obtained from the literature. The RDE (release at delay equilibrium) release policy launches jobs into the shop when the estimated marginal costs of further delay equal benefits. Once released, the MTP (marginal tardiness penalty) dispatching rule prioritizes jobs by calculating a dynamic apparent marginal tardiness penalty for each job, and dispatches that job with the highest apparent priority for processing. The results showed that the MTP/RDE (MTP job dispatching rule and RDE job release rule) policy provided the best average performance for each of the cost-based criteria (e.g., net present value and early/tardy costs), but did not provided superior performance for non-cost criteria (e.g., flow time and tardiness).

Rohleder and Scudder (1992) re-examined the results of a study done by Christy and Kanet (1990). In Christy and Kanet's study, they examined the performance of five priority scheduling rules in a job shop with forbidden early shipment and their primary performance measure was time-weighted inventory value. In this study (Rohleder and Scudder, 1992), the authors examined the performance of ten scheduling rules using the net present value measure and other measures used in Christy and Kanet's study (1990).

The ten scheduling rules used in Rohleder and Scudder's study included five scheduling rules used in Christy and Kanet's study (1990) and other five scheduling rules, i.e., OPSLK (operation slack per remaining operation), LWKR (least work remaining), CR (critical ratio), EDD (earliest due date), and MDD (modified due date). Computer simulation was utilized to model a job shop which consisted of nine machines. The results showed that using net present value and inventory objectives led to different scheduling decisions and the job-based rules outperformed the operation-based rules. Another interesting result was the overall poor economic performance of MOD (modified operation due date) across all analysis.

Summary

From the published research on manufacturing systems with forbidden early shipment as reviewed above, some important conclusions can be drawn:

(1) Most researchers have proposed heuristics as a solution to solve scheduling problems in forbidden early shipment environments. No heuristic performed better than others in all cases. The choice of heuristics depended on the systems definition (including shop model, job characteristics, and assumptions made), shop floor conditions, and performance measures used.

(2) Most researchers utilized computer simulation as the tool to evaluate relative performance of different heuristics under various shop floor conditions in forbidden early shipment environments.

(3) Some studies (e.g., Scudder and Hoffmann, 1989, Scudder et al., 1990) considered both time-based scheduling rules and value-based rules scheduling rules. The results showed that, in general, time-based rules (e.g., critical ratio) outperformed value-based rules on most measures. But, in earlier research (e.g., Scudder and Hoffmann, 1987, Scudder and Smith-Daniels, 1989), when early shipments were allowed, some value-based rules (e.g., VLADRAT) performed very well when compared with other rules.

(4) In the previous studies, the performance measures used frequently were average flow time, average tardiness, average percent tardy, average number in inventory, average inventory value, and average net present value.

(5) In the previous studies, the most common assumptions made were sequence independent setups, no key or shared

machines, only machine availability constrained (i.e., labor is not a constraint), no machine breakdown, transportation time (or cost) neglected, and no cycling allowed.

(6) Most studies assumed that jobs arrived according to a Poisson process and processing times were either normally distributed or exponentially distributed.

(7) Most studies only considered a job shop containing eight or nine machines with forbidden early shipment. Only two studies (Morton et al., 1988, Scudder and Hoffmann, 1989) included flow shops with forbidden early shipment. No study was investigated in cellular manufacturing with forbidden early shipment.

CHAPTER III

RESEARCH PLAN

Introduction

In this chapter, research areas are identified from the previous research on group scheduling in cellular manufacturing and on manufacturing systems with forbidden early shipment. Then, based on these identified areas, the goal and objectives of the research are developed and the scope of the research is discussed.

Research Areas Identified from the Literature Review

In Table 3.1, the classifications of the major published research on group scheduling in cellular manufacturing are presented. The classifications are based on the shop models and the types of shipments. The shop models include job shop cells and flow shop cells (both with and without workcenter sharing). Two types of shipments are used in the table, i.e., early shipments are allowed and early shipments are forbidden.

TABLE 3.1

CLASSIFICATIONS OF PUBLISHED RESEARCH ON GROUP SCHEDULING IN CELLULAR MANUFACTURING

Shop Model	Early Shipments are Allowed	Early Shipments are Forbidden
Job Shop Cells without Workcenter Sharing	Vaithianathan and McRoberts (1982) Mosier et al. (1984) Kelly et al. (1986) Flynn (1987) Sassani (1990) Mahmoodi et al. (1990a,b) Mahmoodi and Dooley (1991) Ruben et al. (1993)	
Flow Shop Cells without Workcenter Sharing	Vakharia and Chang (1990) Wemmerlov and Vakharia (1991) Russell and Philipoom (1991) Mahmoodi et al. (1992)	
Job Shop Cells with Workcenter Sharing	Ang and Willy (1984)	
Flow Shop Cells with Workcenter Sharing		

In Table 3.1, several studies related to group scheduling in cellular manufacturing where early shipments were allowed have been done. No effort was directed to investigate the performance of group scheduling heuristics in cellular manufacturing with forbidden early shipment.

The results of a survey (Wemmerlov and Hyer 1989) showed that most cells implemented in industry were flow shop cells (or close to flow shop cells). This survey also showed that 20% of the companies with manned cells and 14% of those with unmanned cells reported that machines were shared between cells. In Table 3.1, it can be found that no work has been done to study the flow shop cells with workcenter sharing. Also, no study related to flow shop cells considered economically (or monetary) based measures, and different routings and order sizes for different part types.

Based on the research areas identified above, this research effort investigated the performance of group scheduling heuristics in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment. Five group scheduling heuristics have been developed and then evaluated under various shop conditions. Ten performance measures, which include an economically based measure and other time-based or inventory-based measures, have been selected to collect statistics. Order characteristics will include different routings and order sizes for different part types.

Research Goal

The goal of this research was to identify the important factors impacting system performance and to evaluate the performance of group scheduling heuristics under various shop conditions in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment. To assure applicability of the results to the industry, this research included different routings and order sizes for different part types. Ten performance measures, which are typically used in industry and previous studies, have been chosen in this research.

Results of the evaluation have been used to rank the performance of group scheduling heuristics under various shop conditions with respect to the performance measures chosen. The best performing heuristics can provide guidance for schedulers in the selection of heuristics based on the shop conditions and the performance measures that are most important in their industry.

Research Objectives

In order to achieve the research goal, the following research objectives have been identified.

Objective 1. The first objective of this study was to propose a group scheduling process in cellular manufacturing. The purpose of the group scheduling process is to define how and when orders enter the system, are processed, and leave the system. This process includes the following stages: order arrival (or order entry), order release, queue selection, job dispatching, order storage (if completed before due date) and order shipment.

Objective 2. The second objective of this study was to define the system, to identify experimental factors, and to choose performance measures for this research. The system definition includes the shop model, order characteristics, and all assumptions made. The issue of experimental (or shop) factors is to identify the critical factors impacting the system performance and to define their critical levels. Five group scheduling heuristics (group scheduling heuristic is one of the experimental factors included in this research) have been developed in this study. Ten performance measures, which include an economically based measure and other time-based or inventory-based measures, have been selected to collect the statistics.

Objective 3. The third objective of this study was to develop an experimental design and to decide the data generation procedures. The issues related to the experimental design include the type of design, research vehicle, number of replications for each experiment, and statistical analysis procedures. The data generation procedures were used to generate the input data for this research.

Objective 4. The fourth objective of this study was to conduct all experiments and then to analyze the experimental results by statistical analysis procedures. The results of the statistical analysis have been used to rank the performance of heuristics under various shop conditions with respect to the performance measures chosen. The best performing heuristics can provide guidance for schedulers in the selection of heuristics based on the shop conditions and the performance measures that are most important in their industry.

Research Scope

The scope of the research effort will be limited to a small production system (i.e., five workcenters in each cell and two cells in the cellular system) due to economic and time constraints. Large systems are not directly investigated. We can treat the system used in this study as a sub-system of a larger production system. The basic assumption guiding the investigation is that the findings would be generally transferable to larger systems operating under the same conditions.

This research presumes that the part family/machine group formation (i.e., cell design) is not a research question. It is assumed that part subfamilies and machine group formation procedures, already developed, are suitable to identify part families and their corresponding production cells (Farrington, 1991, Mahmoodi et al., 1992, Ruben et al., 1993).

CHAPTER IV

RESEARCH METHODOLOGY

Introduction

A schematic diagram for the basic elements of a scheduling study using computer simulation is shown in Figure 4.1. In this diagram, the inputs include the system definition, performance measures, experimental factors, experimental design, and data gathering/generation. The experimental factors include controllable factors (e.g., scheduling rules) and environmental factors (e.g., demand pattern). The outputs are preferable scheduling rules based on simulation outcome.

This chapter discusses the research methodology employed in conducting this study. First of all, a group scheduling process in cellular manufacturing with forbidden early shipment is proposed. The proposed process serves as the procedures to schedule jobs in the system. Next, the basic elements required for this study are defined based on Figure 4.1. These elements include the system definition (which consists of the shop model, order/job characteristics, and assumptions made), performance

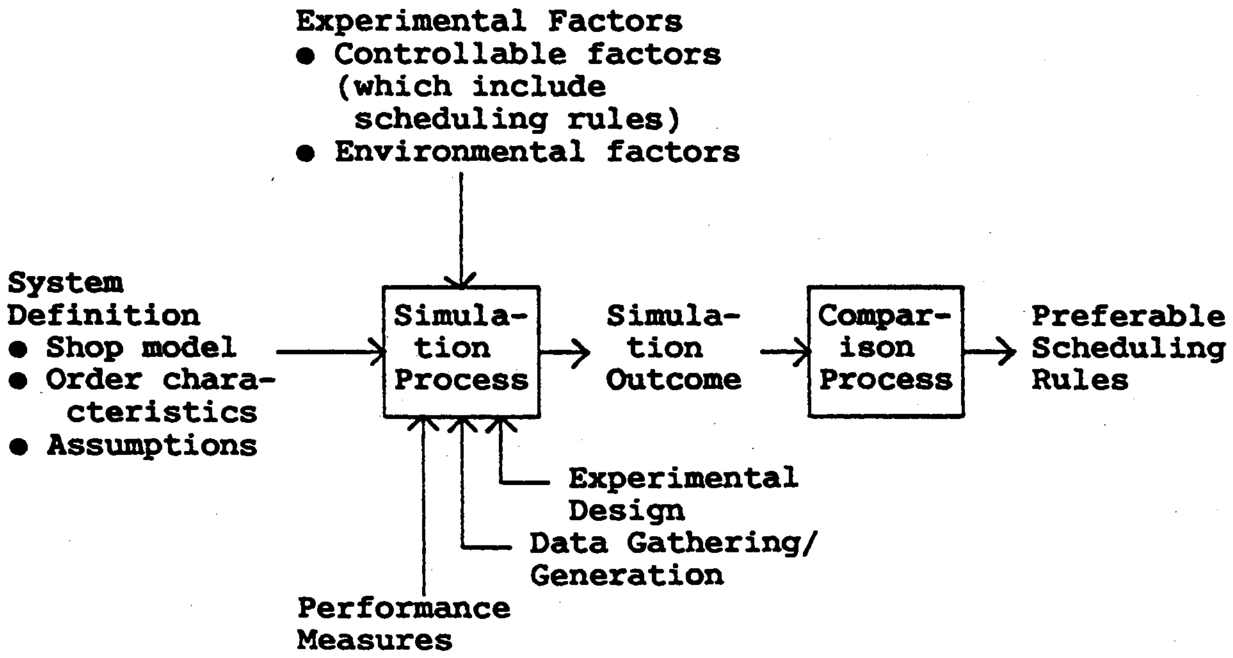


Figure 4.1 **Schematic Diagram for the Basic Elements of a Scheduling Study Using Computer Simulation**

measures, and experimental factors selected. Then, the data generation procedures and the implementation issues such as the experimental design considerations, generation of simulation model, and model verification and validation are discussed.

Group Scheduling Process in Cellular Manufacturing with Forbidden Early Shipment

The proposed group scheduling process in cellular manufacturing with forbidden early shipment is a composite and modified version of Mahmoodi and Dooley's (1992) model of group scheduling systems and Kanet and Christy's (1984) model of manufacturing systems with forbidden early shipment. This proposed process consists of five stages, as shown in Figure 4.2. In the first stage, order entry, the due date of an arriving order is externally assigned by the customer.

In the second stage, order release, the order release policy determines the release time of an order by subtracting its allowance, which can be defined as the amount of time that is budgeted to complete an order, from its externally assigned due date. There are several reasons for not releasing orders as they are received. The major reason is that orders released to the shop long before they are needed will compete with more urgent orders for resources (e.g., workcenters) and may interfere with

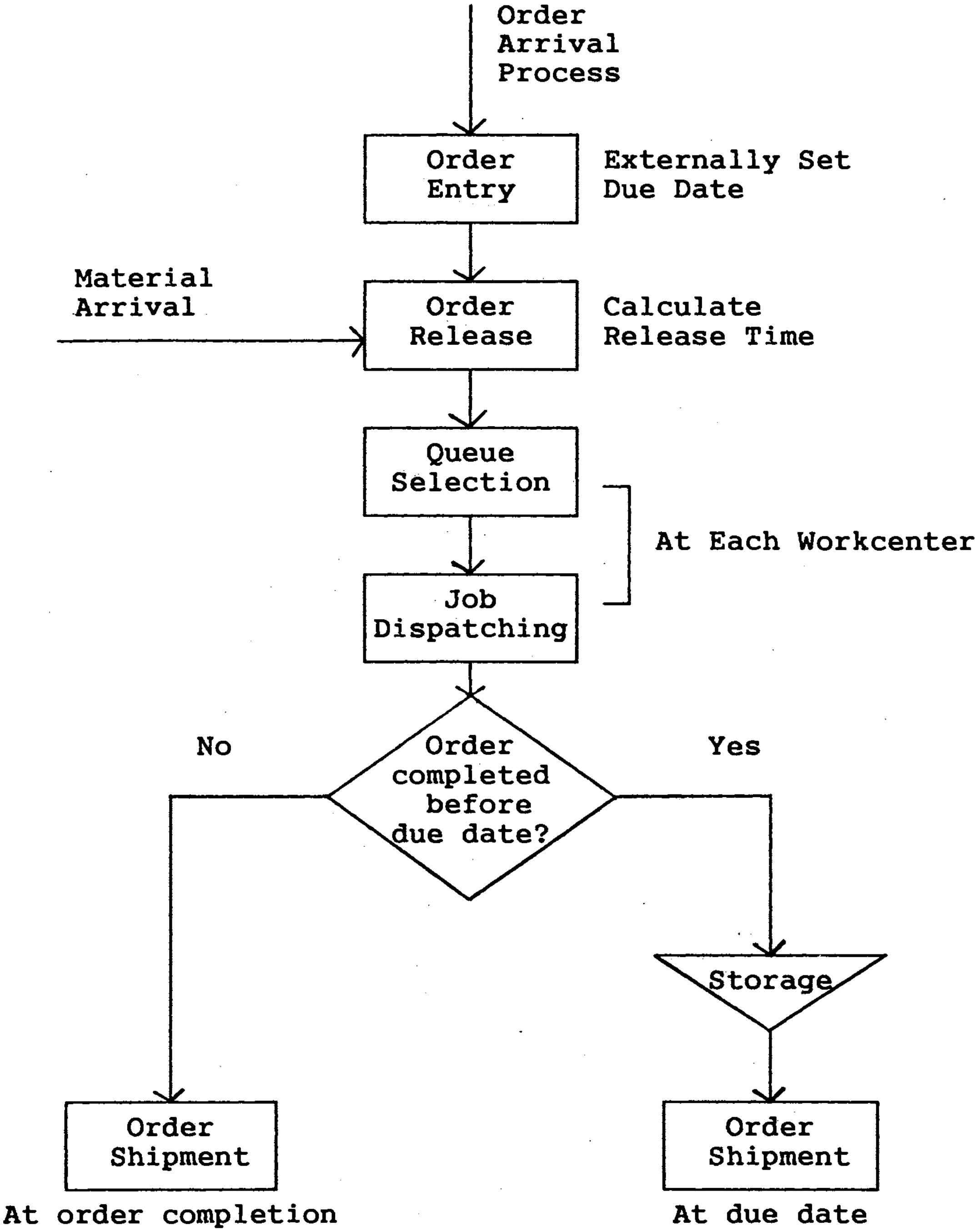


Figure 4.2 **Group Scheduling Process in Cellular Manufacturing with Forbidden Early Shipment**

the progress of those orders (Ragatz and Mabert, 1988). Determining an order's release time is a very important issue for shops in make-to-order and forbidden early shipment environments. Also, materials (raw materials or components) should arrive at the shop in this stage.

In the third stage, which is queue selection, the subfamilies are sequenced at each workcenter, and the fourth stage, which is job dispatching, involves sequencing jobs (i.e., transfer batches) within the subfamilies at each workcenter. In the last stage, an order's completion date is checked with its due date. If the order is completed before its due date, it will be held in storage until its due date. If the order is completed at or after its due date, it will be shipped immediately.

The System

Shop Model

This research utilized a flow shop cellular system which consisted of two flow shop cells as shown in Figure 4.3. Each of the flow shop cells had five workcenters, with the last workcenter shared between the cells; each workcenter contained a single machine. Orders could enter the system from workcenter 1, 2, or 3 and exit from workcenter 3, 4, or 5 (in cell A or B). The routings of orders could be different and depended on their part types.

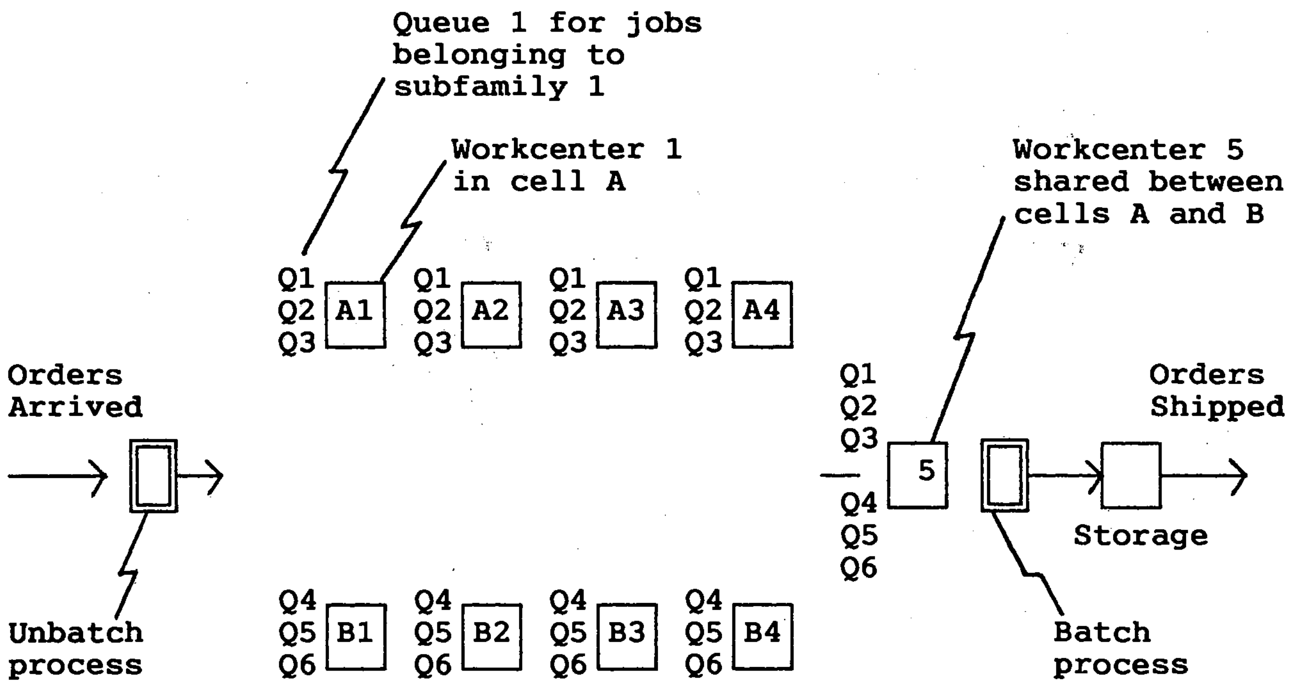


Figure 4.3 Schematic Diagram for the Shop Model

The shared workcenter can be a coordinate measuring machine (CMM), a paint booth, a degreaser, or a heat treating facility, etc. This arrangement may be required because of its cost, toxicity, requirement for energy, or other dominating criterion. Workcenter sharing can decrease the total number of workcenters (or machines) necessary and increase machine utilization. On the other hand, workcenter sharing can cause control problems when machine availability conflicts occur (Greene and Sadowski, 1983 and 1984). To cope with these problems, more complex scheduling heuristics may be required.

The shop model was intended to generally represent a real flow shop cellular system (Wemmerlov and Hyer, 1989), and the use of a hypothetical shop might lead to more generalizable results than the use of any particular real model (Ragatz, 1985). Also, the use of a hypothetical model could be linked to previous research more readily since much of the previous research has used hypothetical models (Mahmoodi, 1989).

The reason to select the flow shop cellular system in this research was because most cells implemented in industry were flow shop cells (or close to flow shop cells) based on a survey (Wemmerlov and Hyer, 1989). Another reason was that one of the characteristics that the cellular layout outperforms the process (or functional) layout is "unidirectional flow of work within a cell (i.e., flow shop cell)" (Morris and Tersine, 1990). The reason to select the cellular system with workcenter sharing was because 20% of the companies with manned cells and 14% of those with unmanned cells reported that machines were shared between cells (Wemmerlov and Hyer, 1989), but no study has been done in this area.

The purpose of selecting the specified shop size (i.e., two cells) was to investigate the effects of workcenter sharing with the least number of cells (due to economic and computational constraints). The cell size (i.e., five workcenters per cell) considered was well within the norm since the survey (Wemmerlov and Hyer, 1989) reported that the average cell size was 6.2 machines for the manned cells and 4.7 machines for the unmanned cells.

Order/Job Characteristics

Thirty part types were generated in this study; five part types belonged to each of six part subfamilies. The first cell produced the first three subfamilies and the second cell produced the remaining subfamilies. A subfamily is a grouping of part types with similar setups (i.e., no major or subfamily setups required between part types within a subfamily). In most cases, a cell is designed to produce a part family (Burbidge, 1971) which may contain several subfamilies. In this study, each order only contained one part type. For the purpose of comparison, the number of subfamilies per cell and the number of part types per subfamily were adopted from a recent paper (Mahmoodi et al., 1992) which examined group scheduling heuristics in a 5-workcenter flow shop cell.

An order size depended on its part type and was normally distributed with a mean shown in Appendix A and a coefficient of variation (CV) of 0.25. The distribution was truncated on the left so that negative order sizes were prohibited. An order was divided into transfer batches (called jobs in this study) before entering the shop. The size of a transfer batch was equal to the arriving order size (at the high level of cell transfer batch) or the standard container size (10 parts per container) (at the low level of cell transfer batch). A job's routing depended on its part type and the generation procedures of routings are explained in the section titled "Data Generation Procedures" in this chapter. Jobs were processed by between three and five operations with one operation on one workcenter. The routing table for all part types is shown in Appendix A.

Orders arrived according to a Poisson process. The interarrival time of a part type was exponentially distributed with a mean calculated by the following equation.

Mean interarrival time for part type i =
(120,000 minutes per year) /
(Annual orders for part type i)

The exponential distribution for the interarrival time was selected because most research on group scheduling in flow shop cellular systems and on manufacturing systems with forbidden early shipment used this distribution (see summaries of literature review in chapter II). This distribution was selected to represent the worst case from a broad range of interarrival time variability.

The processing times of a job depended on its part type and the workcenters on which it was processed. They were initially generated from a normal distribution which was truncated on the left so that negative processing times were prohibited. Then, a normalization process was applied to guarantee that the annual processing workload across all workcenters in the system was, on average, 90% of the average annual workload (AAW). The truncated normal distribution was selected since it was commonly used in previous research on group scheduling in flow shop cellular systems and on manufacturing systems with forbidden early shipment. The generation procedures of the processing times are explained in the section titled "Data Generation Procedures" in this chapter.

The minor setup times between any two part types in the same subfamily were assumed to be included in the processing times. A major (or subfamily) setup time was added to the first job when a new subfamily was selected. Again, a normalization process was applied to make sure that the major setup workload across all workcenters in the system was, on average, 10% of the average annual workload (AAW). The generation procedures of the major setup times are explained in the section titled "Data Generation Procedures" in this chapter.

An order's due date was externally assigned by the customer and was equal to the order's arriving time plus the duration randomly generated from a Uniform (11 days,20 days) distribution. The criterion to choose the interval was to result in the same degrees of the average percent tardy and average percent early. Choosing this interval (i.e., 11 to 20 days) resulted in, approximately, the same degrees of the average percent tardy and average percent early (around 47%) for ADD/EDD, ACR/CR, and ASLK/SLK heuristics when the average annual workload and demand pattern variability were both set to high level.

Assumptions in the System

The following is a summary of the basic assumptions in the flow shop cellular system:

(1) No backtracking was allowed, e.g., a job could not move from workcenter 1 to workcenter 3 and then back to workcenter 2.
(2) No cycling was allowed, i.e., a workcenter was visited by a job a maximum of one time.
(3) There were no scrapped or reworked parts.

(4) Each machine could handle, at most, one operation at a time.

(5) Each job involved a strict sequence of operations without assembly or partition.

(6) The system was constrained by machine availability only (i.e., labor was not a constraint).

(7) Operations of jobs could not be interrupted once the operations have been started (no preemption).

(8) Machine breakdowns were not considered, i.e., machines were maintained in such a good condition that the frequency of machine breakdowns was very low.

(9) The part subfamilies and cells have already been designed using one of the many part/machine group formation techniques available. That is, cell formation problems would not be considered in this research.

(10) Average annual workload was set to 90% (high level) or 80% (low level) of total capacity of the system, where total capacity is equal to 2,000 hours per year (i.e., system operates 50 weeks per year, 40 hours per week).

(11) Materials (raw materials or components) in the required amounts arrived in order release stage.

(12) There were no limits on the queue sizes and, therefore, no blocking occurred.

(13) Transportation times (or costs) were neglected since workcenters arranged in a cellular layout were likely to be in close proximity to one another.

Identification of Factors Impacting System Performance

Based on the literature review in Chapter II and the characteristics of the system defined in this study, there are several factors which may have an impact on system performance. These factors are listed as follows: (1) group scheduling heuristic, (2) demand pattern, (3) processing time, (4) setup time, (5) processing time and setup time at the shared workcenter, (6) order release policy, (7) order allowance method, (8) average annual workload, (9) machine utilization, (10) shop size and cell size, (11) part mix, (12) interest rate and tardy/early cost ratio, and (13) cell transfer batch.

This section presents a brief discussion of each of these factors and their potential impact on system performance. The reasons for inclusion or exclusion of each factor in the study are also discussed. Based on resource constraints (i.e., cost of computer usage and time available to complete the study, etc.) and the evaluation of the importance of the individual factors, some factors have to be eliminated from consideration (i.e., held constant) during experimentation.

Group Scheduling Heuristic

Group scheduling studies (see literature review in Chapter II) have received considerable attention recently since group scheduling heuristics can maximize the advantages of cellular manufacturing by further reducing the overall machine setup time. Another reason is that group scheduling heuristics can reduce the disadvantage of cellular manufacturing (i.e., inflexibility of shop) by employing a diverse range of part subfamilies to increase the shop flexibility (Mahmoodi et al., 1990b, Lee, 1985).

Since the primary goal of this research effort is to investigate the group scheduling heuristics in a flow shop cellular system, group scheduling heuristics should be naturally included in this study. Due to the complexity of the group scheduling heuristics used in this study, they are discussed in the section titled "Group Scheduling Heuristics".

Demand Pattern

In a Ph.D. dissertation done at the Oklahoma State University, Farrington (1991) developed a methodology for selecting the appropriate system design based on the prevailing characteristics of the production environment. Based on the author's evaluation of factor importance, it appeared that the demand pattern variability was one of the factors which had the greatest impact on system performance (Farrington, 1991).

One of systems tested in Farrington's study was a single flow shop cell with 2 to 7 machines. A limitation in his study was that only the first-come-first-served (FCFS) dispatching rule was used. It is unclear from the literature what is the effect of various levels of demand pattern variability on the performance of group scheduling heuristics in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment. Since customer demand is what drives a manufacturing operation, it was felt that gaining an understanding of the impact of demand pattern variability on performance of heuristics was extremely important and should be investigated in this study.

Two levels of the demand pattern variability were investigated. The objective in choosing these two levels was to pick values that were far enough apart that a discernible difference, if any, in performance could be observed. The two levels of demand pattern variability investigated were: high demand pattern variability - a demand pattern with long interarrival times and large order sizes (i.e., on average 21 orders per year and 34 units per order) and low demand pattern variability - a demand pattern with short interarrival times and small order sizes (i.e., on average 40 orders per year and 17 units per order).

Processing Time

Some previous research on group scheduling in cellular manufacturing and on manufacturing systems with forbidden early shipment used processing time (either average processing times or processing time variability) as an experimental factor in their studies. In general, low average processing times tend to result in good performance measures such as flow time and work-in-process (when other factors hold constant), and vice versa. Also, it was felt that identifying the effect of the key machine(s) was more important than considering the processing time variability in a flow shop cellular system. Therefore, it was decided that this factor should not be investigated in this study.

Setup Time

Some previous research on group scheduling in cellular manufacturing and on manufacturing systems with forbidden early shipment used setup time (e.g., setup time to run time ratio) as an experimental factor in their studies. In general, as the major (or subfamily) setup time decreases, the average flow time and average work-in-process will decrease when other factors hold constant, and vice versa. Since it was already known that a system always performs better with setup time reduction, it was decided that this factor should not be investigated in this study.

Processing Time and Setup Time at the Shared Workcenter

Workcenter sharing was one of the features included in this study. Since the last workcenter was shared between two cells in the flow shop cellular system, the processing time and setup time at the last workcenter alone might have an impact on performance of heuristics. Because a normalization process was applied to generate processing times and subfamily setup times at all workcenters in this study, the last workcenter was not necessarily the bottleneck. It was felt that identifying the effect of the key machine(s) was more important than considering the processing time or setup time at the shared workcenter in a flow shop cellular system. Therefore, it was decided that this factor should not be investigated in this study.

Order Release Policy

Orders released to the shop long before they are needed will compete with more urgent orders for resources and may interfere with the progress of those orders (Ragatz and Mabert, 1988). Order release is accomplished by periodically examining all unreleased orders and deciding which, if any, to release. The order release policies which were frequently considered in previous research were immediate release and delayed (i.e., controlled) release (e.g., Scudder et al., 1990, Mahmoodi et al., 1990a). Immediate release policy releases orders to the shop as they arrive. Delayed release policy releases orders by estimating orders' allowances in the shop in order to meet their due dates.

This factor was originally included in the simulation pilot runs and two policies (i.e., immediate release and delayed release) were investigated. But, the results of the analysis of variance (see Appendix F) showed that this factor was not a major factor compared with the other three factors (i.e., group scheduling heuristic, average annual workload, and demand pattern variability). Therefore, it was decided that this factor should not be included for further investigation and delayed release would be used as the order release policy in this study. An order's release time can be determined by subtracting its allowance from its due date. If an order's release time is less than its arrival time, the order will be released as it arrives.

Order Allowance Method

Order allowance (or system lead time) refers to the amount of time that is budgeted to complete an order. Estimating orders' allowances is a very important issue for shops which use delayed release as their order release policy. Some previous research on manufacturing systems with forbidden early shipment considered different order allowance methods in their studies. Two order allowance methods which were frequently considered were TWK (total work content) and PPW (processing plus waiting). The results of previous studies (e.g., Kanet and Christy, 1989, Christy and Kanet, 1990) indicated that TWK was the dominant order allowance method with respect to most of the measures used (e.g., average tardiness and percent tardy). Therefore, it was decided that this factor should not be included in this study and TWK would be used as the order allowance method.

The TWK order allowance method assigns an order's allowance in proportion to the total mean processing time of the order. Mathematically, an order's allowance calculated by TWK can be expressed as:

$$A_i = K \sum_{j=1}^{M_i} P_{ij}$$

Where:

K = Order allowance level
A_i = Allowance of order i
M_i = Number of operations of order i
P_{ij} = Processing time of operation j of order i

It is generally accepted that an order allowance level (K) of 10 is often seen in industrial settings (Christy and Kanet, 1990). Therefore, an order allowance level (K) of 10 was used in this research.

Average Annual Workload

A fair comparison among several system configurations requires that these system configurations are similar and have identical machine workloads. Many previous research on group scheduling in cellular manufacturing and on manufacturing systems with forbidden early shipment (e.g., Mahmoodi et al., 1992, Lawrence, 1991) have shown that the machine workload had a major impact on performance of heuristics. It was felt that gaining an understanding of the impact of the machine workload on performance of heuristics was extremely important and should be investigated in this study.

One way to specify the machine workloads for different system configurations is to use the average annual workload (AAW). For example, an average annual workload of 80% (across all workcenters in the system) would have an average workload of 1600 hours/year if it is assumed that there are 2000 hours/year for a one shift operation. The average annual workload across all workcenters is, in fact, made up of two components: average annual processing workload and average annual setup workload.

In simulation pilot runs, several levels of the average annual workload were tested (e.g., 60%, 70%, 80%, and 90%). The results showed that the performance of the five heuristics was not significantly different on most of measures at a significance level of 0.10 when the average annual workload was equal to or less than 70% (see Appendix F). Based on the simulation pilot runs, two levels of the average annual workload were investigated in final simulation runs: high and low. The high level of the average annual workload had an average annual workload of 90% (i.e., 1800 hours/year). The high average annual workload consisted of the average annual processing workload of 1620 hours/year (90% of AAW) and setup workload of 180 hours/year (10% of AAW). The low level of the average annual workload had an average annual workload of 80% (i.e., 1600 hours/year). The low average annual workload consisted of the average annual processing workload of 1440 hours/year (90% of AAW) and setup workload of 160 hours/year (10% of AAW).

Machine Utilization

Based on the previous research, it was concluded that the machine utilization might have an impact on performance of heuristics. Because of the setup avoidance by using group scheduling heuristics, the average machine utilizations may not be identical (usually an interval, e.g., 75% -79%) among different heuristics for a given experimental condition. Moreover, by specifying the average annual workloads for all workcenters in the system, the average machine utilizations and the key (or bottlenecked) machine(s) would be consequently decided.

Since it was possible to specify the fixed levels of the average annual workload and the average annual workload has been chosen as an experimental factor, it was decided that the machine utilization would not be chosen in this study.

Shop Size and Cell Size

Due to economic (i.e., computer usage costs) and time constraints (see Appendix F), this factor would not be included in this study and a small production system (i.e., five workcenters in each cell and two cells in the cellular system) would be used. The basic assumption was that the findings would be generally transferable to larger systems operating under the same conditions. The shop size of two (i.e., two cells) with a workcenter shared between two cells was selected in this study. The cell size of five (i.e., five workcenters per cell) which was well within the norm (Wemmerlov and Hyer, 1989) was selected in this study. Future research should be undertaken to gain an understanding of the impact of shop size and cell size on performance of heuristics.

Part Mix

A cell is initially designed for a family of parts with a fixed part mix. In addition, a cell is designed with the maximum amount of flexibility possible to handle the maximum number of different part types. But to realistically obtain the cellular manufacturing benefits, there is a limited amount of flexibility. The problem arises when the part mix changes over a period of time that can cause an imbalance in cell loading (Greene and Sadowski, 1984).

Since the above situation was expected to be monitored in real-world cellular manufacturing systems, it was assumed that part mix was stable and, therefore, part mix would not be included as a factor in this study. This assumption is reasonable since only mature products can be produced in cellular manufacturing systems. Future research should be undertaken to explicitly consider changing part mix. The purpose of the future research will be to determine when a cell reorganization is required due to the changing part mix. Cell reorganization which includes modifying cell layout and grouping of part families is often costly.

Interest Rate and Tardy/Early Cost Ratio

Some previous research on manufacturing systems with forbidden early shipment used interest rate or tardy/early cost ratio as an experimental factor (e.g., Scudder et al., 1990, Lawrence, 1991). If the group scheduling heuristics used do not included any value or cost information, this factor can only affect the economically based measures (e.g., net present value) and cannot affect time-based (e.g., flow time) or inventory-based measures (e.g., percent tardy). In addition, it may be known that a system performs better or worse with changing levels of this factor. For example, increasing tardy/early cost ratio decreases the measure of net present value. Several simulation pilot runs were executed with the tardy/early cost ratios ranging from 1 to 10. The results of pilot runs (see Appendix F) showed that the ranking orders of the five heuristics were identical when changing the values of the tardy/early cost ratio although the magnitude of net present values were different. Therefore, it was decided that this factor should not be investigated and a tardy/early cost ratio of 5 was chosen in this study.

Cell Transfer Batch

A survey of cellular manufacturing systems in the U.S. industry (Wemmerlov and Hyer, 1986) reported that, in most cases, the cell transfer batch (i.e., batch size moved between workcenters) was reduced and/or determined by standard container size. It is unclear from this survey and other literature what is the effect of various levels of the cell transfer batch on performance of the cellular system defined in this study. It was felt that gaining an understanding of the impact of the cell transfer batch on performance of heuristics was extremely important and should be investigated in this study.

Two levels of the cell transfer batch were investigated. Again, the objective in choosing these two levels was to pick values that were far enough apart that a discernible difference, if any, in performance could be observed. The two levels of the cell transfer batch investigated were: high cell transfer batch with the size equalling the arriving order size and low cell transfer batch with the size equalling the standard container size (10 parts per container). When choosing the low cell transfer batch, several sizes (e.g., 1, 5, 10, 15, and 20 parts/container) were tested in the pilot runs. The results of pilot runs (see Appendix F) indicated that small cell transfer batch, say 1, would require extremely long computer run time because of numerous transfer batches (i.e., jobs) in the system. Due to economic and computational constraints, a size of 10 parts per container was selected for the low level of the cell transfer batch.

Group Scheduling Heuristics

Group scheduling heuristics are two-stage heuristic procedures used to sequence jobs in cellular manufacturing systems. In the first stage, a subfamily queue is selected based on a chosen queue selection rule. In the second stage, jobs are ordered within the subfamily queue based on a chosen job dispatching rule (Ruben et al., 1993).

Many combinations of queue selection rules and job dispatching rules were tested through a series of simulation pilot runs. The queue selection rules tested included FCFS (first-come-first-served), SPT (shortest processing time), APT (average processing time), EDD (earliest due date), ADD (average due date), CR (critical ratio), ACR (average critical ratio), SLK (slack), and ASLK (average slack). The job dispatching rules tested included FCFS (first-come-first-served), SPT (shortest processing time), SI* (a two-class truncated SPT rule), EDD (earliest due date), CR (critical ratio), and SLK (slack). Based on the results of pilot runs (see Appendix F) and the consideration of the features (e.g., forbidden early shipment) included in this research, three heuristics (i.e., ADD/EDD, ACR/CR, and ASLK/SLK) were selected for further investigation.

The three heuristics selected (i.e., ADD/EDD, ACR/CR, and ASLK/SLK) contain job information (e.g., jobs' due dates and/or total remaining processing times) in both their queue selection and job dispatching rules, but do not consider the workcenter status in the system. It was felt that gaining an understanding of the performance of different types of heuristics was very important. To consider the workcenter status and/or job information, several heuristics which included NJQA/CR, NJQA/SLK, NJQA/EQ, NEQA/EQ, NJQB/CR, NJQB/SLK, NLQB/CR, and NLQB/SLK were developed and initially tested. NJQA selects the subfamily queue with the smallest total number of jobs in the same type of queue ahead. NEQA selects the subfamily queue with the largest number of empty queues for the same type of queues ahead. NJQB selects the subfamily queue with the largest total number of jobs in the same type of queue behind. NLQB selects the subfamily queue with the largest number of lengthy queues (defined as a queue which has 5 jobs or more than 5 jobs in it) for the same type of queues behind. EQ sequences jobs within a subfamily queue by selecting the jobs which will go to the empty queues ahead. Based on the results of initial tests (see Appendix F) and the consideration to include both looking ahead and looking behind heuristics in this study, NEQA/EQ and NLQB/CR were selected for further investigation. These two heuristics contain both workcenter status and job information (e.g., jobs' routings or due dates).

To summarize, five group scheduling heuristics were selected for further investigation in this study: (1) ADD/EDD, (2) ACR/CR, (3) ASLK/SLK, (4) NEQA/EQ, and (5) NLQB/CR. These five heuristics are defined and discussed below:

ADD/EDD Heuristic

ADD/EDD selects the subfamily queue with the smallest average due date (ADD) and then utilizes the earliest due date (EDD) rule to sequence jobs within this subfamily queue. ADD (average due date) sums N_{qd} (a maximum value of 5 jobs) earliest due dates in a subfamily, then this total is divided by N_{qd}. Mathematically, ADD (average due date) determines the queue priority (at a workcenter) as:

$$\text{Minimum} \left(\sum_{i=1}^{N_{qd}} D_i \right) / N_{qd} \rightarrow \text{Priority queue}$$

Where:

D_i = Due date of job i
N_{qd} = Number of jobs in queue q used to calculate ADD (a maximum value of 5)

The prioritizing mechanism applied to ADD/EDD heuristic focuses on finishing processing of jobs before their due dates to avoid tardiness. Since this heuristic does not consider total remaining processing time, it was expected to have good performance on percent tardy only, but not on percent early and percent on time. This heuristic is an improved version of EDD/EDD which was one of the best performing heuristics when early shipments were allowed (Russell and Philipoom, 1991). ADD/EDD was selected here to compare with other heuristics which contain the feature of forbidden early shipment (e.g., ACR/CR).

ACR/CR Heuristic

ACR/CR selects the subfamily queue with the smallest average critical ratio (ACR) and then utilizes the critical ratio (CR) rule to sequence jobs (minimum CR first) within the subfamily queue. ACR (average critical ratio) sums N_{qc} (a maximum value of 5 jobs) smallest critical ratios in a subfamily, then this total is divided by N_{qc}. The critical ratio (CR) rule is defined as the ratio of time remaining until due date to total remaining processing time. Mathematically, ACR (average critical ratio) queue selection rule and CR (critical ratio) job dispatching rule can be defined as:

ACR queue selection rule:

$$\text{Minimum} \quad \left(\sum_{i=1}^{N_{qc}} CR_{mqi} \right) / N_{qc} \quad \text{-> Priority queue}$$

CR job dispatching rule:

$$\text{Minimum} \quad CR_i = (D_i - t) / \left(\sum_{j \varepsilon \phi} P_{ij} \right) \quad \text{-> Priority job}$$

Where:

ϕ = Set of uncompleted operations
D_i = Due date of job i
N_{qc} = Number of jobs in queue q used to calculate ACR (a maximum value of 5)
P_{ij} = Processing time of operation j of job i
CR_i = Critical ratio of job i at time t
CR_{mqi} = Critical ratio of job i within queue q at workcenter m

ACR/CR considers both jobs' due dates and total remaining processing times. The prioritizing mechanism applied to ACR/CR focuses on hitting jobs' due dates to avoid tardiness and earliness. Since this heuristic considers the feature of forbidden early shipment, it was expected to have good performance on the measure of the percentage of orders on time. Previous research on job shops or flow shops (e.g., Scudder and Hoffmann, 1989, Scudder et al., 1990) showed that CR was the best performer when early shipments were forbidden. Results of these studies for the forbidden early shipment environment can be used to compare the results from this research.

ASLK/SLK Heuristic

ASLK/SLK selects the subfamily queue with the smallest average slack (ASLK) and then utilizes the slack (SLK) rule to sequence jobs (minimum SLK first) within the subfamily queue. ASLK (average slack) sums N_{qs} (a maximum value of 5 jobs) smallest slacks in a subfamily, then this total is divided by N_{qs}. The slack (SLK) rule is defined as the difference between time remaining until due date and total remaining processing time. Mathematically, ASLK (average slack) queue selection rule and SLK (slack) job dispatching rule can be defined as:

ASLK queue selection rule:

$$\text{Minimum} \quad \left(\sum_{i=1}^{N_{qs}} S_{mqi} \right) / N_{qs} \quad \text{-> Priority queue}$$

SLK job dispatching rule:

$$\text{Minimum} \quad S_i = (D_i - t) - \left(\sum_{j \varepsilon \phi} P_{ij} \right) \quad \text{-> Priority job}$$

Where:

ϕ = Set of uncompleted operations
D_i = Due date of job i
N_{qs} = Number of jobs in queue q used to calculate ASLK (a maximum value of 5)
S_i = Slack of job i at time t
P_{ij} = Processing time of operation j of job i
S_{mqi} = Slack of job i within queue q at workcenter m

Like ACR/CR, ASLK/SLK considers both jobs' due dates and total remaining processing times. The prioritizing mechanism applied to ASLK/SLK focuses on hitting jobs' due dates to avoid tardiness and earliness. Since this heuristic considers the feature of forbidden early shipment, it was expected to have good performance on the measure of the percentage of orders on time.

NEQA/EQ Heuristic

First, the queue selection rule "NEQA" selects the subfamily queue with the largest number of empty queues for the same type of queues ahead of the current workcenter. This rule breaks ties by selecting the subfamily with the smallest number of jobs in the same type of queues ahead.

At the last workcenter, the subfamily with the largest number of jobs in it is selected. Then, the job dispatching rule "EQ" is used to sequence jobs within the subfamily queue by selecting the jobs which will go to the empty queues ahead. If there are no empty queues ahead, the critical ratio (CR) rule is used to sequence jobs. Mathematically, NEQA determines the queue priority (at any workcenter except the last) as:

$$\text{Maximum} \left(\sum_{m \varepsilon \phi_1} I_{mq} \right) \rightarrow \text{Priority queue}$$

Rule for breaking ties:

$$\text{Minimum} \left(\sum_{m \varepsilon \phi_1} NB_{mq} \right) \rightarrow \text{Priority queue}$$

Where:

ϕ_1 = Set of workcenters ahead
I_{mq} = Empty queue index for queue q at workcenter m (I=1: empty, I=0: not empty)
NB_{mq} = Number of jobs within queue q at workcenter m

NEQA/EQ was selected here in an attempt to combine the workcenter status ahead of the current workcenter (included in queue selection rule) and job information (i.e., jobs' routings, included in job dispatching rule). The prioritizing mechanism applied to NEQA/EQ focuses on reducing shop congestion by processing jobs which are expected to go into empty queues for the next operation.

NLQB/CR Heuristic

First, the queue selection rule "NLQB" selects the subfamily queue with the largest number of lengthy queues for the same type of queues behind the current workcenter. A lengthy queue was defined as a queue which had 5 jobs or more than 5 jobs in it. This rule breaks ties by selecting the subfamily with the largest number of jobs in the same type of queues behind. At the first workcenter, the subfamily with the largest number of jobs in it is selected. Then, the job dispatching rule "CR" (critical ratio) is used to sequence jobs within the subfamily queue. Mathematically, NLQB determines the queue priority (at any workcenter except the first) as:

$$\text{Maximum} \left(\sum_{m \varepsilon \phi_2} J_{mq} \right) \rightarrow \text{Priority queue}$$

Rule for breaking ties:

$$\text{Maximum} \left(\sum_{m \varepsilon \phi_2} NB_{mq} \right) \rightarrow \text{Priority queue}$$

Where:

ϕ_2 = Set of workcenters behind
J_{mq} = Lengthy queue index for queue q at workcenter m
(J=1: queue length $\geq$ 5, J=0: queue length < 5)
NB_{mq} = Number of jobs within queue q at workcenter m

NLQB/CR was selected here in an attempt to combine the workcenter status behind the current workcenter (included in queue selection rule) and job information (i.e., critical ratio, included in job dispatching rule). Since this heuristic considers the critical ratio in its job dispatching rule, it was expected to have moderate to good performance on the measure of the percentage of orders on time.

In order to illustrate the mechanisms of the five group scheduling heuristics discussed above, examples are provided in Appendix C. All five heuristics developed in this research were non-exhaustive. These heuristics attempt to minimize the number of setups by not switching processing to another subfamily until a maximum of five jobs (called queue truncation criterion) in the current subfamily has been processed.

The value of five jobs was used in several places such as queue truncation criterion, lengthy queue, and queue selection rules (e.g., ADD, ACR, and ASLK) in this chapter. This value (i.e., five jobs) was selected because the maximum average queue length (averaging over all queues in the system) was about 5 which happened at the high level of average annual workload and low level of cell transfer batch. Other values such as twice or three times five jobs were tested in pilot runs (see Appendix F) and the principle to choose this value is to avoid continuing processing jobs within a queue with very long queue length (e.g., a length of 50).

Also, in order to incorporate the feature of workcenter sharing into the group scheduling heuristics, a cell selection rule was applied before a queue selection rule was used at the shared workcenter. The cell selection rule used at the shared workcenter was to select the cell with the largest number of jobs in the cell.

In the shop model (see Figure 4.2) there were three queues for each of the first four workcenters in both cells and six queues for the last (shared) workcenter. Each queue was dedicated to a subfamily. For example, queue 1 (i.e., Q1 in Figure 4.2) in the first cell (i.e., cell A) was used to store the jobs belonging to the first subfamily.

Performance Measures

The performance of a cellular system can be measured in four ways: how efficiently the orders are processed through the system, how well the orders meet the promised due dates, how much inventory exists in the system, and what profit results from processing the orders. While the first three ways have always been considered in previous research on group scheduling in cellular manufacturing, the fourth way has been ignored for the most part.

To meet the above four ways ten performance measures, which include an economically based measure and nine time-based or inventory-based measures, were selected in this research. The following is the listing of measures selected: (1) average time in system, (2) average waiting time in queue, (3) average net present value, (4) average work-in-process, (5) average number of orders in system, (6) percentage of orders tardy, (7) percentage of orders early, (8) percentage of orders on time, (9) average order tardiness, and (10) average order earliness. The definitions of these measures are shown below:

(1) The time in system for an order can be defined as the difference between order shipment time and order release time. This definition is based on previous research on manufacturing systems with forbidden early shipment (e.g., Lawrence, 1991). The average time in system per order can be expressed as:

$$\text{Average Time in System per Order} = \frac{\sum_{i=1}^{N} (T_i - RT_i)}{N}$$

(2) The waiting time in queue for a job (i.e., a transfer batch) is the amount of time that this job waited in the queues. The average waiting time in queue per job can be expressed as:

$$\text{Average Waiting Time in Queue per Job} = \frac{\sum_{j=1}^{N_j} WT_j}{N_j}$$

(3) The average net present value per order which is an economically based (or monetary) measure can be expressed as:

$$\text{Average Net Present Value per Order} = \frac{\sum_{i=1}^{N} NPV_i}{N}$$

(4) The average work-in-process is a time-persistent measure. This measure includes the orders which have been released and are waiting for processing or being processed.

(5) The average number of orders in the system is a time-persistent measure. This measure includes work-in-process and the finished orders that are held in the storage waiting for shipment.

(6) The percentage of orders tardy is the ratio of total number of tardy orders shipped to total number of orders shipped, as shown below:

$$\text{Percentage of Orders Tardy} = \frac{N_t}{N} \times 100$$

(7) The percentage of orders early is the ratio of total number of early orders to total number of orders shipped, as shown below:

$$\text{Percentage of Orders Early} = \frac{N_e}{N} \times 100$$

(8) The percentage of orders on time is the ratio of total number of on time orders to total number of orders shipped, as shown below:

$$\text{Percentage of Orders on Time} = \frac{N_o}{N} \times 100$$

(9) The tardiness for an order which is completed after its due date can be defined as the difference between order completion time and order due date. The average order tardiness can be expressed as:

$$\text{Average Tardiness} = \frac{\sum_{i=1}^{N_t} (C_i - D_i)}{N_t}, \text{ where } C_i \geq D_i$$

(10) The earliness for an order which is completed before its due date can be defined as the difference between order due date and order completion time. The average order earliness can be expressed as:

$$\text{Average Earliness} = \frac{\sum_{i=1}^{N_e} (D_i - C_i)}{N_e}, \text{ where } D_i \geq C_i$$

Where:

N = Total number of orders shipped
C_i = Completion time of order i

D_i = Due date of order i
N_j = Total number of jobs processed
N_e = Total number of early orders shipped
N_o = Total number of on time orders shipped
N_t = Total number of tardy orders shipped
T_i = Shipping time of order i
RT_i = Release time of order i
WT_j = Queue waiting time of job j
NPV_i = Net present value of order i

Due to the complexity of the net present value measure, the remainder of this section describes the contents of this measure. The net present value measure used in this research was based on two studies (Scudder et al., 1990, Rohleder and Scudder, 1992) which examined scheduling rules in random job shops with forbidden early shipment. The net present value for each order should be calculated for four components: the present value of the cash outflows associated with material and labor costs (PV1), out-of-pocket inventory holding costs (PV2), a tardiness penalty (PV3), and the present value of the payment for the order (PV4). Mathematically, the components of an order's net present value can be defined as follows:

1) The present value of the material and labor (including setup and processing) costs for order i (as of the order's release time at time 0):

$$PV1_i = W_i + \sum_{j=1}^{M_i} ((V_{ij} + U_{ij}) \exp(-rt_{ij}))$$

2) The present value of the out-of-pocket holding costs for order i:

$$PV2_i = (W_i \exp(hT_i) - W_i) \exp(-rT_i) + \sum_{j=1}^{M_i} ((V_{ij} + U_{ij}) \exp(h(T_i - t_{ij})) - (V_{ij} + U_{ij})) \exp(-rT_i)$$

3) The present value of the tardiness penalty for order i:

$$PV3_i = \pi R_i (C_i - D_i) \exp(-rC_i), \quad \text{if } C_i > D_i$$

$$\text{Where, } R_i = (1 + f)(W_i + \sum_{j=1}^{M_i} U_{ij})$$

4) The present value of the revenue (which is received at time T_i) for order i:

$$PV4_i = R_i \exp(-rT_i)$$

To sum up the four components, the net present value for order i can be expressed as:

$$NPV_i = - PV1_i - PV2_i - PV3_i + PV4_i$$

The average net present value per order, by averaging over all orders shipped, can be expressed as:

$$\text{Average Net Present Value} = \frac{\sum_{i=1}^{N} NPV_i}{N}$$

Where:

f = Profit margin (percent of total undiscounted order cost excluding setup charge)
h = Annual out-of-pocket holding cost rate
N = Total number of orders shipped
r = Annual interest rate (continuous compounding)
π = Tardiness penalty cost rate (percentage of order revenue per year)
C_i = Completion time of order i
D_i = Due date of order i
M_i = Number of operations of order i
R_i = Undiscounted revenue of order i
T_i = Shipping time of order i
W_i = Material cost of order i
t_{ij} = Time when operation j of order i is started
U_{ij} = Labor processing charge for operation j of order i
V_{ij} = Labor setup charge for operation j of order i
NPV_i = Net present value for order i

The economic parameters (or factors) required to calculate the net present value for an order include the raw material cost, labor processing cost, labor setup cost, interest rate, out-of-pocket holding cost rate, profit margin percentage, and tardiness penalty percentage. In this study, most of the data for these economic parameters were adopted from Rohleder and Scudder's paper (1992). Raw material costs varied uniformly between \$50/part and \$100/part, while labor processing and setup costs were charged \$9/hour and \$15/hour, respectively. The annual interest rate and out-of-pocket holding cost rate were set to 15% and 20%, respectively. The profit margin used to calculate revenue was set to 95%. The tardiness penalty cost was charged the amount of revenue per year, i.e., a tardy/early cost ratio of 5.

Experimental Design Considerations

A Full Factorial Design

Based on the discussion in the section titled "Identification of Factors Impacting Performance of Heuristics" in this chapter, it was apparent that a number of factors could have been legitimately included in this investigation. It was ultimately decided to study four factors which were expected to have a major impact on the performance of the system defined in this research. These four factors were group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch. As can be seen from Table 4.1, five group scheduling heuristics and two levels of each of the other three factors were investigated.

A full factorial design was used in this research, with all factors crossed. Thus, this was a 5 x 2^3 full factorial design with 40 experiments (i.e., system configurations or treatment combinations). Eight experimental conditions (which are the combinations of the three factors: average annual workload, demand pattern variability, and cell transfer batch) as shown in Table 4.2 were carried out to test all 40 experiments, that is, five group scheduling heuristics were tested in each experimental condition.

TABLE 4.1

EXPERIMENTAL FACTORS UTILIZED

Factor	Level	Description
Group Scheduling Heuristic (GSH)		ADD/EDD Heuristic
		ACR/CR Heuristic
		ASLK/SLK Heuristic
		NEQA/EQ Heuristic
		NLQB/CR Heuristic
Average Annual Workload (AAW)	High	90% of System Capacity
	Low	80% of System Capacity
Demand Pattern Variability (DPV)	High	Avg. Annual Orders: 21 & Avg. Order Size: 34
	Low	Avg. Annual Orders: 40 & Avg. Order Size: 17
Cell Transfer Batch (CTB)	High	Arriving Order Size (OS)
	Low	Standard Container Size (CS, 10 parts/container)

TABLE 4.2

EXPERIMENTS CONDUCTED

GSH	AAW=High				AAW=Low			
	DPV=High		DPV=Low		DPV=High		DPV=Low	
	CTB =OS	CTB =CS	CTB =OS	CTB =CS	CTB =OS	CTB =CS	CTB =OS	CTB =CS
ADD/EDD	Exp. Cond. 1	Exp. Cond. 2	Exp. Cond. 3	Exp. Cond. 4	Exp. Cond. 5	Exp. Cond. 6	Exp. Cond. 7	Exp. Cond. 8
ACR/CR								
ASLK/SLK								
NEQA/EQ								
NLQB/CR								

Selection of Research Vehicle

After specifying the basic elements of the study, it was necessary to select an appropriate research vehicle for conducting the various experiments in this research effort. Analytical techniques were not considered for this research because of the complexity and dynamic features of the scheduling system defined and group scheduling heuristics used. Computer simulation was selected for use in this research for the following four reasons: First, in simulation models, the researcher can change those factors of interest faster and hold the other factors constant. Second, computer simulation allows the specification of assumptions to the discretion of the researcher. Third, simulation models can be developed rapidly using any one of the many discrete event simulation languages available. Finally, simulation allows the researcher to examine the performance of various scheduling heuristics over a long time frame.

In particular, the SLAM II (Simulation Language for Alternative Modeling) language (Pritsker, 1986) was used to develop the simulation model utilized in this research effort. SLAM II is a high-level FORTRAN-based simulation language which provides process, discrete event, and continuous modeling capabilities. The process modeling approach was used in this research. In the process modeling, SLAM II employs a "network" structure which consists of specialized symbols called nodes and branches. The entities in the system flow through the network model. In addition, user-written FORTRAN subprograms can be developed by the modeler to perform the more detailed or complex tasks such as scheduling heuristics.

Making a Fair Comparison

The next issue in experimental design considerations was how to fairly compare the performance among different experiments (or system configurations). The problem was that these system configurations might have different numbers of parts processed and machine workloads for a given period of time. It was concluded that in order to fairly and consistently compare the different system configurations, annual demand for individual part type and average annual workload for individual workcenter should be held constant when the level of average annual workload has been specified. Holding part demands and machine workloads constant for different system configurations could ensure that the effect observed when varying the different experimental factors could be isolated and not masked by changes in part demands or machine workloads (Farrington, 1991).

Because the system designs of all system configurations in this study were sufficiently similar, they should be simulated with common random numbers (i.e., correlated sampling) in such a way that the models behaved similarly. In order to make correlated sampling more likely to yield a positive correlation, three guidelines were followed. First, a random stream was dedicated to producing the random numbers for each particular type of input random variate (i.e., each specific purpose). Second, independently chosen seeds were assigned to each stream at the beginning of each run (see Appendix E). Third, all random numbers required for an order were generated at the time of arrival instead of when the order actually needs them, and stored as attributes of the order (Banks and Carson, 1984, Law and Kelton, 1991).

Data Gathering versus Data Generation

After settling the fair comparison issue, the data gathering versus data generation issue had to be addressed. While it would have been desirable to use "real" data, it was not feasible in this situation. As was just mentioned, a fair comparison requires common part demands and machine workloads for different system configurations. It was highly unlikely that any two systems or firms in the real world would have identical part demands and machine workloads. Thus, due to the number of factors that were controlled, generation of the input parameters (e.g., annual part demands, the numbers of orders per year, routings, processing times, and major setup times, etc.)

was the logical conclusion (Farrington, 1991).

Number of Replications

Due to the stochastic nature of the simulation model the observed performance of the system is only an estimate of the mean of the true performances. Therefore, when comparing various system configurations, it is critical to determine how much of the difference in system performances is due to the experimental factors (e.g., group scheduling heuristic) and how much is simply error introduced by the stochastic nature of the simulation. This requires some measure of variability of the estimates to construct confidence intervals and, thus, multiple observations (i.e., several replications in this study) are required (Mahmoodi, 1989).

The procedure discussed in Law and Kelton (1991, p. 537) was used to determine the number of replications required in the experiments.

Step 1. We need to choose a system configuration and then estimate the mean and variance of a specific performance measure (e.g., work-in-process) based on a fixed number of replications (n). The following configuration which had the largest variations on most measures in simulation pilot runs (see Appendix F) was chosen:

Group scheduling heuristic: ASLK/SLK
Average annual workload: High level
Demand pattern variability: High level
Cell transfer batch: Arriving order size

Ten observations (i.e., n=10 replications) of the work-in-process measure were collected with a run length of five years, and the sample mean and variance were calculated, as shown below:

Observations (X): 28.37 40.39 27.27 33.91 30.11
35.77 32.39 31.80 32.35 30.15

Sample mean ($\overline{X}$) = 32.251

Sample variance ($S(X)^2$) = 14.498

Step 2. We assumed that $S(X)^2$ would not change as the number of replications increased, an approximate expression for the number of replications $n^*(\beta)$, required to obtain an absolute error of β is given by

$$n^*(\beta) = \min \{ i \geq n : t_{i-1,\alpha/2} \cdot [S(X)^2/i]^{1/2} \leq \beta \}.$$

We can determine $n^*(\beta)$ by iteratively increasing i by 1 until a value of i is obtained for which $t_{i-1,\alpha/2} \cdot [S(X)^2/i]^{1/2} \leq \beta$. The absolute error β can be defined as $|\overline{X} - \mu|$, where, μ is the population mean. If we used a confidence coefficient of 90% (i.e., $\alpha = 0.10$) and assumed that β was equal to 5% of the sample mean (i.e., 1.6126), the number of replications $n^*(\beta)$ required was 18.

The same procedure was applied to all other measures. The numbers of replications required for all measures were listed below (also see Appendix F):

Time in system:	12 replications
Queue waiting time:	24 replications
Net present value:	13 replications
Work-in-process:	18 replications
# of jobs in system:	10 replications
Percent tardy:	25 replications
Percent early:	17 replications
Percent on time:	20 replications
Tardiness:	25 replications
Earliness:	12 replications

The worst case (i.e., 25 replications) was then used in this study based on a α value of 0.10 and β values which were 5% of the sample means.

Steady State versus Terminating Simulation

A steady state simulation is a simulation whose objective is to study long-run, or steady state, behavior of a nonterminating system. The major issue when simulating a steady state system is to determine when the system is in steady state so as to identify an appropriate warm-up (or start-up) period. A terminating simulation is a simulation that runs for some duration of time. When simulating a terminating system, the initial conditions of the system at time 0 must be specified and the stopping time or event must be defined (Banks and Carson 1984).

It was decided that the experiments should be run as terminating simulations in this study. This decision was based on the characteristics of current manufacturing environments. With the current emphasis in industry on flexibility and reduction in product life cycles, it is unrealistic to assume that conditions are constant for that long a period of time (Farrington, 1991). Kleijnen (1987) argued that in practice most simulations are terminating, whereas in academic studies many simulations are assumed to be nonterminating.

It was assumed that we were interested in the entire process of the cellular system from its starting operation (fully loaded) to its termination. The initial conditions of the system at time 0 were assumed to be idle, but not empty (i.e., machines were idle, but queues might not be empty). The queue states at time 0 were determined in a simulation pilot run (see Appendix F). The pilot run which used ACR/CR heuristic was stopped when the system was fully loaded (i.e., 6 months). The resulting queue states were used as the initial conditions at time 0 for each of the 40 experiments. Jobs which were in the queues at time 0 would not be collected for statistical calculations.

Several situations can result in the termination of a cellular manufacturing system. A frequently happened situation is when a cell reorganization is required due to the changes in product design, mix, or demand. In this situation, the operation of the system should be terminated and a costly cell reorganization which includes modifying cell layouts and selecting part families are required (Sassani, 1990). The stopping time (e.g., the time for cell reorganization) of the cellular system was assumed to be five years in this study. Jobs which have arrived at the system before the stopping time would be finished and their data would be collected for statistical calculations.

Statistical Analysis Procedures

Since correlated sampling (i.e., common random numbers) was used to simulate the models, the paired t-test should be utilized to compare the difference in any two system configurations in this study. It should be noted that to use the paired t-test, the number of observations (i.e., the number of replications) between the two system configurations compared should be the same and we must assume that the distribution of the difference of two means is normal (Banks and Carson, 1984).

In addition to the paired t-test, the analysis of variance (ANOVA) should be used to examine the effects of the experimental factors on the performance of group scheduling heuristics. This method allows the examination of both the individual effect of each of the experimental factors on the performance measures as well as the degree to which the experimental factors interact.

Data Generation Procedures

Given the system and the number of experimental factors to be controlled in this research, the data generation process was a difficult issue. The basic input data required for this simulation model included the annual demands, number of orders per year, average order sizes, average annual setup and processing workloads, total annual workloads, routings, processing times, and major setup times. The last two input data (i.e., processing times and major setup times) were generated by the subprogram INTLC in the simulation program, while other input data were generated by a SAS (SAS, 1985) program (see Appendix A).

Generation of Annual Demands, Annual Orders, and Average Order Sizes

Generation of the annual demands for the individual part types was accomplished by taking thirty random samples from a Uniform (100,1500) distribution, one for each part type in the system. The annual orders (i.e., the number of orders per year) were then generated by sampling from a Uniform (30,50) for low demand pattern variability and Uniform (10,30) for high demand pattern variability. As has been previously noted, the high level of demand pattern variability (i.e., infrequent large orders) had about 21 orders per year, while the low level of variability (i.e., frequent small orders) had about 40 orders per year (see Appendix A).

After these two parameters (i.e., annual demands and annual orders) have been generated, the average order sizes and average interarrival times could be calculated for all part types. As is shown below, the average order size for part type i was equal to the annual demand for part type i divided by the number of orders per year.

Average order size for part type i =
(Annual demand for part type i) /
(Annual orders for part type i)

The average interarrival time for part type i was the number of minutes per year (i.e., 120,000 minutes per year) divided by the number of orders per year.

Average interarrival time for part type i =
(120,000 minutes per year) /
(Annual orders for part type i)

Next, the average annual workload for each workcenter in the system was generated. Average annual workload (AAW) across all workcenters in the system was, on average, 90% (for high workload) or 80% (for low workload) of total capacity of the system. If it is assumed that there are 2000 hours/year for a one shift operation, the average annual workload is 1800 (for high workload) or 1600 (for low workload) hours/year. The average annual workload across all workcenters was, in fact, made up of two components: average annual setup workload (which is 10% of AAW) and average annual processing workload (which is 90% of AAW). Thus, the average annual setup and processing workloads for workcenter m ($AASW_m$ and $AAPW_m$), expressed in hours per year, were assigned by randomly sampling from Normal $(180,9^2)$ and Normal $(1620,81^2)$ (for high workload) or Normal $(160,8^2)$ and Normal $(1440,72^2)$ (for low workload) distributions, respectively. The average annual workload for workcenter m was the sum of $AASW_m$ and $AAPW_m$. The average annual workloads, average annual setup workloads, and average annual processing workloads for all workcenters are shown in Appendix A.

Generation of Routing Table

Orders were processed by between three and five operations with one operation on one workcenter. Orders could enter the system from one of the first three workcenters (A1, A2, A3, B1, B2, or B3) and exit from one of the last three workcenters (A3, A4, B3, B4, or 5). The number of operations for each part type was assigned by randomly sampling from a Uniform (3,5) distribution. The workcenter numbers were also assigned by randomly sampling from a Uniform (1,5) distribution. Therefore, the routing for an order was dependent on its part type (see Appendix A).

Generation of Processing Times

The information required to generate the processing times included the routing table, annual demand for each part type (AD_n), average annual processing workload for each workcenter ($AAPW_m$), and coefficient of variation of processing time (CV_{pt}). First of all, an initial average processing time for part type n ($IAPT_n$) was generated by randomly sampling from a Uniform (5,15) distribution (in minutes per part). The initial average processing time for each part type provided a starting point for generation of processing times.

Next, the initial processing time (IPT_{nm}) for part type n on workcenter m was generated by sampling from a Normal ($IAPT_n$, $IAPT_n*CV_{pt}$) distribution. The initial processing workload for part type n on workcenter m (IPW_{nm}) was calculated by multiplying the initial processing time (IPT_{nm}) by the annual demand (AD_n), as shown below:

$$IPW_{nm} = IPT_{nm} * AD_n.$$

The total initial processing workload for workcenter m ($TIPW_m$) was then calculated by summing the initial workloads (IPW_{nm}) over all part types processed on this workcenter (ϕ_m), as shown below:

$$TIPW_m = \sum_{n \varepsilon \phi_m} IPW_{nm}$$

The final processing time for part type n on workcenter m (PT_{nm}), expressed in minutes per part, was then calculated, via a normalization process, by multiplying the initial processing time (IPT_{nm}) by 60 and the ratio of average annual processing load ($AAPW_m$) to total initial processing workload ($TIPW_m$), as shown below:

$$PT_{nm} = IPT_{nm} * (AAPW_m / TIPW_m) * 60.$$

Generation of Major Setup Times

The information required to generate the major (or subfamily) setup times included the routing table, number of transfer batches required per year for each part type (ATB_n), and average annual setup workload for each workcenter ($AASW_m$). First of all, an initial setup workload for part type n on workcenter m (ISW_{nm}) was assigned by multiplying average annual setup workload for workcenter m ($AASW_m$) by a value which was randomly sampling from a Uniform (0.2,1) distribution (%T), as shown below:

$$ISW_{nm} = AASW_m * \%T.$$

Note that the lower limit was set slightly larger than zero to ensure that final setup times were not unrealistically small. Next, the total initial setup workload for workcenter m ($TISW_m$) was calculated by summing the initial setup workload (ISW_{nm}) for all part types processed on workcenter m (ϕ_m), as shown below:

$$TISW_m = \sum_{n \varepsilon \phi_m} ISW_{nm}.$$

The annual setup workload for part type n on workcenter m (ASW_{nm}) was calculated, via a normalization process, by multiplying the initial setup workload (ISW_{nm}) by the ratio of average annual setup load ($AASW_m$) to total initial setup workload ($TISW_m$), as shown below:

$$ASW_{nm} = ISW_{nm} * (AASW_m / TISW_m).$$

The setup time for part type n on workcenter m (ST_{nm}), expressed in minutes per batch, was calculated by dividing the annual setup workload for part type n on workcenter m (ASW_{nm}) by the number of transfer batches required per year for part type n (ATB_n) and multiplying by 60, as shown below:

$$ST_{nm} = (ASW_{nm} / ATB_n) * 60.$$

The major (or subfamily) setup time for subfamily q on workcenter m, FST_{qm}, was then obtained by averaging the setup times, ST_{nm}, over all part types belonging to this subfamily. A subfamily setup is required only when a new subfamily is selected and the subfamily setup time will be added to the first job processed.

Simulation Model

A network simulation model was created using SLAM II (Pritsker, 1986). In addition, several FORTRAN subprograms were written to perform more detailed or complex tasks such as scheduling heuristics. The SLAM II network model (graphic model) and the listing of the entire simulation programs are shown in Appendixes D and E, respectively. First, the functions of all FORTRAN subprograms are listed below:

SUBROUTINE INTLC: This subprogram was called by SLAM before each simulation run. It was used to set initial variables, to entry jobs at time 0, and to generate the processing times and major setup times for all part types on each workcenter.

SUBROUTINE OTPUT: This subprogram was called by SLAM at the end of each simulation run. It was used to perform non-standard end-of-run processing and output reporting.

SUBROUTINE EVENT: This subprogram was used to assign (or generate) basic data to an arriving order. These data included part type number, subfamily number, mean interarrival time, order size, due date, release time, number of transfer batches required, and processing times and major setup times on the workcenters in its routing.

SUBROUTINE USERF: This subprogram was used to calculate net present value for an order before shipping.

FUNCTION NQS: This subprogram was used to execute the desired queue selection logic and returns the file number of the selected queue to SLAM. Five queue selection logics (i.e., ADD, ACR, ASLK, NEQA, and NLQB) were used in this study.

SUBROUTINE SELJOB: This subprogram was used to execute the desired job selection logic within a queue which was selected by the subprogram NQS. Four job selection logics (i.e., EDD, CR, SLK, and EQ) were used in this study.

The network model (see Appendix D) described both the elements and the operational process (or procedures) when orders flowed through the flow shop cellular system. Thirty part types were created at the beginning of the network model (at CREATE nodes). There were two cells in the system; each cell contained five workcenters (cell A included ACTIVITY 1, 2, 3, 4, and 9, while cell B included ACTIVITY 5, 6, 7, 8, and 9), with the last workcenter (ACTIVITY 9) shared between two cells. The first fifteen part types were processed in the cell A, while other part types were processed in the cell B. An order's routing depended on its part type; an order was processed by between three and five operations, with one operation on each workcenter.

After an order has been created (at a CREATE node), the subprogram EVENT was called (at an EVENT node) to assign basic data (e.g., part type number, subfamily number, mean interarrival time, etc.) to this arriving order. There were two branches coming out of the EVENT node. One branch went back to its CREATE node. The other branch determined the waiting time required (attribute DELAY) before the order could begin its first operation in its routing.

Following the branching, the unbatch process (mainly at the UNBATCH node) was applied to divide the order into transfer batches (called jobs in this research). Then, each job entered the cell A or B for processing based on its part type and routing. The subprogram NQS(N) (where N was equal to the ACTIVITY number) was called (at a SELECT node) in front of each workcenter to execute the desired queue and job selection logics based on the group scheduling heuristic applied. There were three queues (i.e., QUEUE nodes) in front of each workcenter except for the last workcenter which had six queues. Each queue was dedicated to a subfamily (i.e., five part types). For example, QUEUE 1 in front of the first workcenter in the cell A (i.e., ACTIVITY 1) was used to store the jobs belonging to the first subfamily (i.e., part types 1 to 5).

After the job has finished processing on the workcenters in its routing, it would enter a BATCH node. At the BATCH node, jobs were accumulated into their original orders. Then, each order's completion time (attribute TCOMP) was checked with its due date (attribute DUEDATE). If the order was completed before its due date, it would wait in storage until its due date which was controlled by a QUEUE node and an ACTIVITY. If the order was completed at or after its due date, it would be shipped immediately. Finally, some performance data (e.g., time in system) were collected at the end of the network model. The net present value of the order was calculated by calling the subprogram USERF (at an ASSIGN node).

Model Verification and Validation

Verification is the process of comparing the conceptual model with the simulation program that implements that conception. Validation, on the other hand, is the process of checking of the simulation model against reality for the intended application. Verification and validation should begin at the onset of the model constructing process and continue throughout the study. Actually, simulation model construction, verification, and validation often are in a dynamic, feedback loop. Although the concepts of verification and validation are different, in practice they may overlap to a considerable extent (Carson, 1989, Bratley et al., 1987).

The following techniques (and their combinations) were used to verify and/or validate the simulation model in this study: documentation, structured programming and modular testing, debugging (i.e., to include additional checks and outputs in the program that would point out the bugs), sensitivity analysis, traces, input-output transformation, testing deterministic models, testing simplified cases. A brief description of part of the test runs by using traces, input-output transformation, deterministic models, and simplified cases was presented below (also see Appendix F).

Test 1. This test set a run length of 5 years with 10 replications. Additional COLCT nodes were added to network to collect the statistical data for each part type. The purpose of this test was to check the total number of orders shipped per year, the number of orders shipped per year for each part type, and the number of orders shipped per year at each workcenter. All forty experiments were tested and, in general, the simulation results were within five percent of the expected values. For example, the results showed that the average numbers of orders shipped per year (by averaging over five heuristics) were 622 and 1209 (the expected values were 619 and 1207) when the demand pattern variability was set to high level and low level, respectively.

Test 2. This test set a run length of 5 years with 10 replications. No setup avoidance was allowed and no variation of the average annual workload existed among nine workcenters. The purpose of this test was to check the average machine utilization across all workcenters in the system when every job needed a setup. All forty experiments were tested and, in general, the simulation results were within three percent of the expected values. For example, the results showed that the average machine utilizations by averaging over all heuristics across all workcenters were 89.17% and 79.96% (the expected values were 90% and 80%) when the average annual workload was set to high level and low level, respectively.

Test 3. This test released a single order from each CREATE node (i.e., each part type) to the system. The SLAM control statement "MONTR,TRACE" was used to trace the path and timing when jobs flowed through the network model. The statistical data such as time in system and net present value were collected. The trace reports have been carefully checked to ensure that the developed network model met the intended applications and the statistical data were correctly collected.

Test 4. This test released a single order from the first CREATE node to the system. Again, the SLAM control statement "MONTR,TRACE" was used to trace the duration that an order was held before beginning its first operation. The trace reports have been carefully checked to ensure that the order release mechanism in the network model was correctly implemented.

Test 5. This test released a single order from the first CREATE node to the system with either loose (i.e., an early order) or tight (i.e., a late order) due date. The SLAM control statement "MONTR,TRACE" was used to trace the path and timing after an order has finished all operations. The trace reports have been carefully checked to ensure that the forbidden early shipment mechanism in the network model was correctly implemented.

Test 6. This test released a single order from the first CREATE node to the system when the cell transfer batch was set to low level (i.e., used standard container with the capacity of 10 parts/container). The SLAM control statement "MONTR,TRACE" was used to trace the unbatch and batch processes. The trace reports have been carefully checked to ensure that the unbatch and batch mechanisms in the network model were correctly implemented.

Test 7. to 11. These tests released orders, every 15 minutes, from the first, sixth, and eleventh CREATE nodes to the system. In each test, one of the five group scheduling heuristics (i.e., ADD/EDD, ACR/CR, ASLK/SLK, NEQA/EQ, and NLQB/CR) was applied to select a queue and to sequence the jobs within this queue. The SLAM control statement "MONTR,TRACE" was used to trace the status of all queues and activities in the network model. The trace reports have been carefully checked to ensure that the logic for the five group scheduling heuristics in the subprograms NQS and SELJOB was correctly implemented.

CHAPTER V

ANALYSIS AND INTERPRETATION OF RESULTS

Introduction

This chapter presents the data obtained from simulation experiments, the statistical analysis of these data, and the interpretation. First, a summary of the experimental results (i.e., data obtained from simulation experiments) is presented. Second, pairwise comparisons by using the paired t-test analysis were utilized to rank the five group scheduling heuristics with respect to each of the ten performance measures under each of the eight experimental conditions. Third, the analysis of variance (ANOVA) was used to gain a better understanding of the effects of the experimental factors and their interactions with respect to each of the performance measures. Those effects that exhibit statistical significance are then presented graphically and discussed in detail. Finally, conclusions are drawn from the results of the statistical analysis procedures (i.e., the paired t-test analysis and the analysis of variance).

Terminology

As an aid for understanding, a listing of the abbreviations of all of the terms used in the tables and figures presented throughout this chapter is shown in Table 5.1. Also, we term the average annual workload, demand pattern variability, and cell transfer batch as shop environmental factors, and the combinations of these three shop environmental factors as shop environmental conditions (or experimental conditions).

TABLE 5.1

LISTING OF TERMS AND THEIR ABBREVIATIONS

Term	Abbreviation
Group Scheduling Heuristic	GSH or G
Average Annual Workload	AAW or A
Demand Pattern Variability	DPV or D
Cell Transfer Batch	CTB or C
Container Size	CS or Cs
Order Size	OS or Os
Low (Level)	L
High (Level)	H

Experimental Results

A summary of the experimental results (i.e., data obtained from simulation experiments) is presented in Table 5.2. In Table 5.2, the mean values of each of the ten performance measures for the five heuristics under each of the eight experimental conditions (or shop environmental conditions) are listed. Each mean value (e.g., average time in system of 8368 for ADD/EDD under experimental condition 1) was obtained by averaging the 25 values collected in the 25 replications of the experiment.

The units of the ten performance measures used in Table 5.2 are defined as follows:

Performance Measure	Unit
Average time in system	minutes/order
Average queue waiting time	minutes/transfer batch
Average net present value	$/order
Average work-in-process	orders
Average # of orders in system	orders
Average percentage of orders tardy	% of orders
Average percentage of orders early	% of orders
Average percentage of orders on time	% of orders
Average tardiness	minutes/order
Average earliness	minutes/order

Note that the average tardiness (or earliness) is the ratio of total tardiness (or earliness) to total number of tardy (or early) orders.

TABLE 5.2

SUMMARY OF EXPERIMENTAL RESULTS

Expr. Condition 1: AAW_High, DPV_High, CTB_Order Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	8368	5509	2456	34.75	41.40
ACR/CR	8387	5685	2454	35.61	41.49
ASLK/SLK	8536	5761	2453	35.99	42.23
NEQA/EQ	8959	5634	2434	35.38	44.34
NLQB/CR	8730	5638	2437	35.38	43.19

GSH	Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
ADD/EDD	42.33	52.93	4.74	4279	2477
ACR/CR	49.56	44.92	5.51	3612	2563
ASLK/SLK	44.92	50.21	4.87	4374	2442
NEQA/EQ	35.29	60.79	3.91	6916	2949
NLQB/CR	42.24	53.50	4.26	5154	2904

Expr. Condition 2: AAW_High, DPV_High, CTB_Container Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	9020	7027	2415	37.15	44.66
ACR/CR	8813	7056	2422	38.68	44.70
ASLK/SLK	9062	7205	2414	37.50	44.88
NEQA/EQ	10490	7548	2350	38.82	52.20
NLQB/CR	9293	7179	2400	39.34	47.24

GSH	Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
ADD/EDD	44.46	52.37	3.17	5382	2840
ACR/CR	51.83	41.82	6.35	4117	2820
ASLK/SLK	44.95	51.85	3.20	5413	2811
NEQA/EQ	29.38	68.41	2.21	13513	3946
NLQB/CR	48.26	47.12	4.62	5470	3326

TABLE 5.2 (Continued)

Expr. Condition 3: AAW_High, DPV_Low, CTB_Order Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	7055	5034	1282	57.61	69.41
ACR/CR	6938	5036	1282	57.62	68.25
ASLK/SLK	7101	5105	1281	58.31	69.87
NEQA/EQ	9019	6249	1242	69.52	88.68
NLQB/CR	7604	5380	1265	60.98	74.77

GSH	Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
ADD/EDD	42.63	51.25	6.12	3617	2306
ACR/CR	49.56	42.96	7.49	2854	2466
ASLK/SLK	43.45	50.42	6.14	3632	2293
NEQA/EQ	34.51	61.26	4.23	10402	3173
NLQB/CR	45.42	49.00	5.58	4635	2833

Expr. Condition 4: AAW_High, DPV_Low, CTB_Container Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	6888	5450	1285	53.40	67.78
ACR/CR	6705	5468	1289	55.50	67.87
ASLK/SLK	6910	5479	1285	53.88	68.02
NEQA/EQ	9050	6337	1236	65.62	89.36
NLQB/CR	7181	5589	1260	57.39	72.60

GSH	Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
ADD/EDD	35.93	58.40	5.68	3858	2467
ACR/CR	42.87	48.23	8.90	2765	2556
ASLK/SLK	36.37	57.78	5.85	3869	2451
NEQA/EQ	29.15	66.48	4.37	12477	3626
NLQB/CR	41.85	51.95	6.20	4001	2945

TABLE 5.2 (Continued)

Expr. Condition 5: AAW_Low, DPV_High, CTB_Order Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	6691	2786	2511	20.41	33.08
ACR/CR	6743	3068	2510	21.80	33.34
ASLK/SLK	6733	2916	2510	21.05	33.29
NEQA/EQ	6890	2861	2506	20.78	34.07
NLQB/CR	6874	2943	2504	21.18	33.99

GSH	Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
ADD/EDD	14.51	82.35	3.13	3554	3100
ACR/CR	20.02	77.52	4.88	2692	3089
ASLK/SLK	15.73	81.04	3.23	3517	3040
NEQA/EQ	16.98	79.88	3.14	4156	3353
NLQB/CR	18.89	75.10	3.59	3622	3330

Expr. Condition 6: AAW_Low, DPV_High, CTB_Container Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	6670	3128	2503	17.02	32.98
ACR/CR	6653	3294	2505	20.17	33.95
ASLK/SLK	6678	3147	2502	17.26	33.03
NEQA/EQ	7083	3130	2487	18.00	35.21
NLQB/CR	6846	3205	2497	19.97	34.88

GSH	Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
ADD/EDD	11.53	86.88	1.59	4464	3707
ACR/CR	16.09	78.78	5.12	2779	3521
ASLK/SLK	11.73	86.55	1.72	4448	3678
NEQA/EQ	14.91	83.01	2.08	6140	4184
NLQB/CR	18.09	78.04	3.87	3494	3849

TABLE 5.2 (Continued)

Expr. Condition 7: AAW_Low, DPV_Low, CTB_Order Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	5196	1743	1314	24.32	51.19
ACR/CR	5201	1918	1314	26.04	51.25
ASLK/SLK	5201	1766	1313	24.55	51.25
NEQA/EQ	5506	2014	1310	26.99	54.25
NLQB/CR	5369	1959	1311	26.45	52.90
GSH	**Average % Tardy (Orders)**	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
ADD/EDD	9.48	86.10	4.42	2657	3164
ACR/CR	12.64	80.47	6.89	1878	3174
ASLK/SLK	9.63	85.93	4.45	2668	3150
NEQA/EQ	15.67	79.60	4.72	3416	3471
NLQB/CR	14.98	79.44	5.57	2665	3374

Expr. Condition 8: AAW_Low, DPV_Low, CTB_Container Size

GSH	Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in Sys.
ADD/EDD	5163	1918	1314	20.65	50.87
ACR/CR	5153	2121	1315	24.27	52.35
ASLK/SLK	5165	1931	1314	20.86	50.92
NEQA/EQ	5448	1998	1310	23.39	53.92
NLQB/CR	5270	2079	1314	24.44	53.30
GSH	**Average % Tardy (Orders)**	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
ADD/EDD	6.71	90.11	3.17	3524	3401
ACR/CR	9.61	84.42	5.97	1970	3371
ASLK/SLK	6.77	90.06	3.17	3533	3384
NEQA/EQ	12.77	83.00	4.22	3844	3728
NLQB/CR	11.90	82.95	5.15	2470	3526

Paired t-test Analysis

By holding the three shop environmental factors constant (i.e., given an experimental condition or a shop environmental condition), the performance of the five heuristics can be compared (or tested) with respect to each of the ten measures. Because of the use of common random numbers (i.e., correlated sampling) for each experiment (i.e., each combination of the four experimental factors), the paired t-test is the proper method to analyze simulation results (Ruben et al., 1993). By using the paired t-test analysis, pairwise comparisons can be made between each pair of the five heuristics with respect to each of the ten performance measures under each of the eight experiment conditions (or shop environmental conditions).

It should be noted that to use the paired t-test, the number of observations (or replications) must be the same for each experiment although the variance between experiments may be different. The use of the paired t-test analysis can lead to a reduction in variance and thus to a smaller confidence interval (Banks and Carson, 1984, Law and Kelton, 1991). Given an experimental condition (or a shop environmental condition), if the observations of the i-th performance measure for any two heuristics are: Y_{i1j} and Y_{i2j} for j = 1,2,...,n (where n is the number of observations), we can pair Y_{i1j} with Y_{i2j} to define

$Y_{ij} = Y_{i1j} - Y_{i2j}$. Then, the following hypothesis can be tested:

H_o: $Y_{ij} = 0$ (no difference between the observations of the i-th measure for the two heuristics)

H_a: $Y_{ij} \neq 0$ (a difference between the observations of the i-th measure for the two heuristics)

The results of the paired t-test analysis are summarized in Table 5.3. A significance level of 0.10, which is frequently used in practice, was selected in this research. In Table 5.3, the five heuristics were ranked from best to worst (from top to bottom) with respect to each of the ten performance measures under each of the eight experimental conditions. For the following performance measures: the average time in system, average queue waiting time, average work-in-process, average number of orders in system, average percentage of orders tardy, average percentage of orders early, average tardiness, and average earliness, the ranking of the five heuristics was based upon minimum their means for each measure. For the following measures: the average net present value and average percentage of orders on time, the ranking of the five heuristics was based upon maximum their means for each measure.

The results of the paired t-test analysis with respect to each of the ten measures are discussed below, followed by the conclusions drawn from the paired t-test analysis.

TABLE 5.3

SUMMARY OF RESULTS OF PAIRED t-TEST ANALYSIS

Expr. Condition 1: AAW_High, DPV_High, CTB_Order Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ADD/EDD ┬	ADD/EDD	ADD/EDD ┬	ADD/EDD	ADD/EDD ┬
ACR/CR ┴	NEQA/EQ ┬	ACR/CR ┴┬	NEQA/EQ ┬	ACR/CR ┴
ASLK/SLK	NLQB/CR │	ASLK/SLK ┴	NLQB/CR │	ASLK/SLK
NLQB/CR	ACR/CR ┴	NLQB/CR ┬	ACR/CR ┴	NLQB/CR
NEQA/EQ	ASLK/SLK	NEQA/EQ ┴	ASLK/SLK	NEQA/EQ

Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK
NLQB/CR ┬	ASLK/SLK	ASLK/SLK ┬	ADD/EDD ┬	ADD/EDD
ADD/EDD ┴	ADD/EDD ┬	ADD/EDD ┴	ASLK/SLK ┴	ACR/CR
ASLK/SLK	NLQB/CR ┴	NLQB/CR	NLQB/CR	NLQB/CR
ACR/CR	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ

Expr. Condition 2: AAW_High, DPV_High, CTB_Container Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ACR/CR	ADD/EDD ┬	ACR/CR	ADD/EDD ┬	ADD/EDD ┬
ADD/EDD ┬	ACR/CR ┴	ADD/EDD ┬	ASLK/SLK ┴	ACR/CR │
ASLK/SLK ┴	NLQB/CR ┬	ASLK/SLK ┴	ACR/CR ┬	ASLK/SLK ┴
NLQB/CR	ASLK/SLK ┴	NLQB/CR	NEQA/EQ ┴	NLQB/CR
NEQA/EQ	NEQA/EQ	NEQA/EQ	NLQB/CR	NEQA/EQ

Average % Tardy (Orders)	Average % Early (Orders)	Average % On Time (Orders)	Average Tardiness	Average Earliness
NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK ┬
ADD/EDD ┬	NLQB/CR	NLQB/CR	ADD/EDD ┬	ACR/CR │
ASLK/SLK ┴	ASLK/SLK ┬	ASLK/SLK ┬	ASLK/SLK │	ADD/EDD ┴
NLQB/CR	ADD/EDD ┴	ADD/EDD ┴	NLQB/CR ┴	NLQB/CR
ACR/CR	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ

NOTE: The heuristics connected with the symbol "I" are not significantly different at a significance level of 0.10.

TABLE 5.3 (Continued)

Expr. Condition 3: AAW_High, DPV_Low, CTB_Order Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ACR/CR	ADD/EDD	ACR/CR	ADD/EDD	ACR/CR
ADD/EDD	ACR/CR	ADD/EDD	ACR/CR	ADD/EDD
ASLK/SLK	ASLK/SLK	ASLK/SLK	ASLK/SLK	ASLK/SLK
NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR
NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ
Average % Tardy (Orders)	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK
ADD/EDD	NLQB/CR	ASLK/SLK	ADD/EDD	ACR/CR
ASLK/SLK	ASLK/SLK	ADD/EDD	ASLK/SLK	ADD/EDD
NLQB/CR	ADD/EDD	NLQB/CR	NLQB/CR	NLQB/CR
ACR/CR	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ

Expr. Condition 4: AAW_High, DPV_Low, CTB_Container Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ACR/CR	ADD/EDD	ACR/CR	ADD/EDD	ADD/EDD
ADD/EDD	ACR/CR	ADD/EDD	ASLK/SLK	ACR/CR
ASLK/SLK	ASLK/SLK	ASLK/SLK	ACR/CR	ASLK/SLK
NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR
NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ
Average % Tardy (Orders)	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK
ADD/EDD	NLQB/CR	NLQB/CR	ADD/EDD	ADD/EDD
ASLK/SLK	ASLK/SLK	ASLK/SLK	ASLK/SLK	ACR/CR
NLQB/CR	ADD/EDD	ADD/EDD	NLQB/CR	NLQB/CR
ACR/CR	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ

NOTE: The heuristics connected with the symbol "I" are not significantly different at a significance level of 0.10.

TABLE 5.3 (Continued)

Expr. Condition 5: AAW_Low, DPV_High, CTB_Order Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ADD/EDD	ADD/EDD	ADD/EDD	ADD/EDD	ADD/EDD
ASLK/SLK ┬	NEQA/EQ	ASLK/SLK ┬	NEQA/EQ	ASLK/SLK ┬
ACR/CR ┴	ASLK/SLK	ACR/CR ┴	ASLK/SLK	ACR/CR ┴
NLQB/CR ┬	NLQB/CR	NEQA/EQ	NLQB/CR	NLQB/CR ┬
NEQA/EQ ┴	ACR/CR	NLQB/CR	ACR/CR	NEQA/EQ ┴
Average % Tardy (Orders)	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
ADD/EDD	NLQB/CR	ACR/CR	ACR/CR	ASLK/SLK
ASLK/SLK	ACR/CR	NLQB/CR	ASLK/SLK ┬	ACR/CR ┬
NEQA/EQ	NEQA/EQ	ASLK/SLK ┬	ADD/EDD │	ADD/EDD ┴
NLQB/CR	ASLK/SLK	NEQA/EQ │	NLQB/CR ┴	NLQB/CR
ACR/CR	ADD/EDD	ADD/EDD ┴	NEQA/EQ	NEQA/EQ

Expr. Condition 6: AAW_Low, DPV_High, CTB_Container Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ACR/CR ┬	ADD/EDD ┬	ACR/CR	ADD/EDD	ADD/EDD ┬
ADD/EDD │	NEQA/EQ │	ADD/EDD ┬	ASLK/SLK	ASLK/SLK ┴
ASLK/SLK ┴	ASLK/SLK ┴	ASLK/SLK ┴	NEQA/EQ	ACR/CR
NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR
NEQA/EQ	ACR/CR	NEQA/EQ	ACR/CR	NEQA/EQ
Average % Tardy (Orders)	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
ADD/EDD ┬	NLQB/CR	ACR/CR	ACR/CR	ACR/CR
ASLK/SLK ┴	ACR/CR	NLQB/CR	NLQB/CR	ASLK/SLK ┬
NEQA/EQ	NEQA/EQ	NEQA/EQ	ASLK/SLK ┬	ADD/EDD ┴
ACR/CR	ASLK/SLK ┬	ASLK/SLK ┬	ADD/EDD ┴	NLQB/CR
NLQB/CR	ADD/EDD ┴	ADD/EDD ┴	NEQA/EQ	NEQA/EQ

NOTE: The heuristics connected with the symbol "I" are not significantly different at a significance level of 0.10.

TABLE 5.3 (Continued)

Expr. Condition 7: AAW_Low, DPV_Low, CTB_Order Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ADD/EDD ┬	ADD/EDD	ACR/CR ┬	ADD/EDD	ADD/EDD ┬
ASLK/SLK │	ASLK/SLK	ADD/EDD ┴┬	ASLK/SLK	ASLK/SLK │
ACR/CR ┴	ACR/CR	ASLK/SLK ┴	ACR/CR	ACR/CR ┴
NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR	NLQB/CR
NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ	NEQA/EQ
Average % Tardy (Orders)	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
ADD/EDD ┬	NLQB/CR ┬	ACR/CR	ACR/CR	ASLK/SLK ┬
ASLK/SLK ┴	NEQA/EQ ┴	NLQB/CR	ADD/EDD ┬	ADD/EDD │
ACR/CR	ACR/CR	NEQA/EQ	NLQB/CR │	ACR/CR ┴
NLQB/CR	ASLK/SLK ┬	ASLK/SLK ┬	ASLK/SLK ┴	NLQB/CR
NEQA/EQ	ADD/EDD ┴	ADD/EDD ┴	NEQA/EQ	NEQA/EQ

Expr. Condition 8: AAW_Low, DPV_Low, CTB_Container Size

Average Time in System	Average Queue Waiting Time	Average Net Present Value	Average Work-in-Process	Average # of Orders in System
ACR/CR ┬	ADD/EDD ┬	ACR/CR	ADD/EDD	ADD/EDD ┬
ADD/EDD │	ASLK/SLK ┴	ADD/EDD ┬	ASLK/SLK	ASLK/SLK ┴
ASLK/SLK ┴	NEQA/EQ	ASLK/SLK │	NEQA/EQ	ACR/CR
NLQB/CR	NLQB/CR	NLQB/CR ┴	ACR/CR	NLQB/CR
NEQA/EQ	ACR/CR	NEQA/EQ	NLQB/CR	NEQA/EQ
Average % Tardy (Orders)	**Average % Early (Orders)**	**Average % On Time (Orders)**	**Average Tardiness**	**Average Earliness**
ADD/EDD ┬	NLQB/CR ┬	ACR/CR	ACR/CR	ACR/CR ┬
ASLK/SLK ┴	NEQA/EQ ┴	NLQB/CR	NLQB/CR	ASLK/SLK ┴
ACR/CR	ACR/CR	NEQA/EQ	ADD/EDD ┬	ADD/EDD
NLQB/CR	ASLK/SLK ┬	ASLK/SLK ┬	ASLK/SLK ┴	NLQB/CR
NEQA/EQ	ADD/EDD ┴	ADD/EDD ┴	NEQA/EQ	NEQA/EQ

NOTE: The heuristics connected with the symbol "I" are not significantly different at a significance level of 0.10.

Average Time in System

ACR/CR and ADD/EDD exhibited excellent performance on this measure. ACR/CR, which was ranked first under seven of the eight shop environmental conditions (some tied with other heuristics), was the best performer on this measure. ACR/CR was ranked second when the average annual workload was low, the demand pattern variability was high, and the cell transfer batch equalled the order size. ADD/EDD, which was ranked first under six shop environmental conditions (some tied with other heuristics), was the second best performer on this measure. ADD/EDD was ranked second when the average annual workload was high and the cell transfer batch equalled the container size. ASLK/SLK, which was ranked first under three shop environmental conditions (all tied with other heuristics), was the third best performer on this measure. NLQB/CR, which was ranked the second worst under most of the shop environmental conditions, was the second worst performer on this measure. NEQA/EQ, which was ranked last under all shop environmental conditions (one tied with NLQB/CR), was the worst performer on this measure.

It can be stated that the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ), which include workcenter status in their queue selection rules. Furthermore, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

Average Queue Waiting Time

ADD/EDD, which was ranked first under all eight shop environmental conditions (some tied with other heuristics), was the best performer on this measure. ASLK/SLK, which was ranked first under four shop environmental conditions (all tied with other heuristics), was the second best performer on this measure. ACR/CR, which was ranked first under two shop environmental conditions (all tied with other heuristics), was the third best performer on this measure. NLQB/CR, which was ranked the second worst under most of the shop environmental conditions, was the second worst performer on this measure. NEQA/EQ, which was ranked last under four shop environmental conditions, was the worst performer on this measure.

Average Net Present Value

ACR/CR, which was ranked first under seven of the eight shop environmental conditions (some tied with other heuristics), was the best performer on this measure. ACR/CR was ranked second when the average annual workload was low, the demand pattern variability was high, and the cell transfer batch equalled the order size. ADD/EDD, which was ranked first under five shop environmental conditions (some tied with other heuristics), was the second best performer on this measure. ASLK/SLK, which was ranked first under two shop environmental conditions (all tied with other heuristics), was the third best performer on this measure. NLQB/CR, which was ranked the second worst under most of the shop environmental conditions, was the second worst performer on this measure. NEQA/EQ, which was ranked last under seven shop environmental conditions (one tied with NLQB/CR), was the worst performer on this measure.

It can be stated that the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ), which include workcenter status in their queue selection rules. Furthermore, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

Average Work-in-Process

ADD/EDD, which was ranked first under all eight shop environmental conditions (some tied with other heuristics), was the best performer on this measure. ASLK/SLK, which was ranked first under three shop environmental conditions (all tied with other heuristics), was the second best performer on this measure. ACR/CR, which was ranked first under one shop environmental condition (tied with other heuristics), was the third best performer on this measure. NLQB/CR, which was ranked the second worst under most of the shop environmental conditions, was the second worst performer on this measure. NEQA/EQ, which was ranked last under three shop environmental conditions, was the worst performer on this measure.

Average Number of Orders in System

ADD/EDD, which was ranked first under all eight shop environmental conditions (some tied with other heuristics), was the best performer on this measure. ACR/CR, which was ranked first under five shop environmental conditions (all tied with other heuristics), was the second best performer on this measure. ASLK/SLK, which was ranked first under five shop environmental conditions (all tied with other heuristics), was the third best performer on this measure. NLQB/CR, which was ranked the second worst under most of the shop environmental conditions, was the second worst performer on this measure. NEQA/EQ, which was ranked last under all shop environmental conditions (one tied with NLQB/CR), was the worst performer on this measure.

It can be stated that the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ), which include workcenter status in their queue selection rules. Furthermore, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

Average Percentage of Orders Tardy

On this measure, NEQA/EQ performed excellently under the shop environmental conditions with the high average annual workload, while ADD/EDD and ASLK/SLK performed best under the shop environmental conditions with the low average annual workload. NEQA/EQ, ADD/EDD, and ASLK/SLK were ranked first under four, four, and three shop environmental conditions (some tied with other heuristics), respectively. ACR/CR was consistently the worst performer under the shop environmental conditions with the high average annual workload. NLQB/CR was the second worst performers under most of the shop environmental conditions.

It can be stated that the performance differences among the five heuristics were heavily influenced by the levels of the average annual workload with respect to this measure. NEQA/EQ was the best performing heuristic and ACR/CR was the worst performing heuristic when the average annual workload was set to high level. ADD/EDD was the best performing heuristic when the average annual workload was set to low level.

Average Percentage of Orders Early

On this measure, ACR/CR performed best under the shop environmental conditions with the high average annual workload, while NLQB/CR and NEQA/EQ performed best under the shop environmental conditions with the low average annual workload. ACR/CR and NEQA/EQ were ranked first both under four shop environmental conditions (some tied with other heuristics), while NEQA/EQ was ranked first under two shop environmental conditions (all tied with other heuristics). NEQA/EQ and ADD/EDD were consistently the worst performers under the shop environmental conditions with the high and low average annual workload, respectively. ADD/EDD and ASLK/SLK were consistently the second worst performers under the shop environmental with the high and low average annual workload, respectively.

It can be concluded that the performance differences among the five heuristics were heavily influenced by the levels of the average annual workload with respect to this measure. ACR/CR was the best performing heuristic and NEQA/EQ was the worst performing heuristic when the average annual workload was set to high level. NLQB/CR was the best performing heuristic and ADD/EDD was the worst performing heuristic when the average annual workload was set to low level.

Average Percentage of Orders On Time

ACR/CR, which was ranked first under all eight shop environmental conditions, was the best performer on this measure. In general, NLQB/CR was the second best performer on this measure. NEQA/EQ was the worst performer and ADD/EDD tied NLQB/CR for the second worst performers when the average annual workload was set to high level. ADD/EDD tied ASLK/SLK for the worst performers and NEQA/EQ was the second worst performer when the average annual workload was set to low level.

Average Tardiness

When the average annual workload was set to high, the ranking of the five heuristics did not change under any of the four shop environmental conditions and the ranking from best to worst was ACR/CR, ADD/EDD, ASLK/SLK, NLQB/CR, and NEQA/EQ. In general, ACR/CR was the best performer and NEQA/EQ was the worst performer when the average annual workload was set to low level.

It can be stated that when the average annual workload was set to high, the three heuristics which considered jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ). Furthermore, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

Average Earliness

ASLK/SLK, which was ranked first under seven of the eight shop environmental conditions (some tied with other heuristics), was the best performer on this measure. ASLK/SLK was ranked second when the average annual workload was low, the demand pattern variability was high, and the cell transfer batch equalled the container size. ACR/CR tied ADD/EDD for the second best performers on this measure. ACR/CR and ADD/EDD were ranked first both under four shop environmental conditions (some tied with other heuristics). NLQB/CR, which was ranked the second worst under all shop environmental conditions, was the second worst performer on this measure. NEQA/EQ, which was ranked last under all shop environmental conditions, was the worst performer on this measure.

It can be stated that the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ), which include workcenter status in their queue selection rules. Furthermore, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

Conclusions Drawn from Paired t-test Analysis

In Table 5.4, ranking comparisons for the five heuristics based on the results of the paired t-test analysis are presented. The number in each grid denotes the total number of a specific rank for a heuristic under all eight shop environmental conditions with respect to all ten performance measures. For example, the number of 47 of the rank 1 for ACR/CR in Table 3.1 denotes that totally ACR/CR was ranked first for 47 times (out of 80).

TABLE 5.4

RANKING COMPARISONS FOR THE FIVE HEURISTICS

Group Scheduling Heuristic	Rank				
	1 (best)	2	3	4	5 (worst)
ACR/CR	47	18	7	4	4
ADD/EDD	43	23	9	4	1
ASLK/SLK	31	31	13	5	0
NLQB/CR	4	39	27	9	1
NEQA/EQ	7	8	36	26	3

A conclusion drawn from the results of the paired t-test analysis is that no universal heuristic existed in this study. This is because the prioritizing mechanisms applied to the five heuristics were different. No single heuristic always outperformed the other four heuristics under all eight shop environmental condition with respect to all ten performance measures.

Another conclusion drawn from the results of the paired t-test analysis is that the three heuristics (ADD/EDD, ACR/CR, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ) on the following measures: the average time in system, average net present value, average number of orders in system, average earliness, and average tardiness with the high average annual workload. This is because the first three heuristics (ADD/EDD, ACR/CR, and ASLK/SLK) consider jobs' due dates in both their queue selection rules and job dispatching rules, while the other two heuristics include workcenter status in their queue selection rules. Moreover, NLQB/CR consistently outperformed NEQA/EQ on these measures. This because NLQB/CR considers jobs' due dates in its job dispatching rule and NEQA/EQ does not consider jobs' due dates.

As can be seen from Tables 5.3 and 5.4, overall, ACR/CR and ADD/EDD were the best and the second best performing heuristics, respectively. ACR/CR performed best on the measures of the average time in system, average net present value, average percentage of orders on time, average tardiness, and average percentage of orders early with the high average annual workload. ADD/EDD performed best on the measures of the average queue waiting time, average work-in-process, average number of orders in system, and average percentage of orders tardy with the low average annual workload. Overall, ASLK/SLK was the third best performing heuristic which performed best on the measure of the average earliness.

The prioritizing mechanisms applied to ACR/CR and ASLK/SLK are very similar and both focus on hitting jobs' due dates to avoid earliness and tardiness, while the prioritizing mechanism applied to ADD/EDD focuses on finishing processing of jobs before their due dates to avoid tardiness. The results of the paired t-test analysis did conform to the prioritizing mechanisms applied to these three heuristics. For example, ADD/EDD consistently outperformed ACR/CR and ASLK/SLK on the measure of the average percentage of orders tardy, while ACR/CR and ASLK/SLK consistently outperformed ADD/EDD on the measures of the average percentage of orders on time and average percentage of orders early.

Overall, ACR/CR and ADD/EDD performed better than ASLK/SLK. This is consistent with the results of a recent study done by Rohleder and Scudder (1992). They examined several job dispatching rules in a job shop with forbidden early shipment. The results of their study showed that CR (critical ratio) and EDD (earliest due date) outperformed SLK (slack) on the measures of the average net present value, average work-in-process, average number of orders in system, and average tardiness when the average percent tardy for all rules tested was about 77%.

NEQA/EQ and NLQB/CR were the worst and the second worst performing heuristics on most of the measures, respectively. Exceptionally, NEQA/EQ exhibited excellent performance on the measure of the average percentage of orders tardy when the average annual workload was set to high level. This is because the prioritizing mechanism applied to NEQA/EQ focuses on reducing shop congestion by processing jobs which are expected to go into empty queues for the next operation. Since NEQA/EQ does not contain the information of jobs' due dates, it performed worst on the measure of the average tardiness. Another exception is that NLQB/CR was the best and the second best performers on the measures of the average percentage of orders early and average percentage of orders on time, respectively, when the average annual workload was set to low level. This is because the prioritizing mechanism applied to NEQA/EQ focuses on reducing shop congestion by considering the workcenter status behind and job information (i.e., critical ratio).

Analysis of Variance

The analysis of variance (ANOVA) allows the examination of both the individual effect of each of the experimental factors on the performance of the cellular system as well as the degree to which the experimental factors interact. A linear statistical model for analysis of performance measure Y can be given by :

$$Y_{ijkmn} = \mu + G_i + A_j + D_k + C_m + GA_{ij} + GD_{ik} + GC_{im} + AD_{jk} + AC_{jm} + DC_{km} + GAD_{ijk} + GAC_{ijm} + GDC_{ikm} + ADC_{jkm} + GADC_{ijkm} + \varepsilon_{ijkmn}$$

Where:

Y_{ijkmn} = Mean value of performance measure Y with i-th heuristic, j-th level of average annual workload, k-th level of demand pattern variability, m-th level of cell transfer batch, and n-th replication
μ = Common effect (or global mean)
G_i = Main effect of the i-th group scheduling heuristic
A_j = Main effect of the j-th level of average annual workload
D_k = Main effect of the k-th level of demand pattern variability
C_m = Main effect of the m-th level of cell transfer batch
GA_{ij}, ..., DC_{km} = Interaction effects between any two of main effects
GAD_{ijk}, ..., ADC_{ijm} = Interaction effects among any three of main effects
$GADC_{ijkm}$ = Interaction effect among all main effects
ε_{ijkmn} = Random effect

If E_{ij} denotes the j-th effect (either main effect or interaction effect) with respect to the i-th performance measure, the following hypothesis can be tested:

H_o: E_{ij} has no effect on the outcome of experiments
H_a: Not H_o

A summary of results of the analysis of variance is presented in Table 5.5. The SAS (Statistical Analysis System) program and sample SAS outputs of the analysis of variance (ANOVA) are shown in Appendix B. The general conclusion drawn from the results of the analysis of variance is that all four experimental factors (i.e., group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch) exhibited statistical significance with respect to all performance measures at a significance level of 0.10.

The results of the analysis of variance show that the second order interactions between any two experimental factors were statistically significant on most of the measures. While the third order and the fourth order interactions among experimental factors were not statistically significant on most of the measures, the interaction of A*D*C was significant on all measures. The significance of A*D*C means that if we change the levels of A*D (or A*C, D*C) the performance differences between the levels of C (or D, A) are significantly different.

TABLE 5.5

SUMMARY OF RESULTS OF ANALYSIS OF VARIANCE

		Average Time in System		Average Queue Waiting Time		Average Net Present Value	
		$R^2=0.7529$ Y Mean=7115.99		$R^2=0.7757$ Y Mean=4229.85		$R^2=0.9966$ Y Mean=1878.12	
Source	DF	F	p-value	F	p-value	F	p-value
Model	39	75.01	0.0001	85.15	0.0001	7204.01	0.0001
G	4	47.19	0.0001	3.92	0.0036	20.78	0.0001
A	1	1712.17	0.0001	2870.80	0.0001	736.95	0.0001
D	1	833.15	0.0001	244.71	0.0001	9999.99	0.0001
C	1	5.36	0.0209	74.65	0.0001	31.48	0.0001
G*A	4	22.24	0.0001	3.73	0.0051	10.74	0.0001
G*D	4	4.04	0.0030	2.32	0.0555	0.23	0.9242*
G*C	4	2.18	0.0694	0.20	0.9407*	2.99	0.0181
A*D	1	0.00	0.9867*	1.96	0.1613*	81.65	0.0001
A*C	1	8.07	0.0046	31.23	0.0001	14.15	0.0002
D*C	1	22.13	0.0001	29.68	0.0001	47.70	0.0001
G*A*D	4	3.90	0.0038	1.02	0.3983*	0.13	0.9728*
G*A*C	4	1.02	0.3983*	0.17	0.9561*	1.28	0.2774*
G*D*C	4	0.94	0.4404*	0.57	0.6833*	1.08	0.3669*
A*D*C	1	17.52	0.0001	19.30	0.0001	21.59	0.0001
G*A*D*C	4	0.22	0.9266*	0.17	0.9517*	0.28	0.8938*

NOTE:
(1) The effects with the symbol * do not exhibit statistical significance at a significance level of 0.10.
(2) G, A, D, C stand for group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch, respectively.

TABLE 5.5 (Continued)

		Average Work-in-Process		Average # of Orders in System		Average Percentage of Orders Tardy	
		$R^2=0.7938$ Y Mean=34.95		$R^2=0.8376$ Y Mean=51.07		$R^2=0.8172$ Y Mean=27.59	
Source	DF	F	p-value	F	p-value	F	p-value
Model	39	94.75	0.0001	126.92	0.0001	110.07	0.0001
G	4	8.37	0.0001	44.51	0.0001	39.61	0.0001
A	1	2589.79	0.0001	1392.13	0.0001	3792.93	0.0001
D	1	678.55	0.0001	3008.23	0.0001	75.30	0.0001
C	1	9.49	0.0021	4.56	0.0330	33.58	0.0001
G*A	4	4.38	0.0016	23.19	0.0001	41.34	0.0001
G*D	4	6.11	0.0001	14.25	0.0001	3.22	0.0123
G*C	4	0.61	0.6577*	0.86	0.4884*	1.88	0.1111*
A*D	1	301.06	0.0001	149.71	0.0001	3.35	0.0674
A*C	1	5.39	0.0205	2.21	0.1376*	0.15	0.7004*
D*C	1	11.51	0.0007	10.56	0.0012	14.32	0.0002
G*A*D	4	3.00	0.0177	9.91	0.0001	0.36	0.8401*
G*A*C	4	0.04	0.9973*	0.58	0.6802*	1.26	0.2847*
G*D*C	4	0.09	0.9864*	0.30	0.8767*	1.09	0.3610*
A*D*C	1	9.04	0.0027	8.10	0.0045	12.93	0.0003
G*A*D*C	4	0.02	0.9991*	0.04	0.9971*	1.24	0.2937*

NOTE:
(1) The effects with the symbol * do not exhibit statistical significance at a significance level of 0.10.
(2) G, A, D, C stand for group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch, respectively.

TABLE 5.5 (Continued)

		Average Percentage of Orders Early		Average Percentage of Orders On Time		Average Tardiness	
		R^2=0.8442 Y Mean=67.82		R^2=0.7962 Y Mean=4.59		R^2=0.7126 Y Mean=4445.77	
Source	DF	F	p-value	F	p-value	F	p-value
Model	39	133.41	0.0001	96.15	0.0001	61.05	0.0001
G	4	75.92	0.0001	378.00	0.0001	264.30	0.0001
A	1	4521.93	0.0001	527.01	0.0001	452.64	0.0001
D	1	28.40	0.0001	1061.28	0.0001	66.01	0.0001
C	1	52.51	0.0001	101.82	0.0001	73.22	0.0001
G*A	4	54.65	0.0001	45.66	0.0001	116.99	0.0001
G*D	4	3.18	0.0130	1.20	0.3110*	2.80	0.0251
G*C	4	3.92	0.0037	45.50	0.0001	25.71	0.0001
A*D	1	4.76	0.0294	5.14	0.0236	4.19	0.0410
A*C	1	1.10	0.2943*	31.11	0.0001	7.59	0.0060
D*C	1	12.01	0.0006	18.92	0.0001	22.06	0.0001
G*A*D	4	0.41	0.7983*	7.30	0.0001	8.43	0.0001
G*A*C	4	1.83	0.1212*	5.73	0.0001	10.34	0.0001
G*D*C	4	1.90	0.1077*	8.61	0.0001	6.78	0.0001
A*D*C	1	9.56	0.0020	34.31	0.0001	8.90	0.0029
G*A*D*C	4	1.37	0.2424*	0.62	0.6466*	1.21	0.3028*

NOTE:
(1) The effects with the symbol * do not exhibit statistical significance at a significance level of 0.10.
(2) G, A, D, C stand for group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch, respectively.

TABLE 5.5 (Continued)

Source	DF	Average Earliness R^2=0.9406 Y Mean=3119.70 F	p-value
Model	39	389.61	0.0001
G	4	1065.51	0.0001
A	1	6407.90	0.0001
D	1	392.38	0.0001
C	1	2337.00	0.0001
G*A	4	185.11	0.0001
G*D	4	6.13	0.0001
G*C	4	76.64	0.0001
A*D	1	30.30	0.0001
A*C	1	21.68	0.0001
D*C	1	477.70	0.0001
G*A*D	4	11.76	0.0001
G*A*C	4	18.07	0.0001
G*D*C	4	14.88	0.0001
A*D*C	1	11.06	0.0009
G*A*D*C	4	1.08	0.3648*

NOTE:

(1) The effects with the symbol * do not exhibit statistical significance at a significance level of 0.10.

(2) G, A, D, C stand for group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch, respectively.

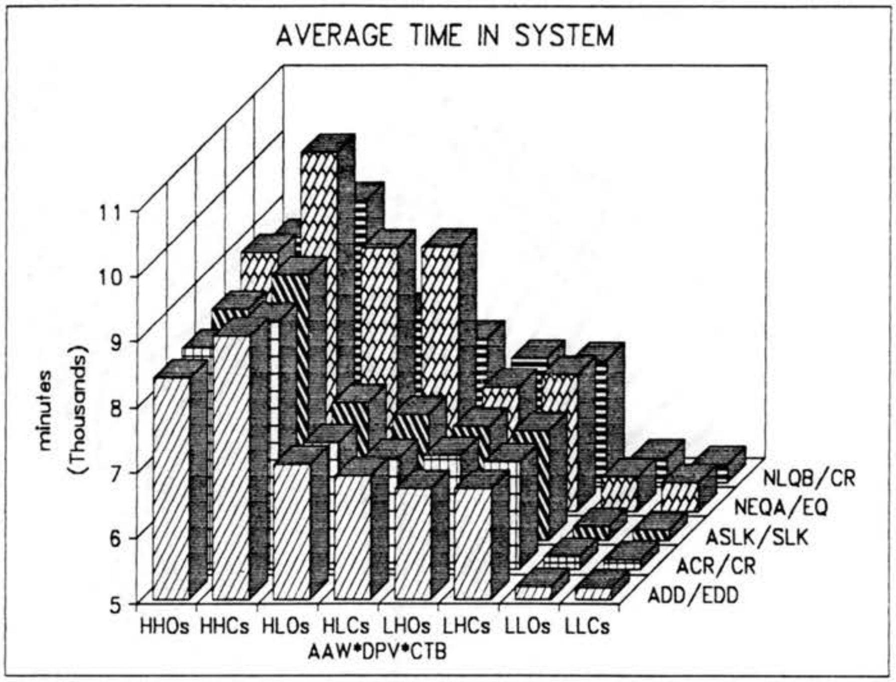
The effects of the three shop environmental factors on the performance of the heuristics are discussed in detail below, followed by the discussion of the second order interactions between heuristic factor and any shop environmental factor. Then, the third order interactions among heuristic factor and any two shop environmental factors are discussed. Note that for the second order and third order interactions, only the performance measures that exhibited statistical significance in Table 5.5 are presented graphically and discussed in detail below.

Shop Environmental Factors

In order to examine the effects of the three shop environmental factors (i.e., average annual workload, demand pattern variability, and cell transfer batch) on the performance of the five heuristics with respect to each of the ten performance measures, graphical representations of the experimental results (Table 5.2) are presented in Figure 5.1. A main effect which exhibited statistical significance means the levels chosen are far enough apart that a discernible difference in performance can be assured.

Average Annual Workload

In Table 5.5, this main effect (i.e., average annual workload) exhibited statistical significance (p-values ≤ 0.0001) with respect



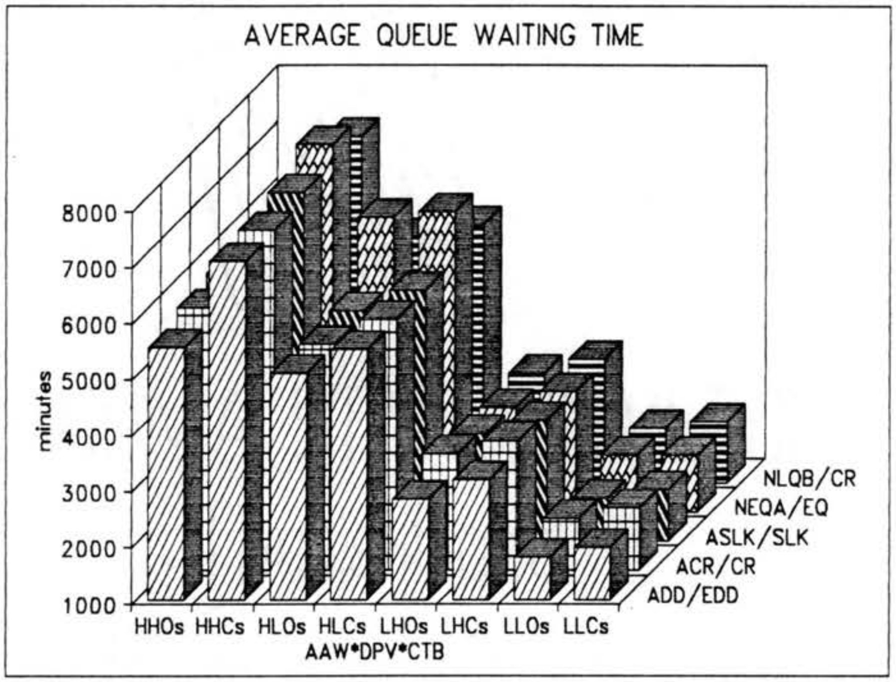
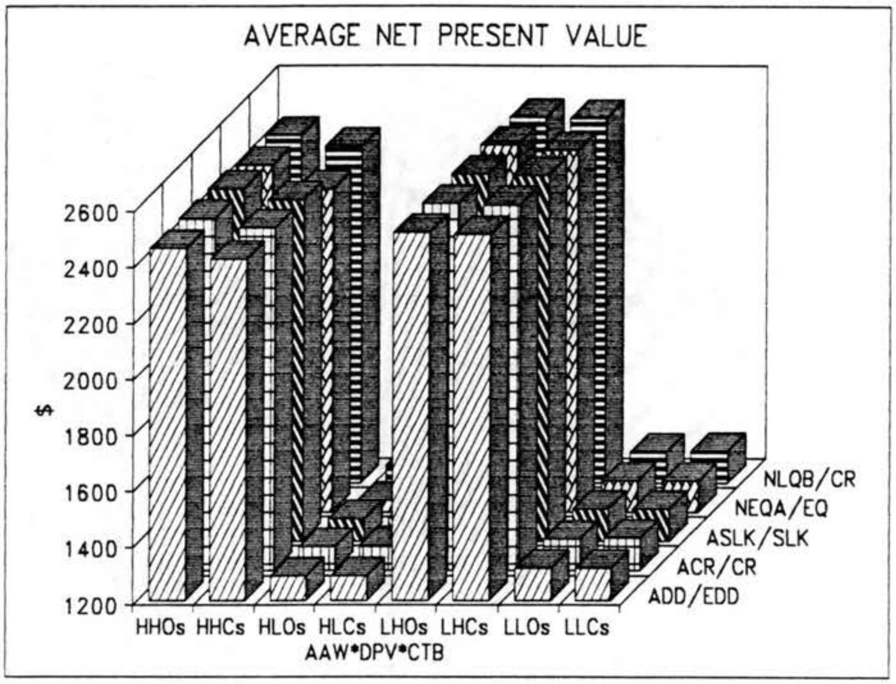


Figure 5.1 Graphical Representations of Experimental Results



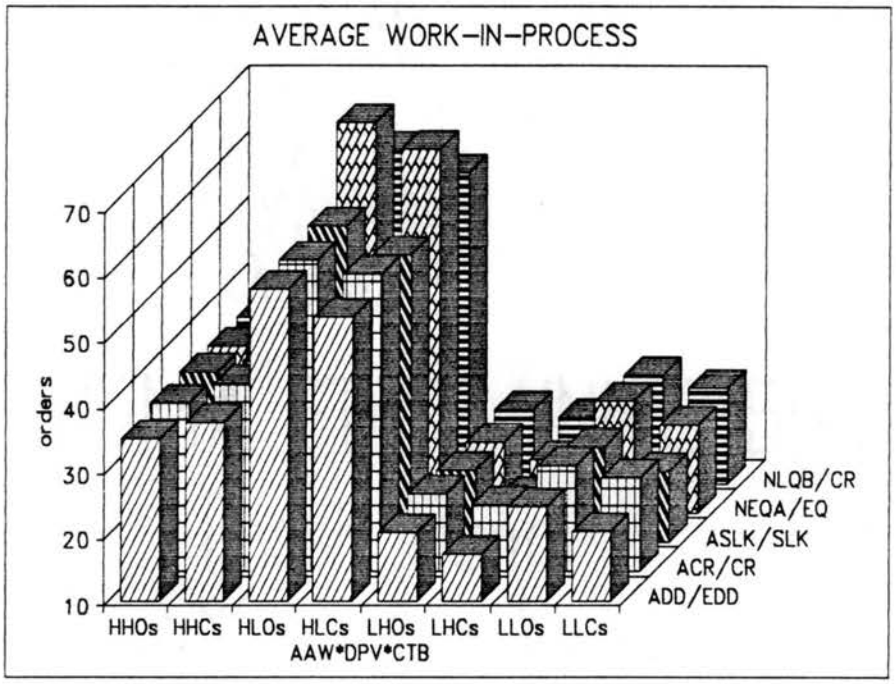
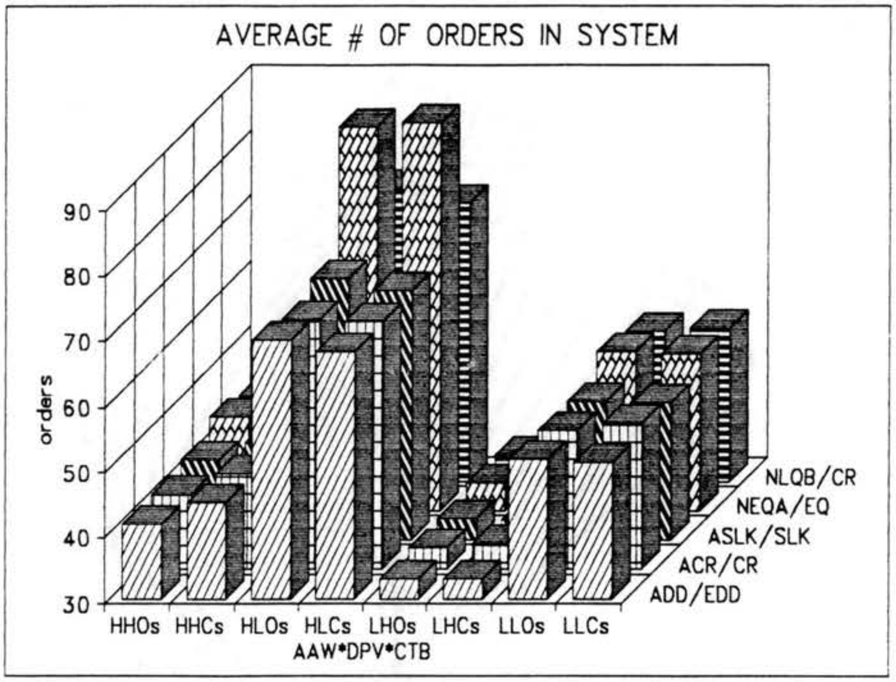


Figure 5.1 (Continued)



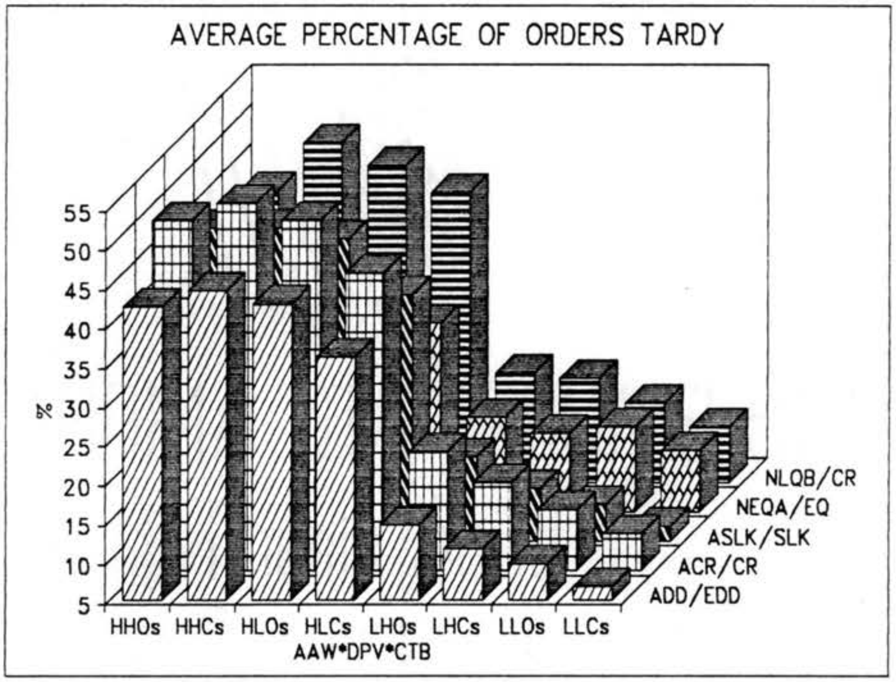
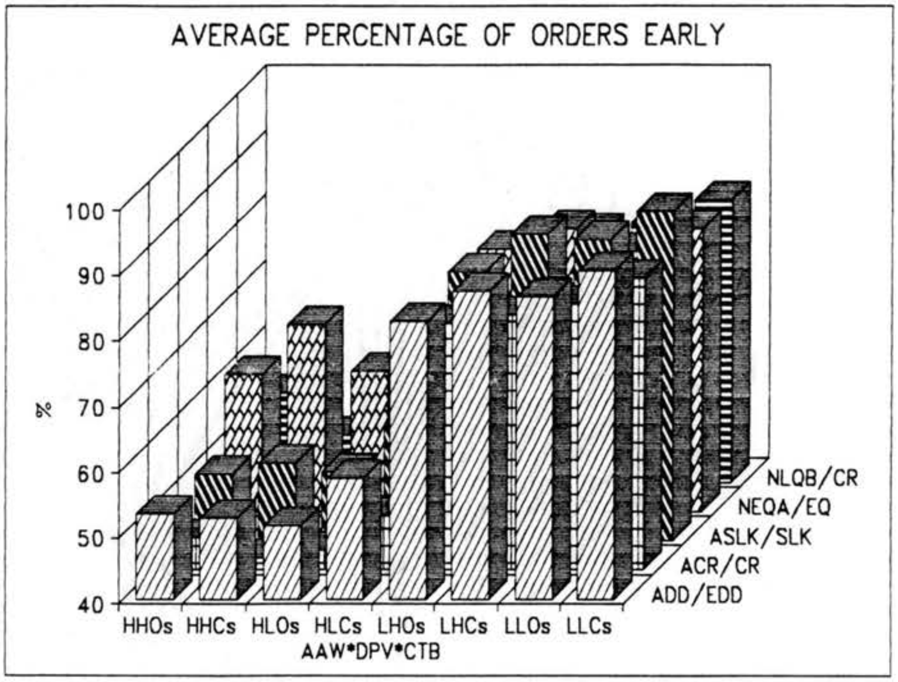


Figure 5.1 (Continued)



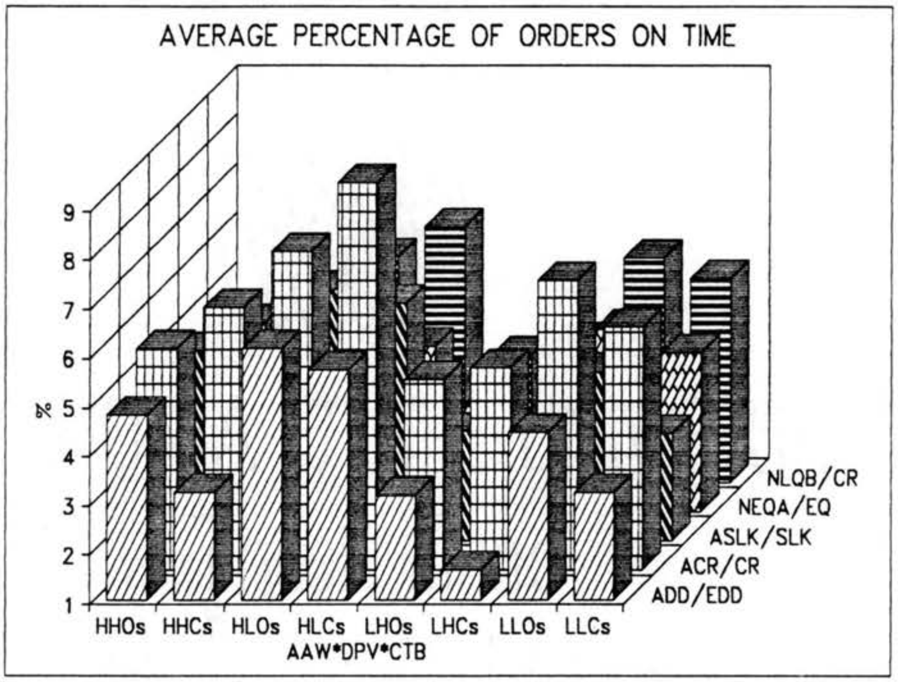
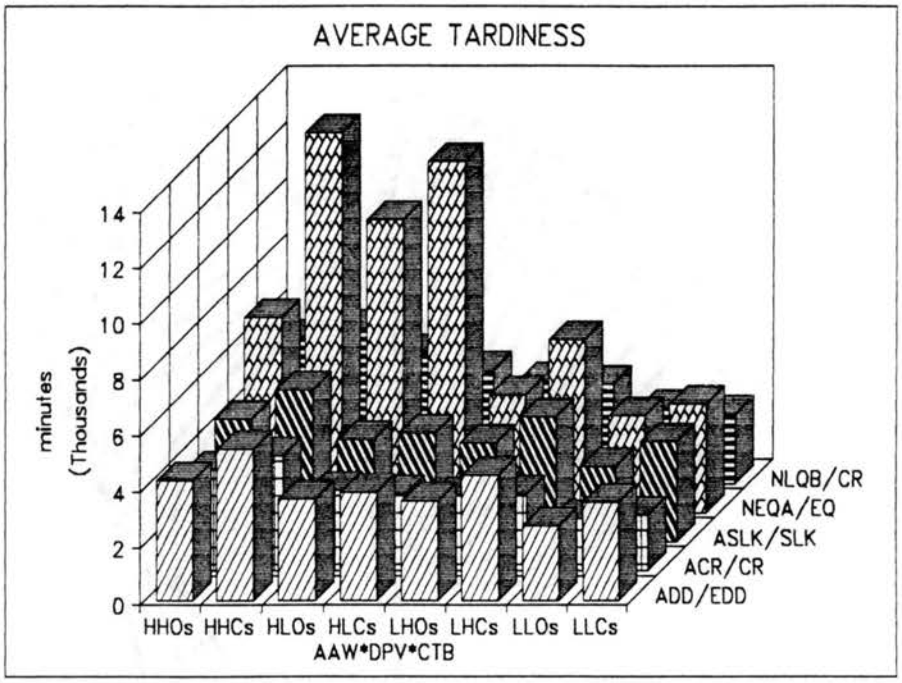


Figure 5.1 (Continued)



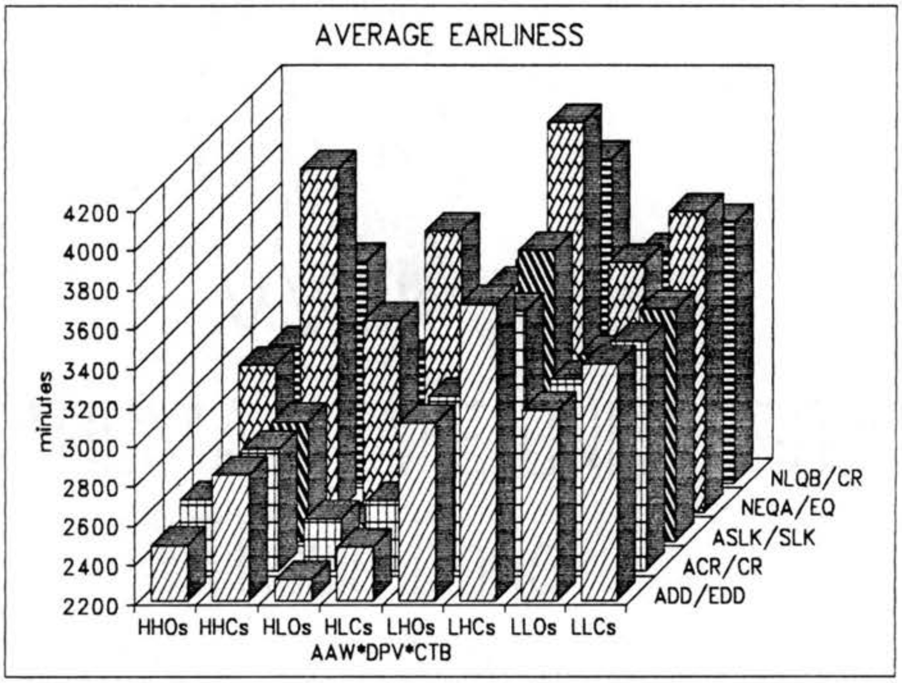


Figure 5.1 (Continued)

to all performance measures at a significance level of 0.10. As can be seen from Figure 5.1, the heuristics with the low average annual workload (i.e., 80% of total system capacity) outperformed those with the high average annual workload (i.e., 90% of total system capacity) on the following measures: the average time in system, average queue waiting time, average net present value, average work-in-process, average number of orders in system, average percentage of orders tardy, and average tardiness.

<u>Demand Pattern Variability</u> In Table 5.5, this main effect (i.e., demand pattern variability) exhibited statistical significance (p-values ≤ 0.0001) with respect to all performance measures at a significance level of 0.10. As can be seen from Figure 5.1, the heuristics with the low demand pattern variability (i.e., frequent arrivals and small order sizes) outperformed those with the high demand pattern variability on the following measures: the average time in system, average queue waiting time, average percentage of orders tardy, average percentage of orders on time, average tardiness, and average earliness.

The worse performance of the heuristics with the low demand pattern variability on the average net present value is because the average order size at the low level of the demand pattern variability (i.e., 17 parts/order) was only a half of that at the high level of the demand pattern variability. The worse performance of the heuristics with the low demand pattern variability on the average work-in-process and average number of orders in system is because the average annual demand at the low level of demand pattern variability (i.e., 40 orders/year) was twice of that at the high level of the demand pattern variability.

Cell Transfer Batch In Table 5.5, this main effect (i.e., cell transfer batch) exhibited statistical significance (p-values ≤ 0.04) with respect to all performance measures at a significance level of 0.10. As can be seen from Figure 5.1, the heuristics with the cell transfer batch which equalled the container size (i.e., 10 parts/container), in general, outperformed those with the cell transfer batch which equalled the order size on the following measures: the average time in system, average net present value, average work-in-process, average number of orders in system, average percentage of orders tardy, and average percentage of orders on time.

One exception is that the heuristics with the cell transfer batch which equalled the order size outperformed those with the cell transfer batch which equalled the container size on most of the measures when the average annual workload and demand pattern variability were both set to high level. This is because using standard containers would increase shop congestion and therefore worsened most of the measures when the average annual workload and demand pattern variability were both set to high level.

Second Order Interactions between Heuristic Factor and Any Shop Environmental Factor

Interaction between Group Scheduling Heuristic and Average Annual Workload (G*A) Graphic representations of this interaction with respect to all performance measures are shown in Figure 5.2. As can be seen, the change patterns of the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) were similar when the level of average annual workload was changed on all measures. This interaction exhibited statistical significance on all measures since NEQA/EQ and NLQB/CR (especially NEQA/EQ) had different change patterns compared with the other three heuristics when the level of average annual workload was changed. Among the graphs in Figure 5.2, the graph of the average queue waiting time was similar to that of the average work-in-process. In addition, the graph of the average time in system was closely similar to that of the average number of orders in system and inversely matched that of the average net present value.

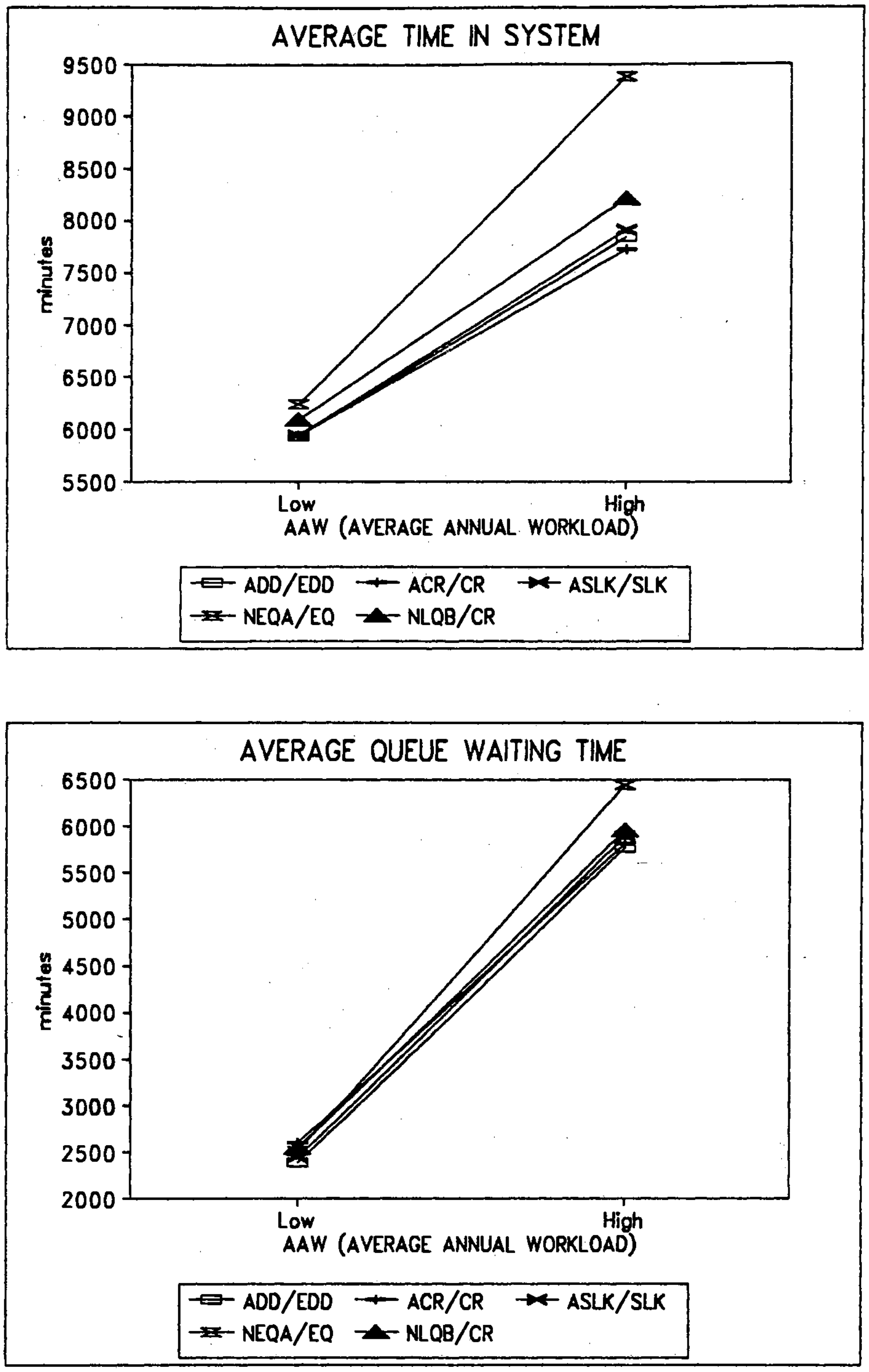
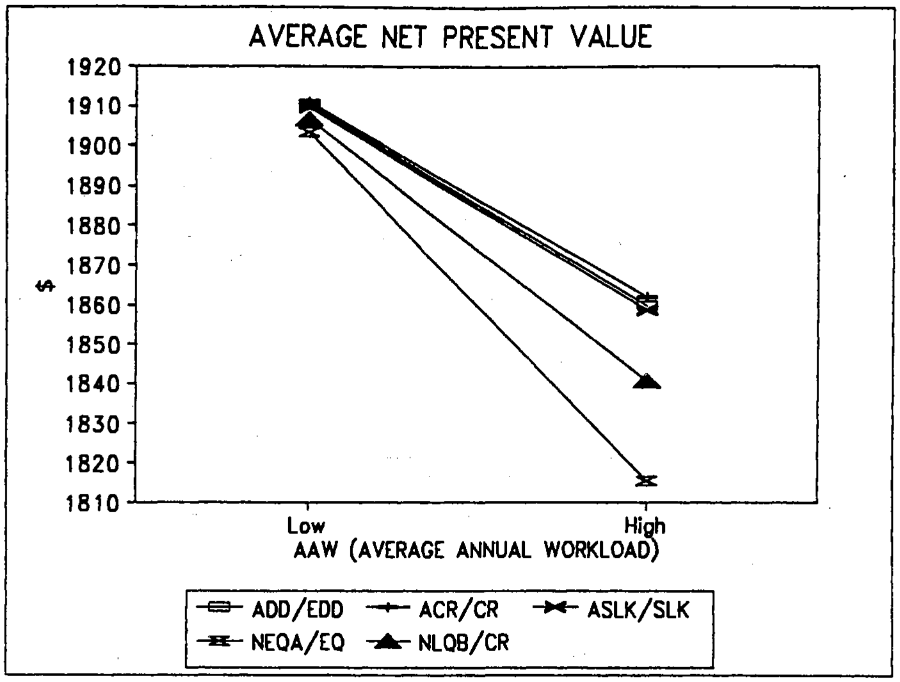


Figure 5.2 Interaction between GSH and AAW (G*A)



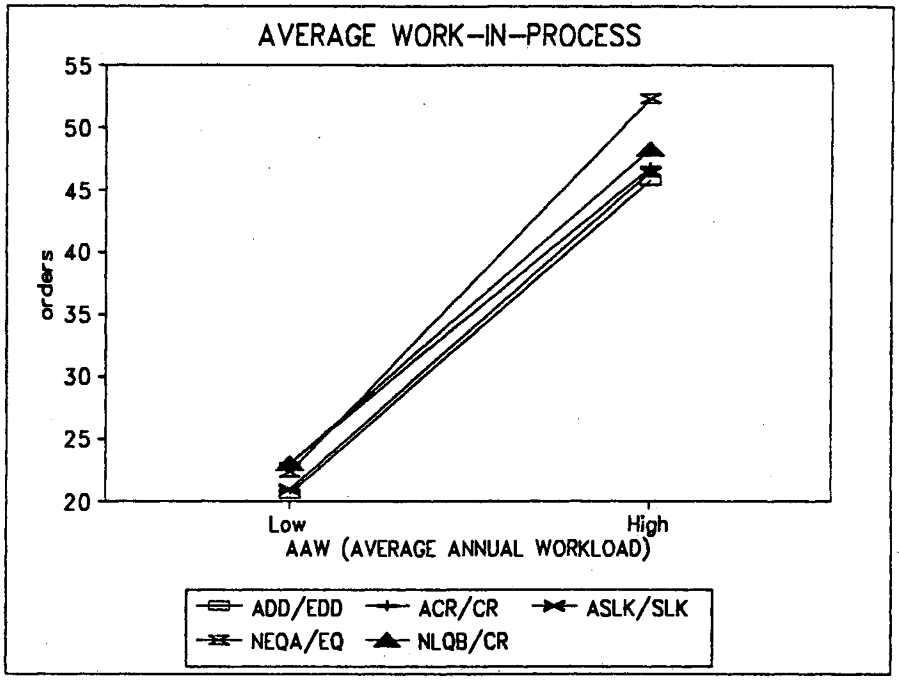


Figure 5.2 (Continued)

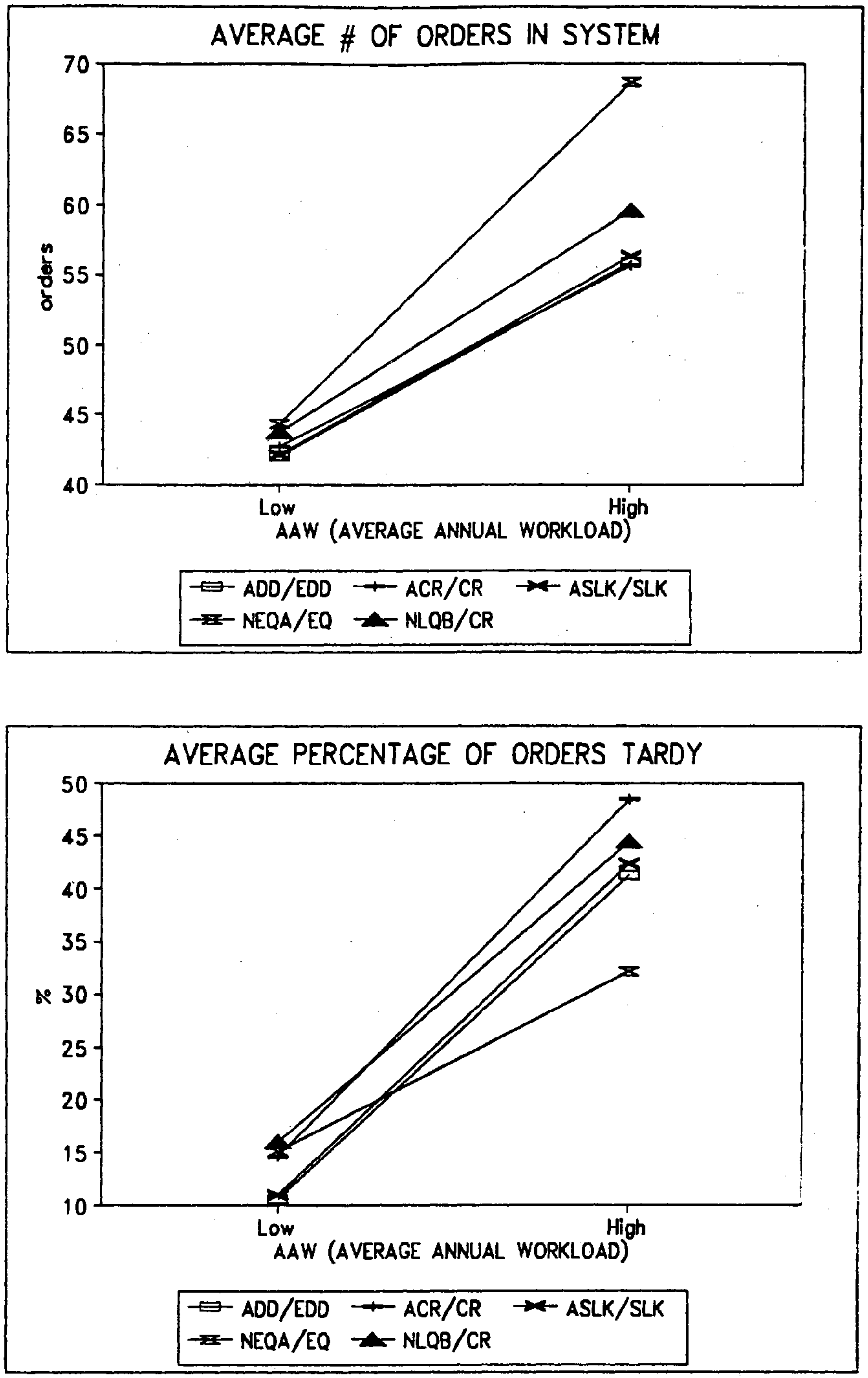
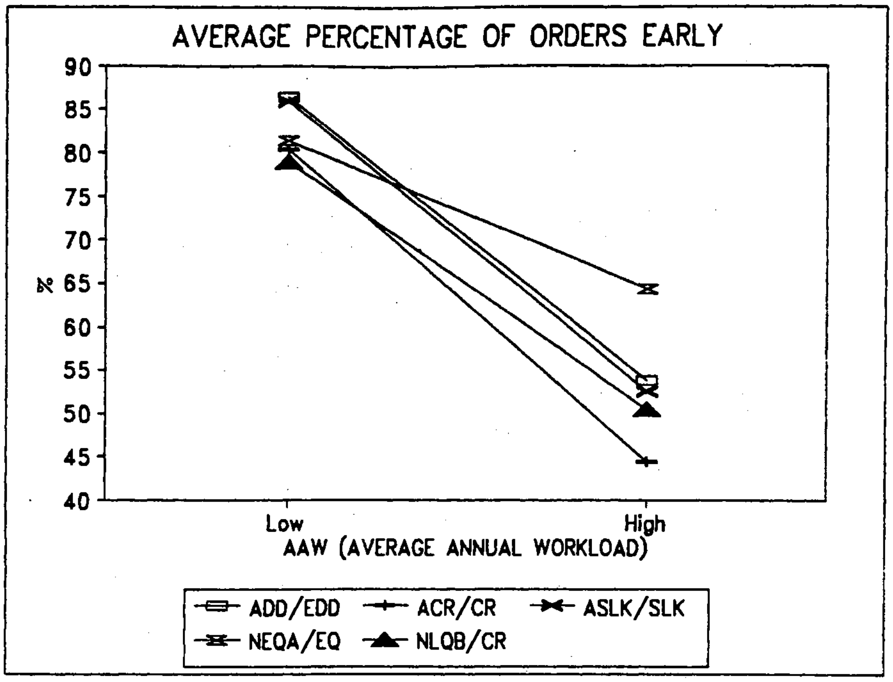


Figure 5.2 (Continued)



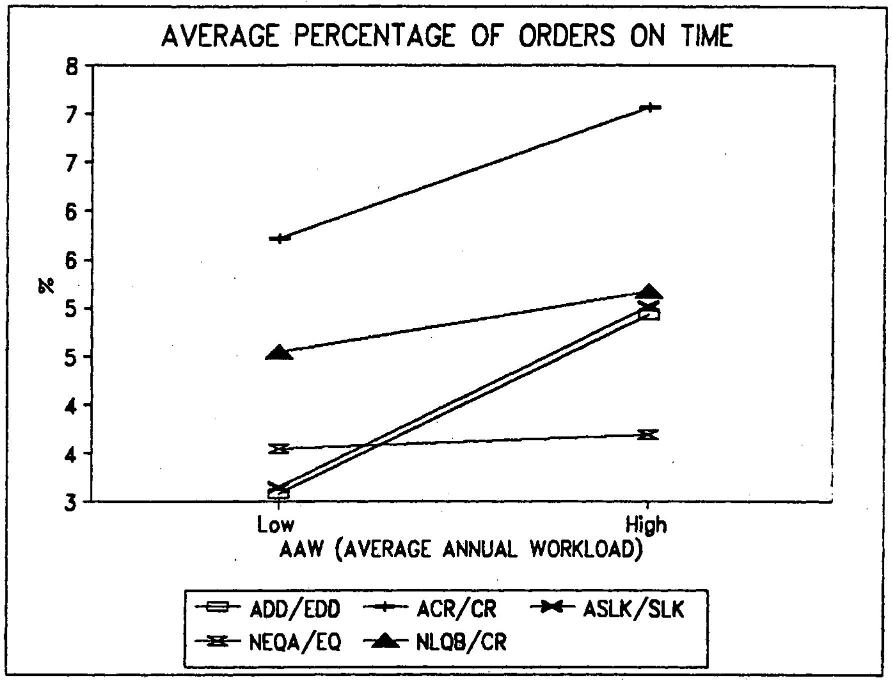
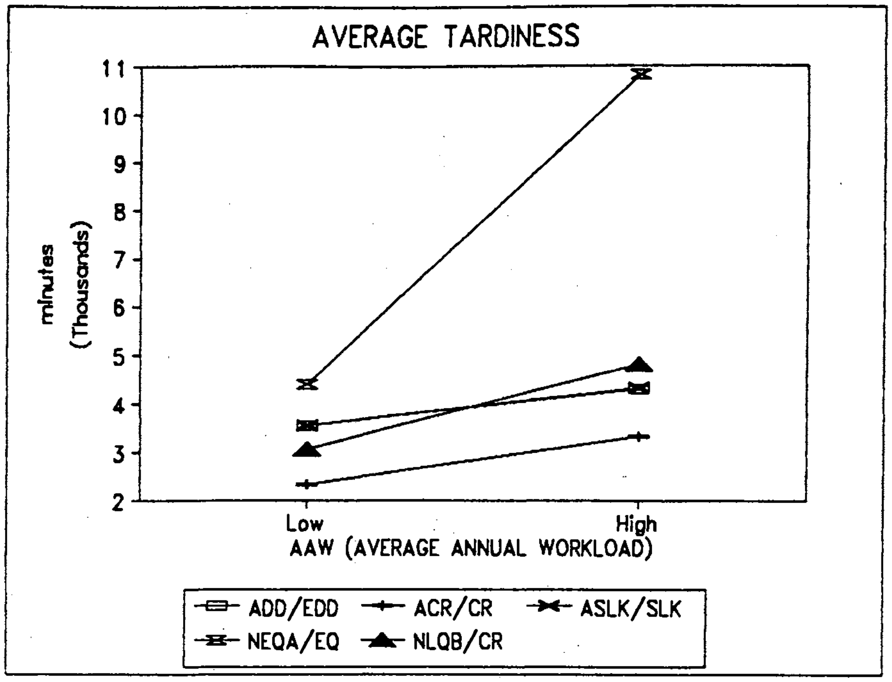


Figure 5.2 (Continued)



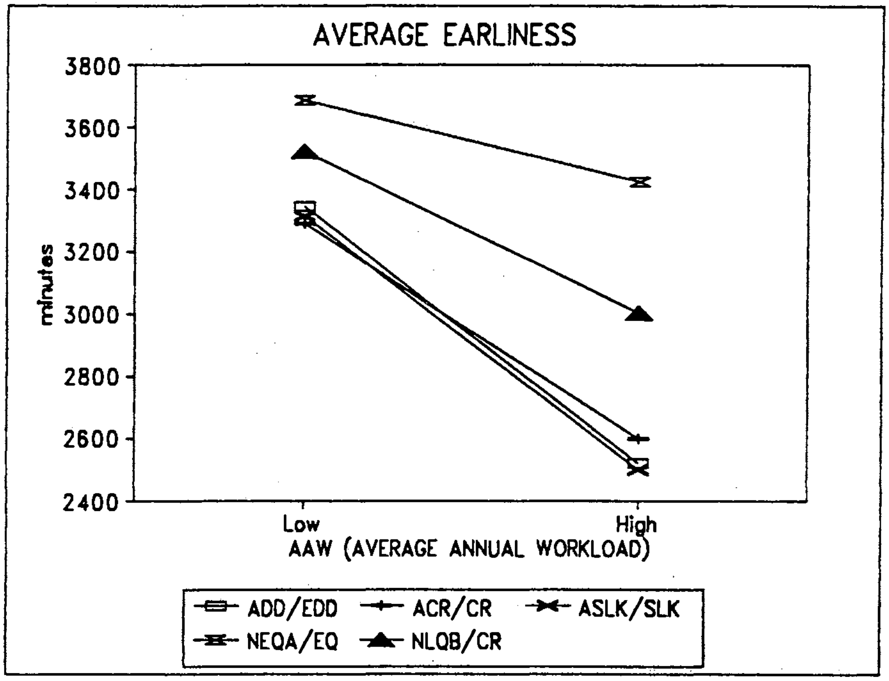
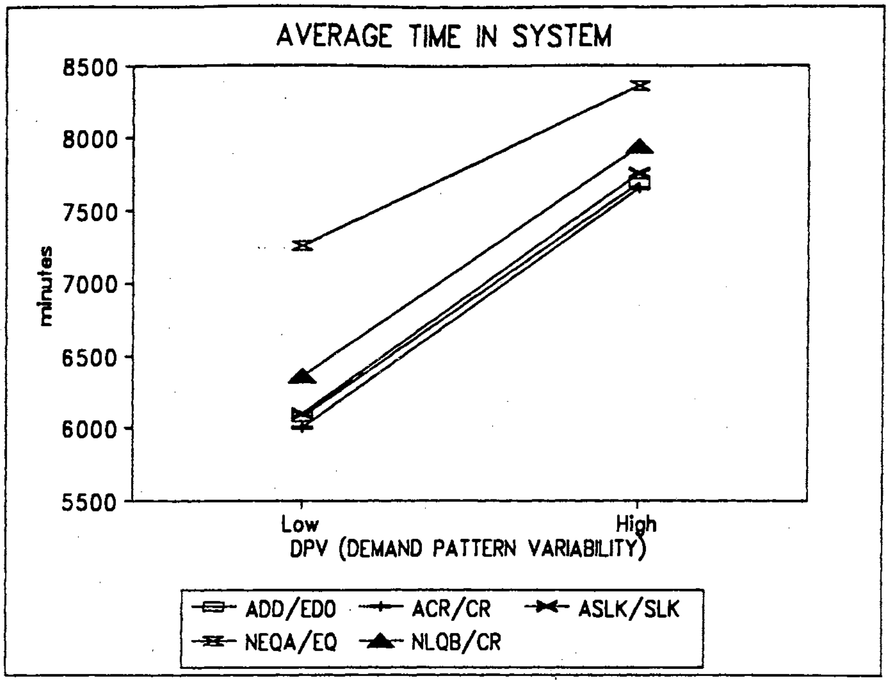


Figure 5.2 (Continued)

Interaction between Group Scheduling Heuristic and Demand Pattern Variability (G*D) Graphic representations of this interaction with respect to various performance measures are shown in Figure 5.3. As can be seen, the change patterns of the three heuristics (ADD/EDD, ACR/CR, and ASLK/SLK) were similar when the level of demand pattern variability was changed on the performance measures included in Figure 5.3. This interaction exhibited statistical significance since NEQA/EQ and NLQB/CR (especially NEQA/EQ) had different change patterns compared with the other three heuristics when the level of demand pattern variability was changed. The graphs of the average queue waiting time and the average work-in-process were similar in shape but opposite in slopes. In addition, the graphs of the average time in system and the average number of orders in system were similar in shape but opposite in slopes.

Interaction between Group Scheduling Heuristic and Cell Transfer Batch (G*C) Graphic representations of this interaction with respect to various performance measures are shown in Figure 5.4. For the graphs in Figure 5.4, the change patterns of the three heuristics (ADD/EDD, ACR/CR and ASLK/SLK) were similar when the level of cell transfer batch was changed. This interaction exhibited statistical



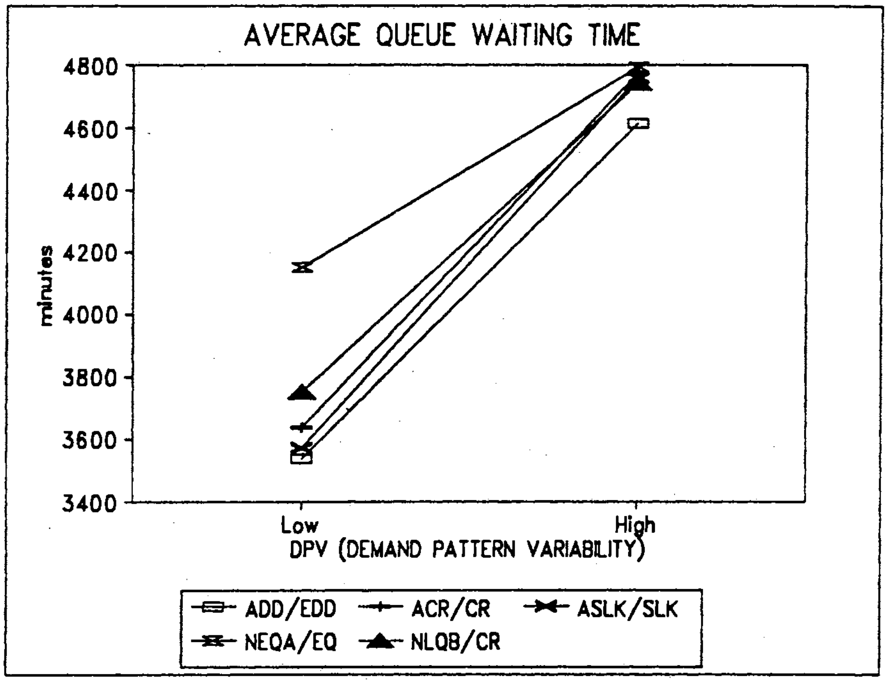
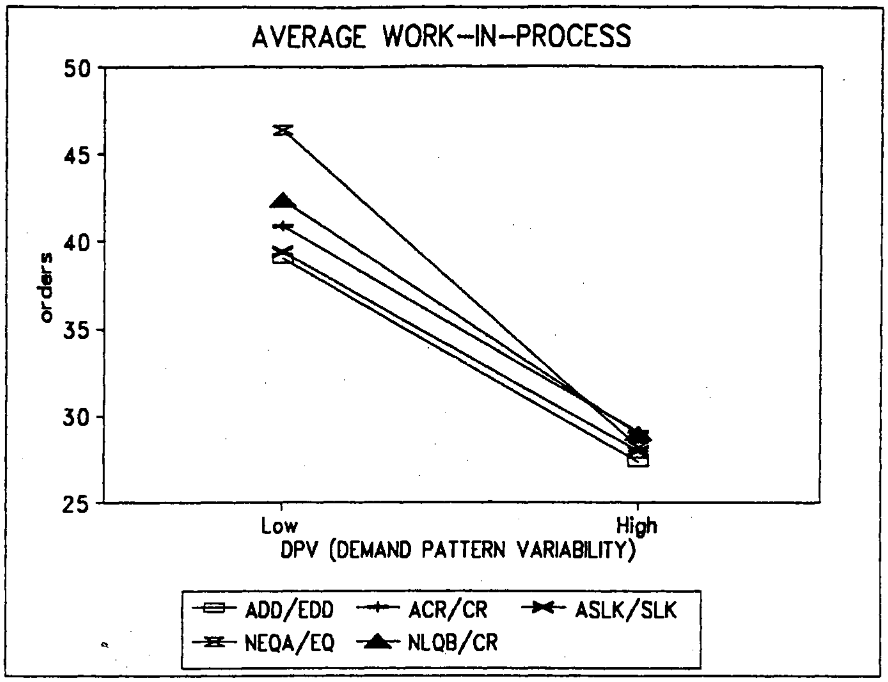


Figure 5.3 Interaction between GSH and DPV (G*D)



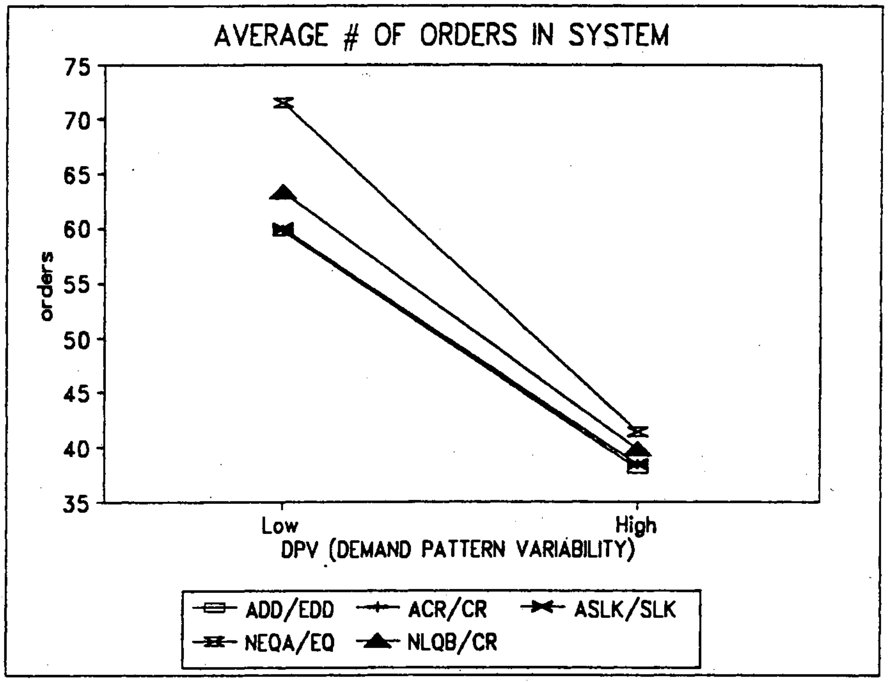
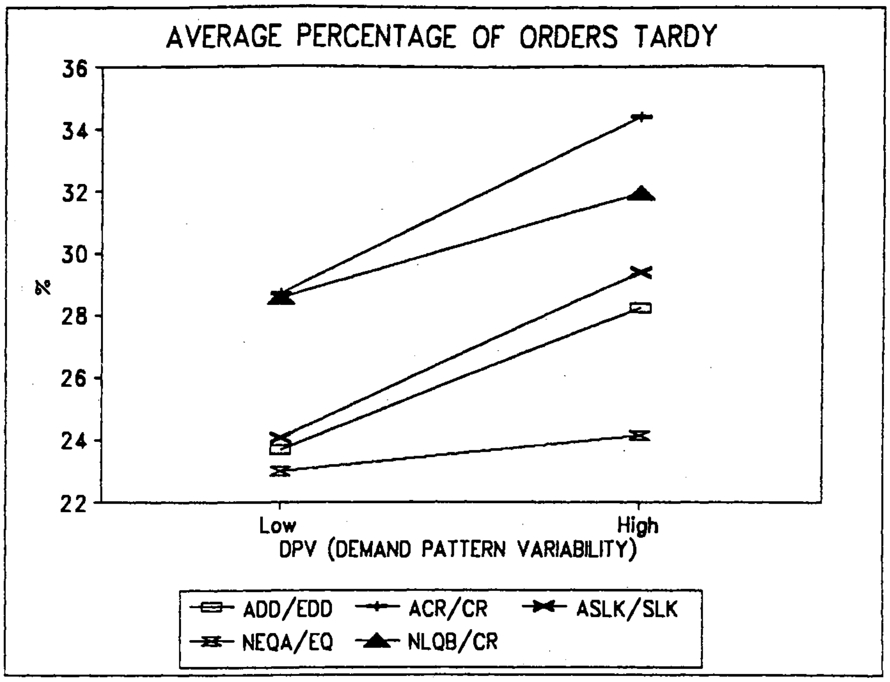


Figure 5.3 (Continued)



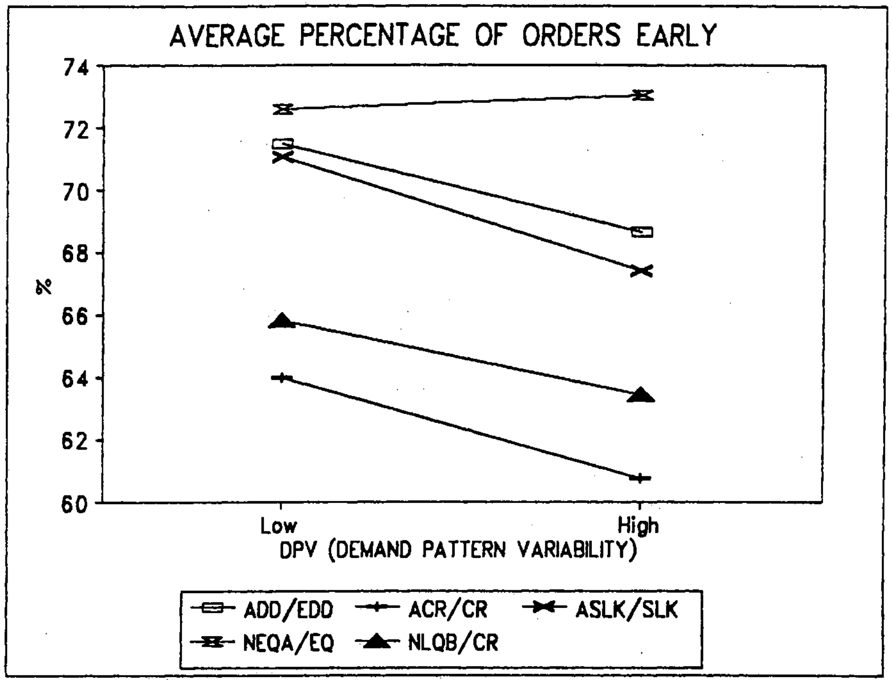
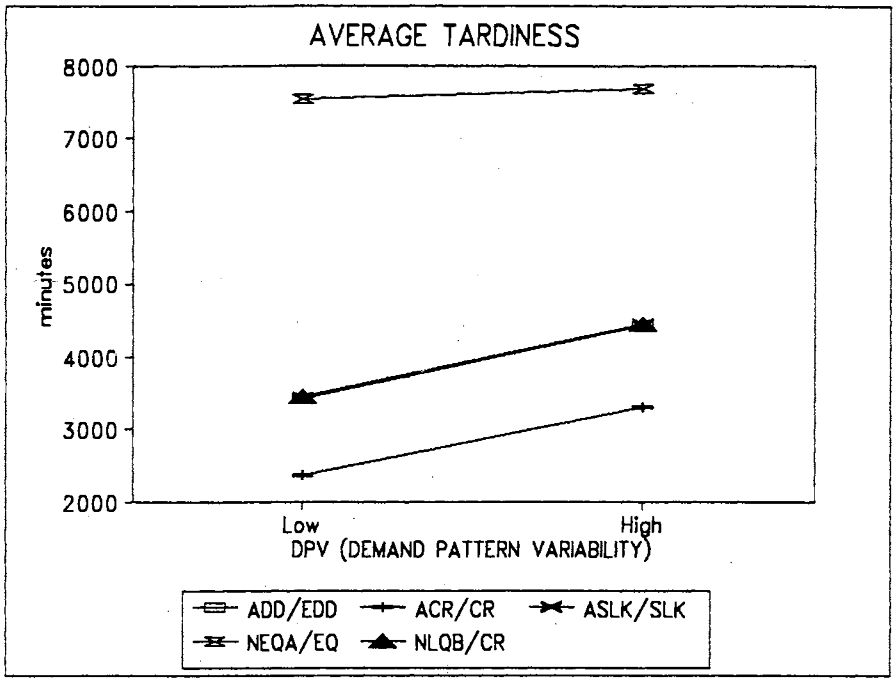


Figure 5.3 (Continued)



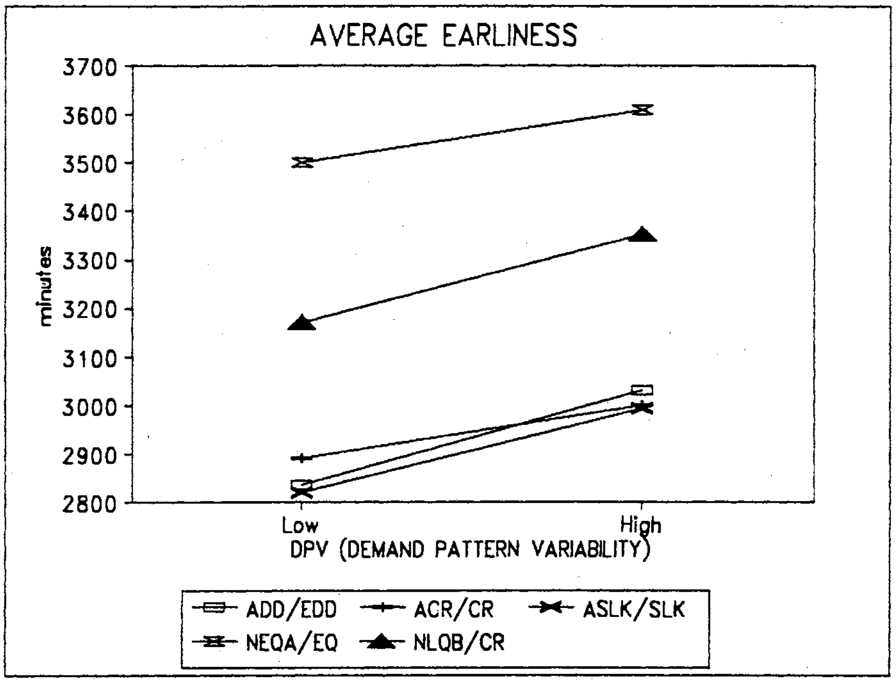


Figure 5.3 (Continued)

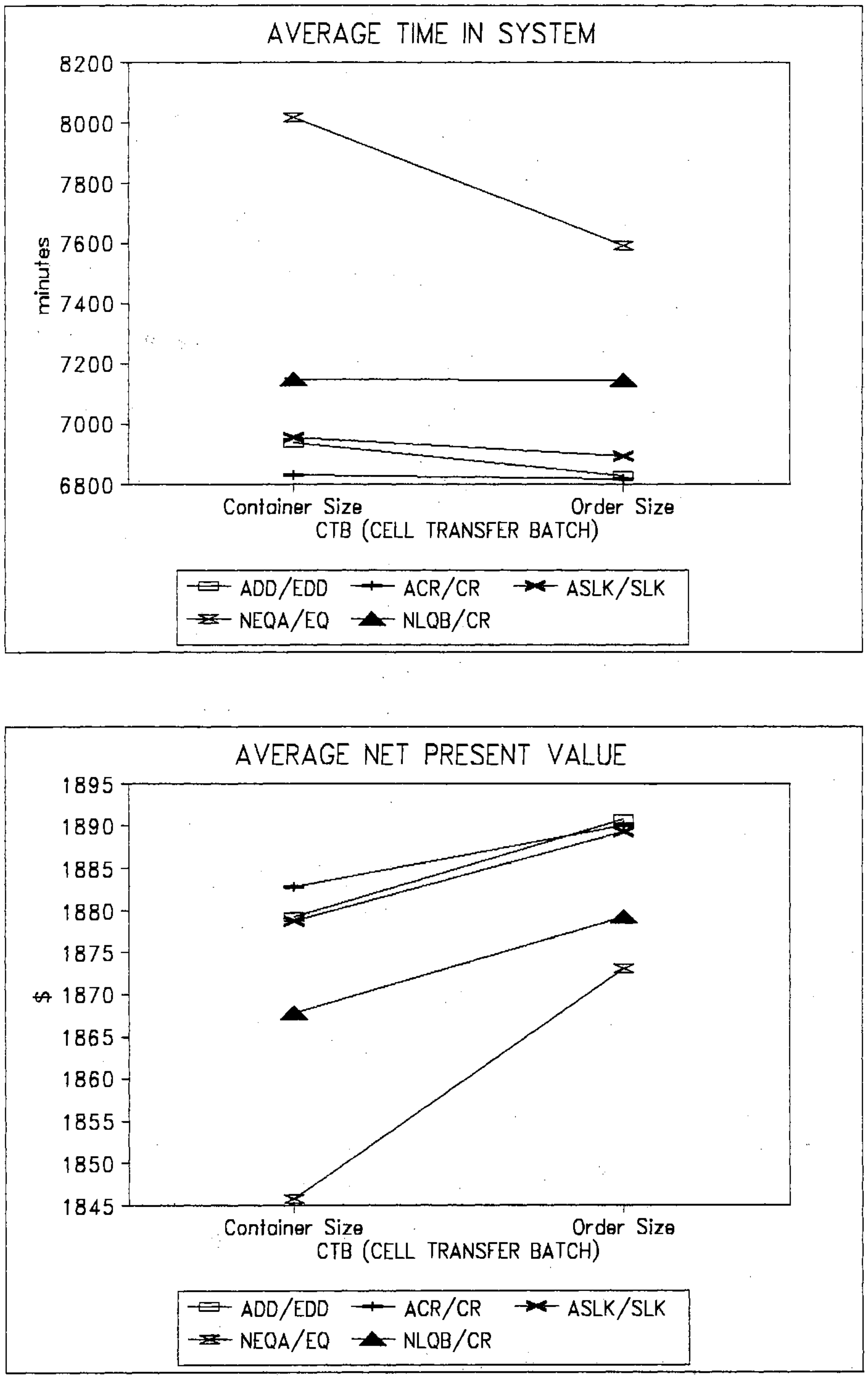


Figure 5.4 Interaction between GSH and CTB (G*C)

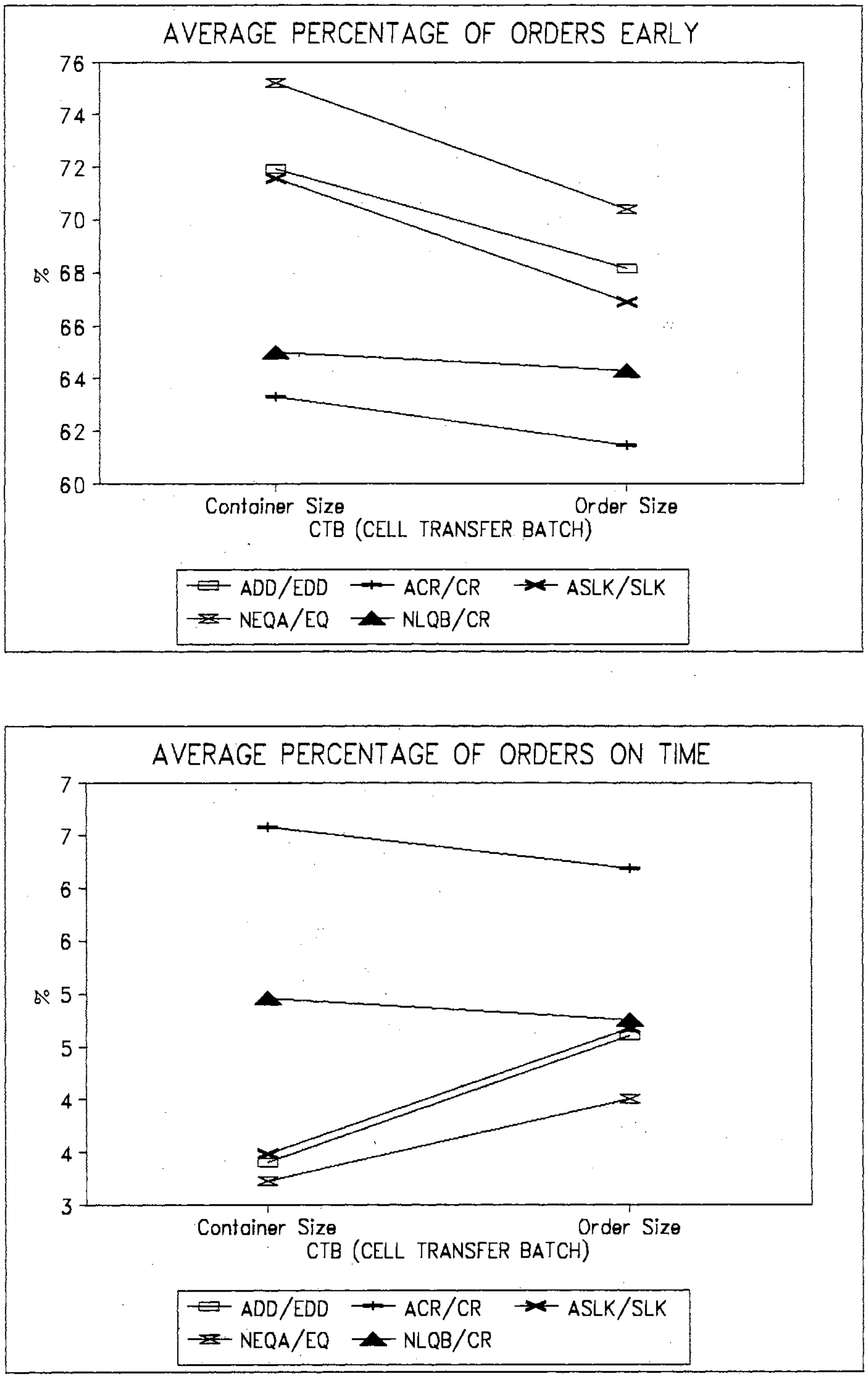
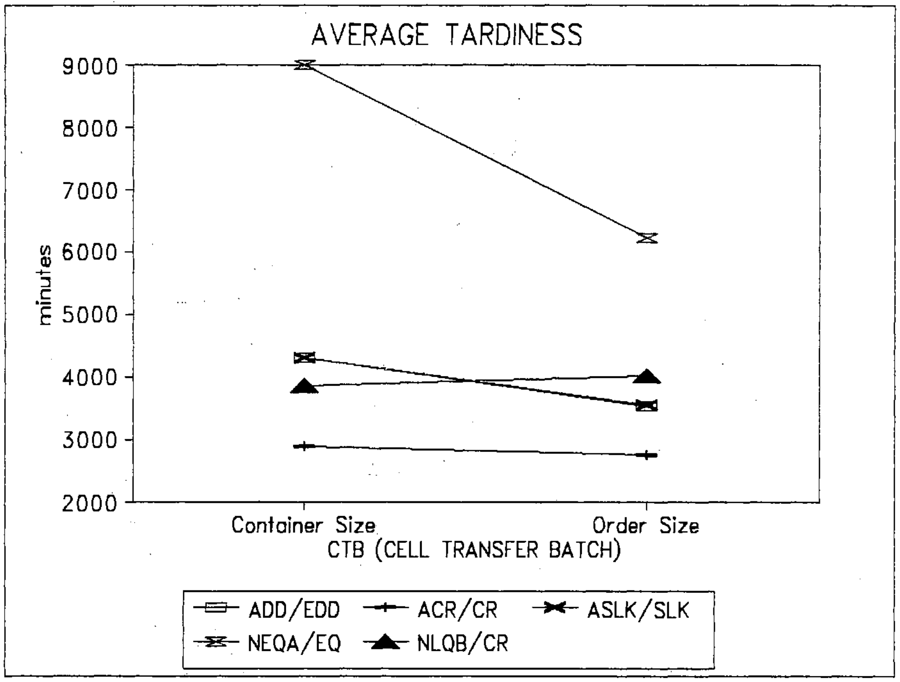


Figure 5.4 (Continued)



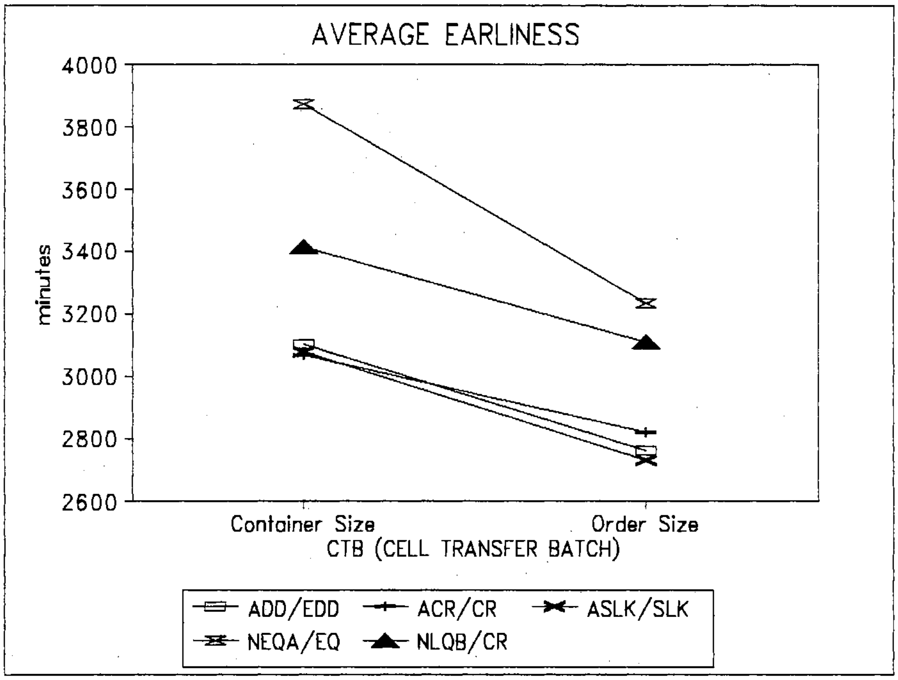


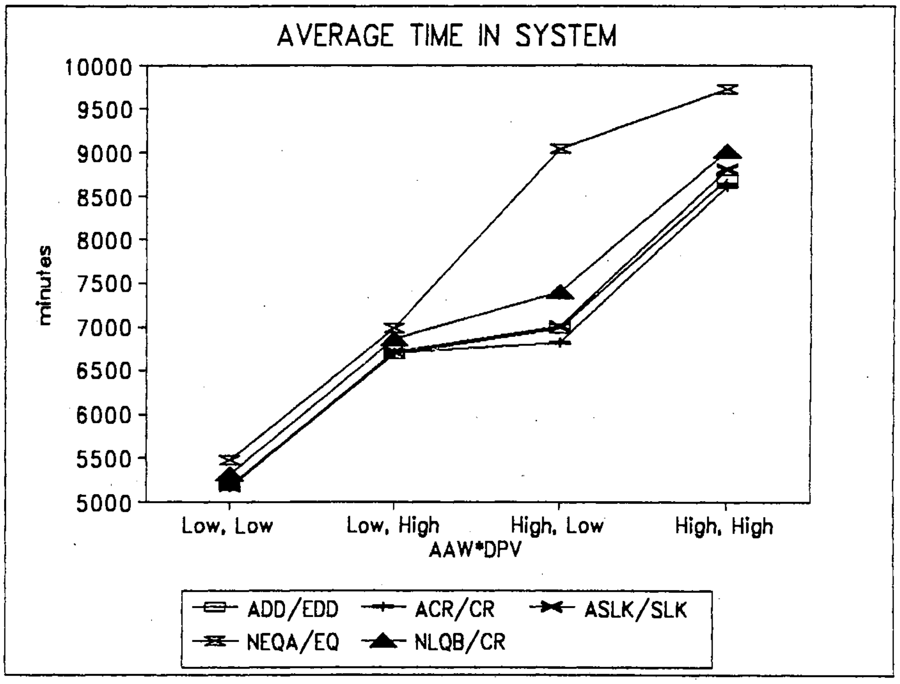
Figure 5.4 (Continued)

significance since the change patterns of NEQA/EQ and NLQB/CR (especially NEQA/EQ) had different change patterns compared with the other three heuristics when the level of cell transfer batch was changed.

Third Order Interactions among Heuristic Factor and Any Two Shop Environmental Factors

Interaction among Group Scheduling Heuristic, Average Annual Workload, and Demand Pattern Variability (G*A*D)

Graphic representations of this interaction with respect to various performance measures are shown in Figure 5.5. As can be seen, the change patterns of the three heuristics (ADD/EDD, ACR/CR, and ASLK/SLK) were similar when the level of AAW*DPV (or A*D) was changed on the performance measures included in Figure 5.5. This interaction exhibited statistical significance since NEQA/EQ and NLQB/CR (especially NEQA/EQ) had different change patterns compared with the other three heuristics when the level of AAW*DPV was changed.



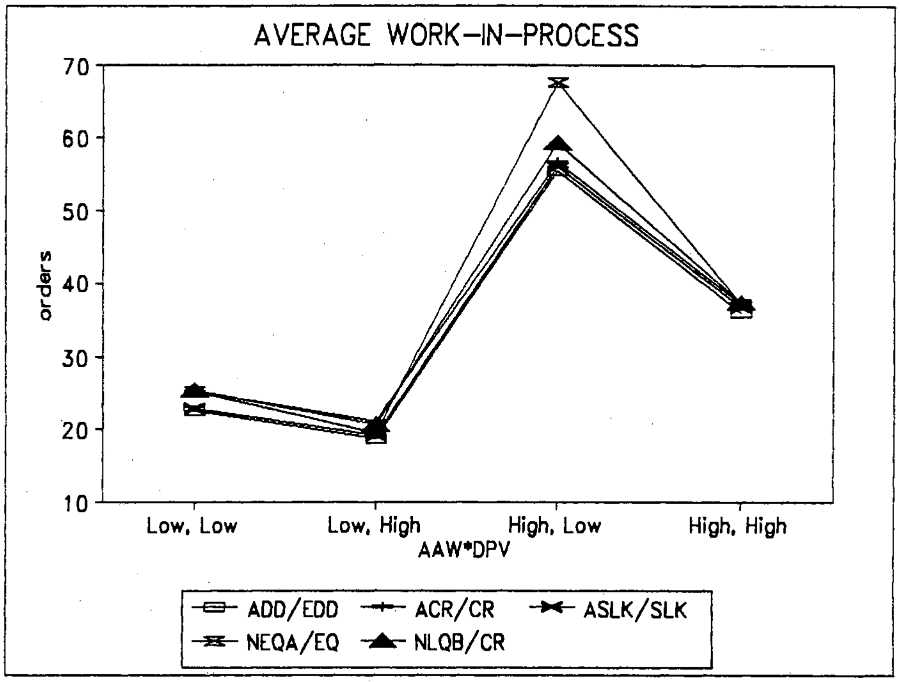
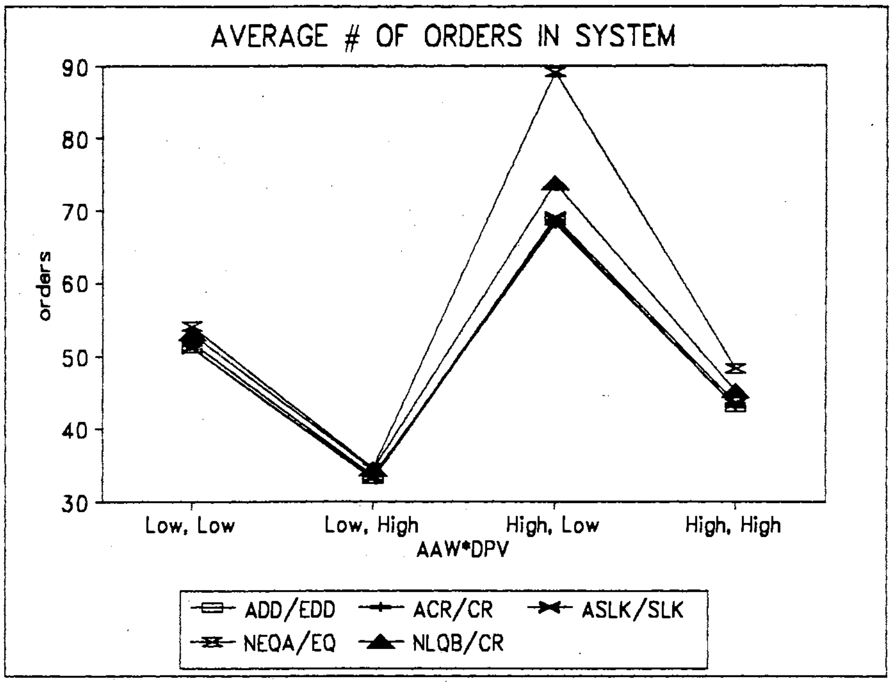


Figure 5.5 Interaction among GSH, AAW, and DPV (G*A*D)



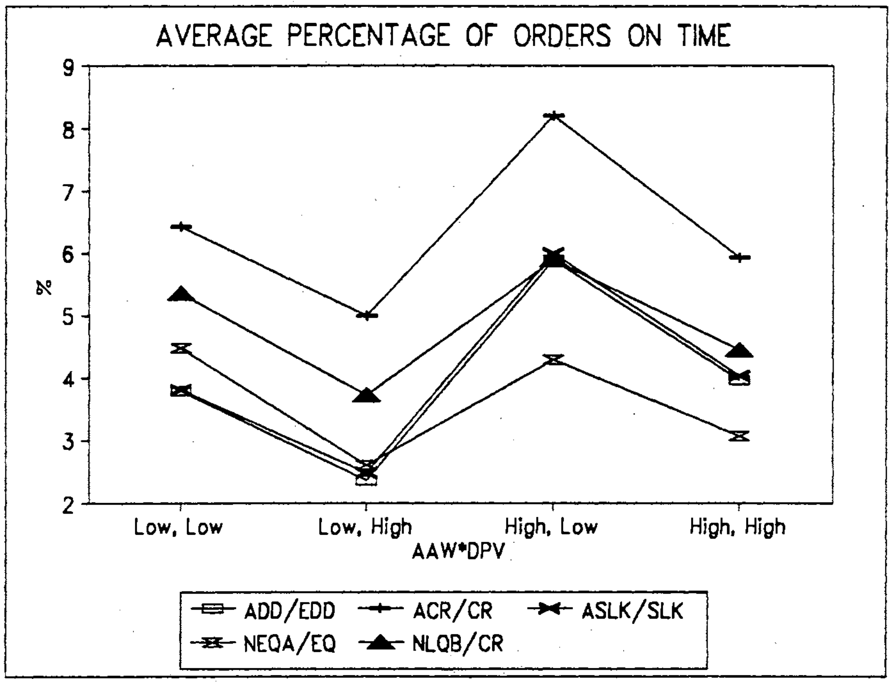


Figure 5.5 (Continued)

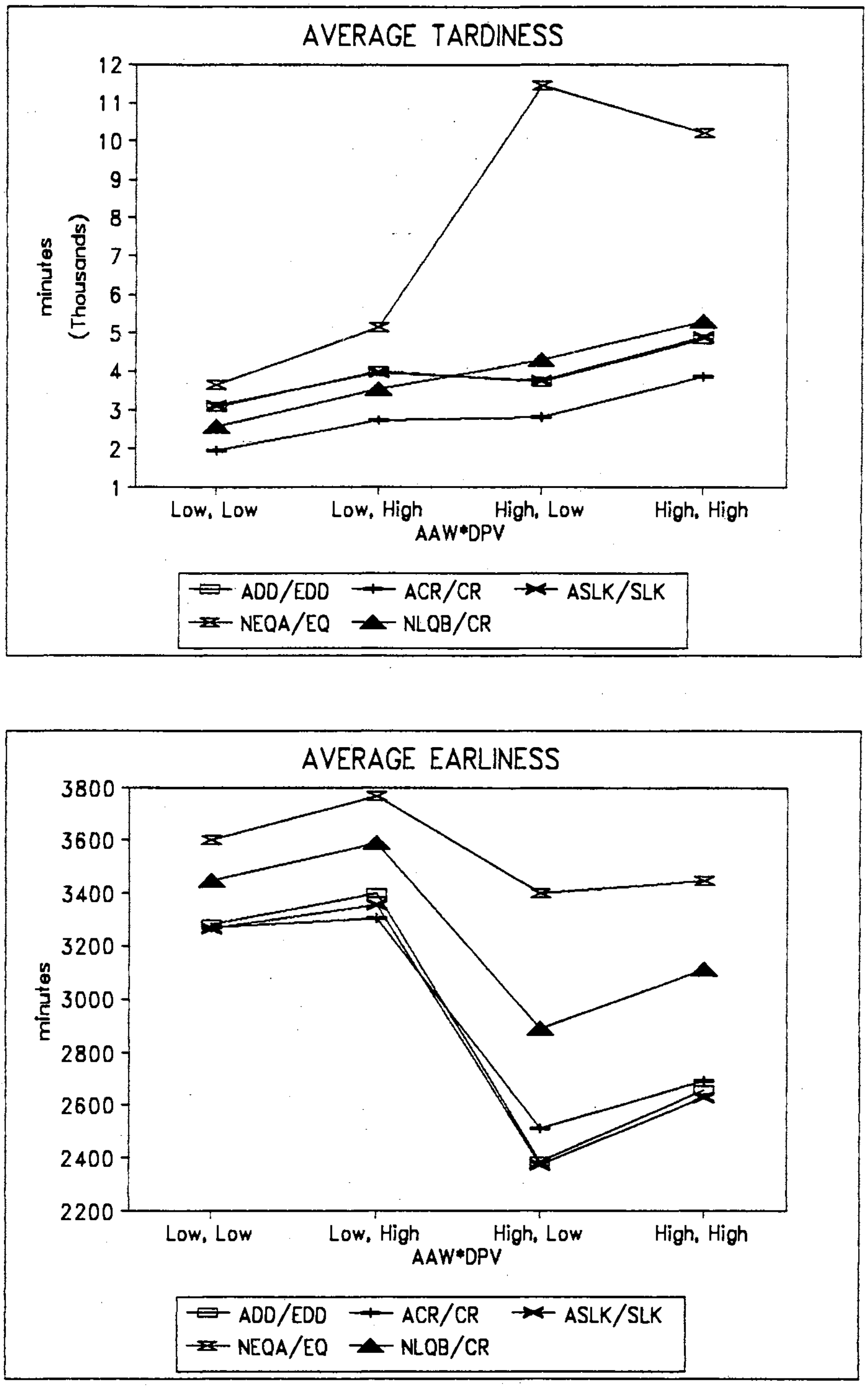
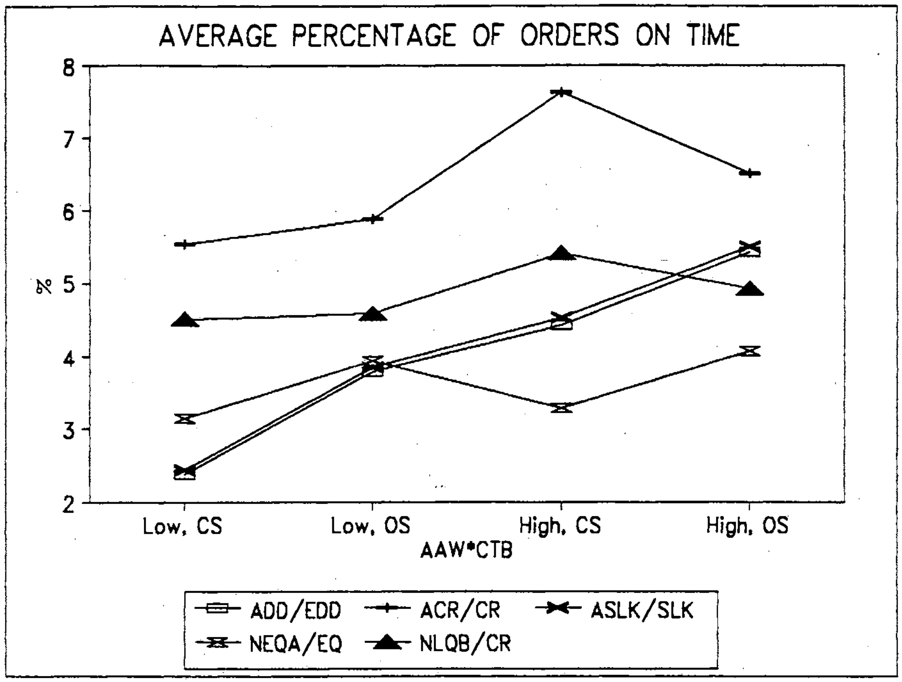


Figure 5.5 (Continued)

Interaction among Group Scheduling Heuristic, Average Annual Workload, and Cell Transfer Batch (G*A*C) Graphic representations of this interaction with respect to various performance measures are shown in Figure 5.6. In Table 5.5, this interaction exhibited statistical significance on four performance measures. For the graphs included in Figure 5.6, the change patterns of ADD/EDD and ASLK/SLK were very similar when the level of AAW*CTB (or A*C) was changed. This interaction exhibited statistical significance since the change patterns of ACR/CR, NEQA/EQ and NLQB/CR (especially NEQA/EQ) were different from each other and, also, different from the other two heuristics when the level of AAW*CTB was changed.

Interaction among Group Scheduling Heuristic, Demand Pattern Variability, Cell Transfer Batch (G*D*C) Graphic representations of this interaction with respect to various performance measures are shown in Figure 5.7. In Table 5.5, this interaction exhibited statistical significance on four performance measures. For the graphs included in Figure 5.7, the change patterns of ADD/EDD and ASLK/SLK were very similar when the level of DPV*CTB (or D*C) was changed. This interaction exhibited statistical significance since the change patterns of ACR/CR, NEQA/EQ and NLQB/CR (especially NEQA/EQ) were different each other and, also, different from the other two heuristics when the level of DPV*CTB was changed.



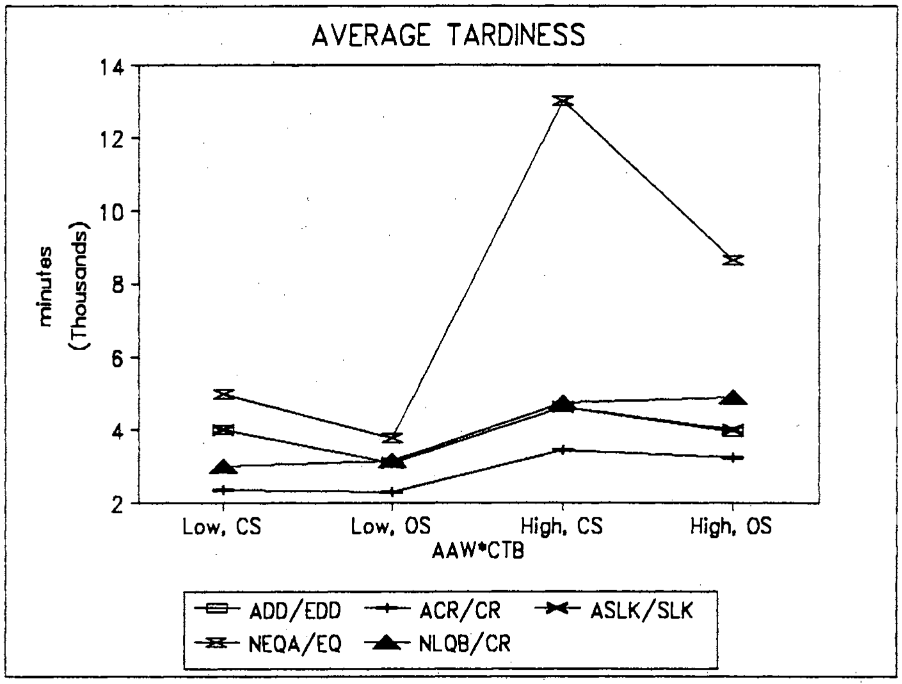


Figure 5.6 Interaction among GSH, AAW, and CTB (G*A*C)

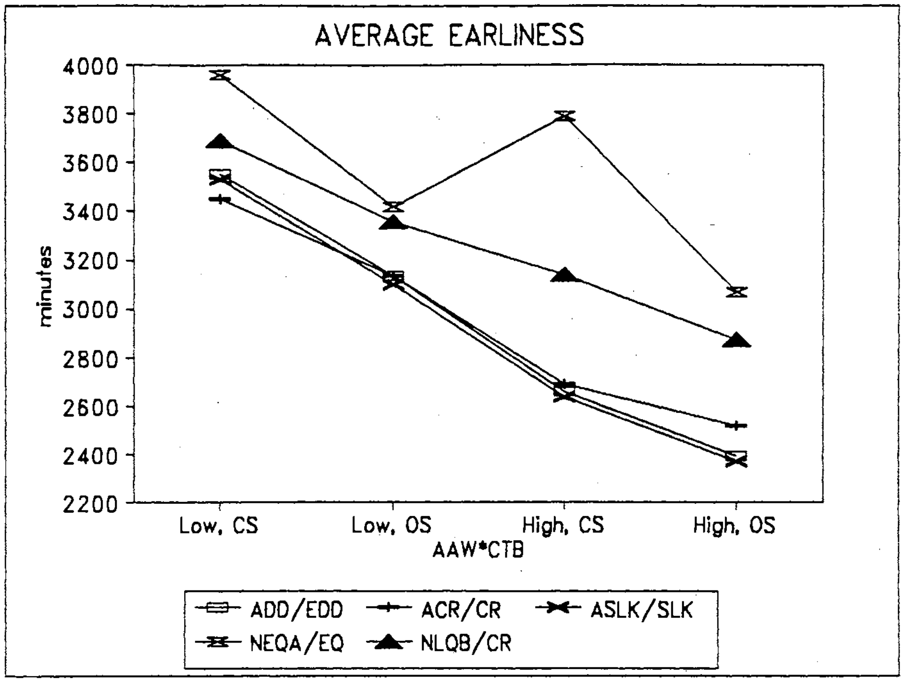
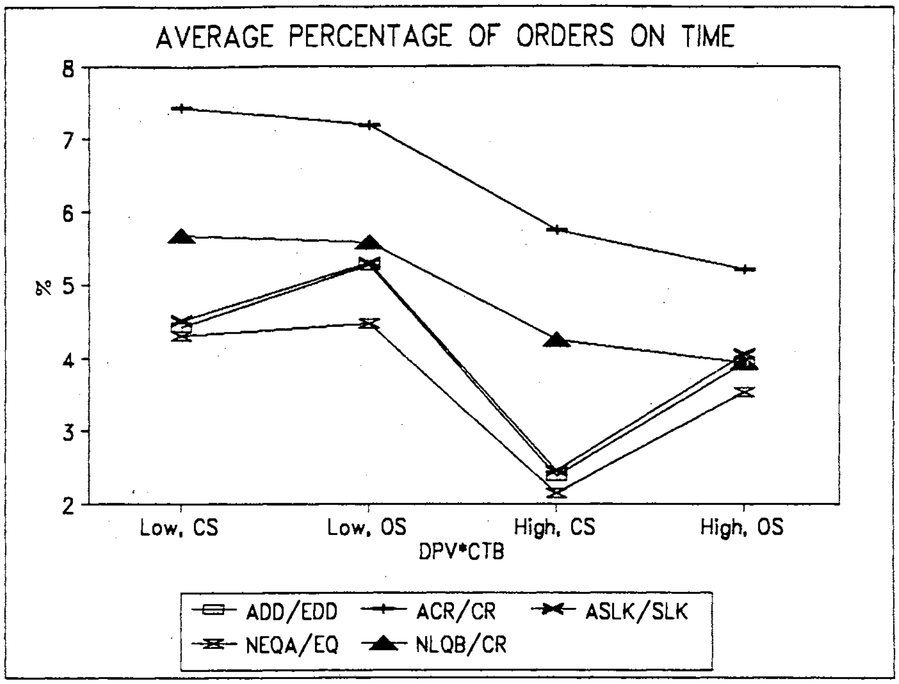


Figure 5.6 (Continued)



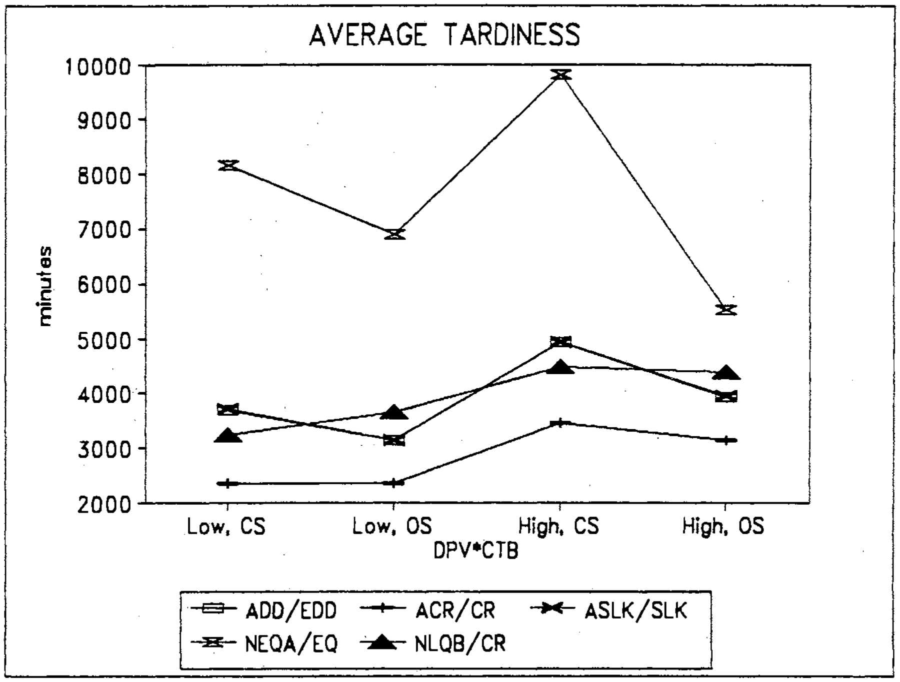


Figure 5.7 Interaction among GSH, DPV, and CTB (G*D*C)

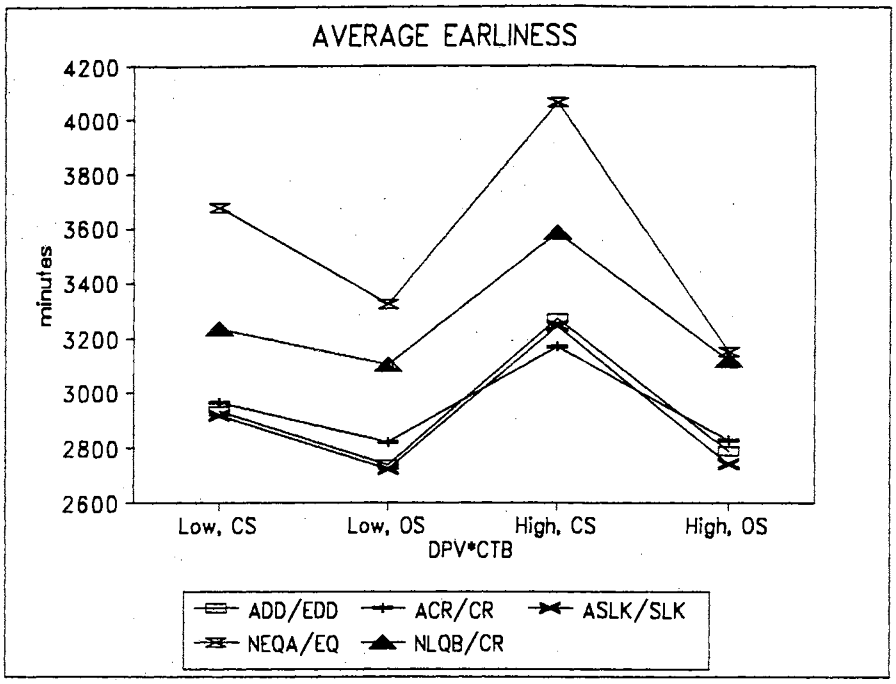


Figure 5.7 (Continued)

Conclusions Drawn from Analysis of Variance

A conclusion drawn from the results of the analysis of variance (ANOVA) is that the three shop environmental factors (i.e., average annual workload, demand pattern variability, and cell transfer batch) all impacted the performance of the five group scheduling heuristics. This means the levels chosen for each of the three shop environmental factors are far enough apart that a discernible difference in performance of the heuristics can be assured.

Another conclusion drawn from the results of the analysis of variance is that the heuristics at the low levels of the three shop environmental factors outperformed those at the high levels of the three factors on most of the measures used. One exception is that the heuristics with the cell transfer batch which equalled the order size outperformed those with the cell transfer batch which equalled the container size on most of the measures when the average annual workload and demand pattern variability were both set to high level.

The second order interactions between any two experimental factors were statistically significant on most of the measures. While the third order interactions among any three experimental factors were not statistically significant on most of the measures, the interaction of A*D*C was significant on all measures. For the highest order interaction (i.e., G*A*D*C), the ANOVA results show that it was not statistically significant on all measures.

For most of the second order interactions between group scheduling heuristic and any shop environmental factor shown in Figures 5.2, 5.3, and 5.4, the change patterns of ADD/EDD, ACR/CR, and ASLK/SLK were similar. The reason that these second order interactions exhibited statistical significance is because NEQA/EQ and NLQB/CR (especially NEQA/EQ) had different change patterns compared with the other three heuristics. For most of the third order interactions between group scheduling heuristic and any two shop environmental factor shown in Figures 5.5, 5.6, and 5.7, the change patterns of ADD/EDD and ASLK/SLK were very similar. The reason that these third order interactions exhibited statistical significance is because ACR/CR, NEQA/EQ and NLQB/CR (especially NEQA/EQ) had different change patterns compared with to the other two heuristics.

Conclusions

In order to further summarize the results of the paired t-test analysis (see Table 5.3) and to gain a better handle on heuristic preferability under different shop environmental conditions across all performance measures, a heuristic preferability table (i.e., Table 5.6) was developed. In Table 5.6, if two or three heuristics were

TABLE 5.6

HEURISTIC PREFERABILITY TABLE

Shop Condition	Performance Measure				
	Time in System	Queue Waiting Time	Net Present Value	Work-in-Process	# of Orders in Sys.
AAW=High DPV=High CTB=OS	ADD/EDD ACR/CR	ADD/EDD	ADD/EDD ACR/CR	ADD/EDD ASLK/SLK	ADD/EDD ACR/CR
AAW=High DPV=High CTB=CS	ACR/CR	ADD/EDD	ACR/CR	ADD/EDD ASLK/SLK	ADD/EDD ACR/CR ASLK/SLK
AAW=High DPV=Low CTB=OS	ACR/CR ADD/EDD	ADD/EDD ACR/CR ASLK/SLK	ACR/CR ADD/EDD ASLK/SLK	ADD/EDD ACR/CR ASLK/SLK	ACR/CR ADD/EDD
AAW=High DPV=Low CTB=CS	ACR/CR	ADD/EDD ACR/CR ASLK/SLK	ACR/CR ADD/EDD ASLK/SLK	ADD/EDD ASLK/SLK	ADD/EDD ACR/CR ASLK/SLK
AAW=Low DPV=High CTB=OS	ADD/EDD	ADD/EDD	ADD/EDD	ADD/EDD	ADD/EDD
AAW=Low DPV=High CTB=CS	ACR/CR ADD/EDD ASLK/SLK	ADD/EDD NEQA/EQ ASLK/SLK	ACR/CR	ADD/EDD	ADD/EDD ASLK/SLK
AAW=Low DPV=Low CTB=OS	ADD/EDD ASLK/SLK ACR/CR	ADD/EDD	ACR/CR ADD/EDD	ADD/EDD	ADD/EDD ASLK/SLK ACR/CR
AAW=Low DPV=Low CTB=CS	ACR/CR ADD/EDD ASLK/SLK	ADD/EDD ASLK/SLK	ACR/CR	ADD/EDD	ADD/EDD ASLK/SLK

TABLE 5.6 (Continued)

Shop Condition	Performance Measure				
	Percent Tardy	Percent Early	Percent On Time	Tardiness	Earliness
AAW=High DPV=High CTB=OS	NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK
AAW=High DPV=High CTB=CS	NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK ACR/CR ADD/EDD
AAW=High DPV=Low CTB=OS	NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK ADD/EDD
AAW=High DPV=Low CTB=CS	NEQA/EQ	ACR/CR	ACR/CR	ACR/CR	ASLK/SLK ADD/EDD
AAW=Low DPV=High CTB=OS	ADD/EDD	NLQB/CR	ACR/CR	ACR/CR	ASLK/SLK
AAW=Low DPV=High CTB=CS	ADD/EDD ASLK/SLK	NLQB/CR	ACR/CR	ACR/CR	ACR/CR ASLK/SLK
AAW=Low DPV=Low CTB=OS	ADD/EDD ASLK/SLK	NLQB/CR NEQA/EQ	ACR/CR	ACR/CR	ASLK/SLK ADD/EDD ACR/CR
AAW=Low DPV=Low CTB=CS	ADD/EDD ASLK/SLK	NLQB/CR NEQA/EQ	ACR/CR	ACR/CR	ACR/CR ASLK/SLK

listed in a grid, the performance of these heuristics was not statistically different, but was ranked in order of the performance.

Given one of the eight shop environmental conditions and one of the ten performance measures examined in this research, schedulers can easily find the best performing heuristic from the heuristic preferability table (Table 5.6). In case of multiple performance measures (i.e., multiple objectives) are under consideration, first, schedulers can look up Table 5.6 and try to find the best performing heuristic. If none can be found by only examining Table 5.6, Table 5.3 can be used to assist in finding a preferable heuristic which best meets all performance criteria.

For example, if our objectives are to minimize the average time in system and average percentage of orders tardy, and the shop environmental condition is set to high levels of the average annual workload and demand pattern variability, and low level of the cell transfer batch, from the heuristic preferability table (Table 5.6) we cannot find a preferred heuristic. By referring to Table 5.3 and assuming that weights (from 5 to 1) can be assigned to heuristics according to their ranking (from 1 to 5), the total weights for ADD/EDD, ACR/CR, ASLK/SLK, NEQA/EQ, and NLQB/CR are 8, 7, 8, 7, and 6, respectively. Therefore, ADD/EDD or ASLK/SLK can be selected as the preferred heuristic in this example.

Even in the situations that we do not have much knowledge of the shop environmental conditions, the heuristic preferability table still can provide valuable suggestion if the performance measures under considerations are known. Take an example. If we only know that the demand pattern variability of the system is at the low level and the measure of the average net present value is the major concern, from the heuristic preferability table (Table 5.6), we can find that ACR/CR is the preferred heuristic.

Finally, based on the results of the paired t-test analysis and the analysis of variance, the conclusions for this research effort can be drawn as follows:

(1) No single heuristic always outperformed the other four heuristics under all shop environmental conditions with respect to all performance measures.

(2) On several measures, the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ACR/CR, ADD/EDD, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ). Moreover, on these measures, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

(3) Of the five heuristics, overall, ACR/CR was the best performing heuristic, followed by ADD/EDD and ASLK/SLK. NEQA/EQ and NLQB/CR were the worst and the second worst performing heuristics on most of the measures, respectively.

(4) The results of the analysis of variance (ANOVA) show that the three shop environmental factors (i.e., average annual workload, demand pattern variability, and cell transfer batch) all impacted the performance of the five group scheduling heuristics.

(5) The heuristics at the low levels of the three shop environmental factors outperformed those at the high levels of the three factors on most of the measures used. One exception is that the heuristics with the cell transfer batch which equalled the order size outperformed those with the cell transfer batch which equalled the container size on most of the measures when the average annual workload and demand pattern variability were both set to high level.

(6) The heuristic preferability table can provide guidance for schedulers in the selection of a preferable heuristic from the five group scheduling heuristics based on the shop environmental conditions and the performance measures that are most important in their industry.

(7) In this research, ACR/CR which outperformed ADD/EDD was the best performing heuristic with respect to the average tardiness. Compared with a recent study (Russell and Philipoom, 1991), the authors investigated several group scheduling heuristics in a single flow shop cell with five workcenters when early shipments were allowed. The results of their study showed that EDD/EDD was the best performer with respect to the average tardiness when the total work content (TWK) was used to determine the due date. Therefore, it can be concluded that the forbidden early shipment and workcenter sharing have a significant influence on the performance of group scheduling heuristics in the flow shop cellular system.

CHAPTER VI

SUMMARY, CONTRIBUTIONS, AND FURTHER RESEARCH

Summary

The goal of this research was to evaluate the performance of group scheduling heuristics in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment. First, this study proposed a group scheduling process in cellular manufacturing. The purpose of the group scheduling process was to define how and when orders enter the system, being processed, and leave the system. This process included the following stages: order arrival (or order entry), order release, queue selection, job dispatching, order storage (if an order is completed before due date) and order shipment.

Second, an effort was made to define the system, to identify experimental factors, and to choose performance measures. The system definition included the shop model, order characteristics, and all assumptions made. The shop model consisted of two flow shop cells; each of the flow shop cells had five workcenters, with the last workcenter shared between cells. The experimental factors sought to identify the critical factors impacting the performance of the cellular system and to define their critical levels. The critical factors which were identified included the group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch. Ultimately, five group scheduling heuristics and two levels of each of the other three factors were selected for investigation. The five heuristics included the three heuristics which considered jobs' due dates in both their queue selection rules and job dispatching rules (i.e., ADD/EDD, ACR/CR, and ASLK/SLK), and the two heuristics which included workcenter status in their queue selection rules (i.e., NEQA/EQ and NLQB/CR). The performance measures, which were chosen to collect the statistics, included the average time in system, average queue waiting time, average net present value, average work-in-process, average number of orders in system, average percentage of orders tardy, average percentage of orders early, average percentage of orders on time, average tardiness, and average earliness.

Third, a full factorial design was chosen to conduct a series of experiments for evaluating the performance of the five group scheduling heuristics under each of eight experimental conditions (or shop environmental conditions). It should be noted that this study held machine workloads and part demands constant when the five heuristics was evaluated under each of eight experimental conditions. These constraints forced the study in two directions. One was that because of the number of factors controlled it required that system input data be generated rather than using data from an existing system. The other was that it contributed to the decision to use computer simulation as the research vehicle to carry out the experiments. After choosing the research vehicle, the simulation models were developed, verified, validated, and carried out.

Finally, the paired t-test analysis and the analysis of variance (ANOVA) were used as statistical tools to analyze the simulation results. A conclusion drawn from the results of the paired t-test analysis is that no single heuristic always outperformed the other four heuristics under all shop environmental conditions with respect to all performance measures.

Another conclusion drawn is that the three heuristics which consider jobs' due dates in both their queue selection rules and job dispatching rules (ADD/EDD, ACR/CR, and ASLK/SLK) consistently outperformed the other heuristics (NLQB/CR and NEQA/EQ) on the following measures: the average time in system, average net present value, average number of orders in system, average earliness, and average tardiness with the high average annual workload. Moreover, on these measures, NLQB/CR which considers jobs' due dates in its job dispatching rule consistently outperformed NEQA/EQ.

Of the five heuristics, overall, ACR/CR was the best performing heuristic, followed by ADD/EDD and ASLK/SLK.

ACR/CR performed best on the measures of the average time in system, average net present value, average percentage of orders on time, average tardiness, and average percentage of orders early with the high average annual workload. ADD/EDD performed best on the measures of the average queue waiting time, average work-in-process, average number of orders in system, and average percentage of orders tardy with the low average annual workload. ASLK/SLK performed best on the measure of the average earliness.

NEQA/EQ and NLQB/CR were the worst and the second worst performing heuristics on most of the performance measures, respectively. Exceptionally, NEQA/EQ exhibited excellent performance on the measure of the average percentage of orders tardy when the average annual workload was set to high level, and NLQB/CR was the best performer on the measure of the average percentage of orders early when the average annual workload was set to low level.

A conclusion drawn from the results of the analysis of variance (ANOVA) is that the three shop environmental factors (i.e., average annual workload, demand pattern variability, and cell transfer batch) all impacted the performance of the five group scheduling heuristics. Another conclusion drawn is that the heuristics at the low levels of the three shop environmental factors outperformed those at the high levels of the three factors on most of the measures used. One exception is that the heuristics with the cell transfer batch which equalled the order size outperformed those with the cell transfer batch which equalled the container size on most of the measures when the average annual workload and demand pattern variability were both set to high level.

In order to further summarize the results of paired t-test analysis (Table 5.3) and to gain a better handle on heuristic preferability under different shop environmental conditions across all performance measures, a heuristic preferability table (Table 5.6) was developed. These tables (i.e., Tables 5.3 and 5.6) provide guidance for schedulers in the selection of a preferable heuristic from the five group scheduling heuristics based on the shop environmental conditions and the performance measures that are most important in their industry.

Compared with previous research on group scheduling in a single flow shop cell when early shipments were allowed (e.g., Russell and Philipoom, 1991), the preferred heuristics in this study are different from the preferred heuristics in previous research. Therefore, it can be concluded that the forbidden early shipment and workcenter sharing have a significant influence on the performance of group scheduling heuristics in the flow shop cellular system.

Research Contributions

In reviewing the accomplishments of this research effort, it can be concluded that the goal and objectives of this study have been achieved. The primary contribution of this research is to gain an understanding of the performance of various group scheduling heuristics in a flow shop cellular system with workcenter sharing for the forbidden early shipment environment. It is felt that the following contributions would be made to the body of knowledge of the production sequencing and scheduling in cellular manufacturing systems, while pursuing the above mentioned primary contribution:

(1) Realization of the influence of the forbidden early shipment and workcenter sharing on the performance of group scheduling heuristics in the flow shop cellular system.

(2) Development of five group scheduling heuristics in the flow shop cellular system with workcenter sharing for the forbidden early shipment environment.

(3) Understanding of the ranking of the five group scheduling heuristics under each of the shop environmental conditions with respect to each of the performance measures.

(4) Creation of the heuristic preferability table which provides guidance for schedulers in the selection of a preferable heuristic based on the shop environmental conditions and the performance measures that are most important in their industry.

(5) Identification and verification (by ANOVA) of the critical factors (i.e., group scheduling heuristic, average annual workload, demand pattern variability, and cell transfer batch) and their levels which impact the performance of the cellular system.

(6) Presentation of a group scheduling process in cellular manufacturing with forbidden early shipment.

(7) Inclusion of the net present value measure in evaluating the performance of group scheduling heuristics in cellular manufacturing.

Areas for Further Research

By necessity, the scope of this research has been limited to four experimental factors, five group scheduling heuristics, and a flow shop cellular system with workcenter sharing. However, this research has provided the foundation for further research. Some examples of such research directions are:

(1) Since the results of this research were obtained through the simulation of a hypothetical shop model, the question arises as to the applicability of the results to a real production system. We can see this research as a preliminary experimental study in the area of group scheduling in cellular manufacturing with forbidden early shipment. Further research needs to be done to develop new group scheduling heuristics, which are expected to perform efficiently and effectively, and then to examine these new heuristics in broader scenarios of cellular manufacturing systems (or flexible manufacturing systems). These scenarios can have different shop types (e.g., flow shop cells and job shop cells), cell sizes, and part mixes, etc.

(2) In this research, only the four factors which appeared to have a major impact on the performance of heuristics were selected and only two levels of each factor (except for group scheduling heuristic) were chosen. Further research needs to be done to investigate the effects of different factors and/or to include more levels of each factor in the investigation. For example, one conclusion drawn from the results of the analysis of variance in this research was that the cell transfer batch impacted the performance of the group scheduling heuristics. Therefore, it is logical to extend this research to include other transfer batches

(e.g., the use of standard containers with the capacity of 1, 5, or 15, etc.)

(3) It was assumed in this research that the cellular system was operated in a make-to-order environment. All group scheduling heuristics evaluated in this research were developed to be used on this platform. However, some cellular manufacturing systems have been implemented in other types of production planning and control systems such as material requirements planning (MRP) systems, reorder point (ROP) systems, pull systems, and optimized production technology (OPT) systems (Wemmerlov and Hyer, 1989). Further research should be done to examine the group scheduling heuristics for such systems.

(4) This research did not consider the functions of the production planning and control. Production planning and control includes the functions of directing or regulating the movement of goods through the manufacturing cycle. Due to its complexity, the regulation of the conversion process requires a hierarchical decision process consisting of the following three levels (Hyer and Wemmerlov 1982):

Level 1: The determination of when and in what quantities final products are to be produced;

Level 2: The determination of which specific parts are to be produced during a specific time period and in what quantities these parts should be produced; and

Level 3: The determination of when and in what order jobs (operations) should be processed at various workcenters.

We can use MRP systems as an example to illustrate the three levels. The first level corresponds to the generation of the master production schedule (MPS), the second level represents the explosion of the MPS into time-phased requirements for component parts, and the third level is equivalent to the disaggregation of orders into operations and the execution of these, as directed by the shop floor control system (Hyer and Wemmerlov, 1982). The group scheduling problems fall into the third level. Group scheduling provides an opportunity to improve the shop performance. But, studying group scheduling as an isolated function may result in sub-optimal solutions. Therefore, further research should be done to consider all the functions included in the three levels when studying the group scheduling problems in order to optimize the overall performance of cellular manufacturing systems.

BIBLIOGRAPHY

Ahmed, I., and W. W. Fisher, (1992). Due Date Assignment, Job Order Release, and Sequencing Interaction in Job Shop Scheduling, Decision Sciences, Vol. 23, No. 3, pp. 633-647.

Ang, C. L., and P. C. T. Willey, (1984). A Comparative Study of the Performance of Pure and Hybrid Group Technology Manufacturing Systems using Computer Simulation Techniques, International Journal of Production Research, Vol. 22, No. 2, pp. 193-233.

Baker, K. R., (1974). Introduction to Sequencing and Scheduling, John Wiley and Sons Publishing, New York.

Baker, K. R., and J. J. Kanet, (1983). Job Shop Scheduling with Modified Due Dates, Journal of Operations Management, Vol. 4, No. 1, pp. 11-22.

Banks, J., and J. S. Carson, (1984). Discrete-Event Simulation, Prentice-Hall Publishing, Englewood Cliffs, NJ.

Bratley, P., B. L. Fox, and L. E. Schrage, (1987). A Guide to Simulation, Springer-Verlag Publishing, New York.

Burbidge, J. L., (1971). Production Flow Analysis, The Production Engineer, Vol. 50, No. 4, pp. 139-152.

Burbidge, J. L., (1975). The Introduction of Group Technology, John Wiley and Sons Publishing, New York.

Campbell, H. G., R. A. Dudek, and M. L. Smith, (1970). A Heuristic Algorithm for the n Job, m Machine Sequencing Problem, Management Science, Vol. 16, No. 10, pp. B630-B637.

Carson, J. S., (1989). Verification and Validation: A Consultant's Perspective, Proceedings of the 1989 Winter Simulation Conference, pp. 552-558.

Chang, C.-P., (1989). Scheduling in a Group Technology Manufacturing Cell with Due Date Consideration, Unpublished Ph.D. Dissertation, University of Pittsburgh.

Cho, K.-K., (1982). Group Scheduling under Due Date Constraints in Multi-Stage Manufacturing Systems, Unpublished Ph.D. Dissertation, Pennsylvania State University.

Cho, K.-K., E. E. Enscore, Jr., and I. Ham, (1982). A Heuristic Algorithm for Multi-Stage Group Scheduling Problem to Minimize Total Tardiness, Proceedings of the 10th North American Manufacturing Research Conference, Ontario, Canada, pp. 514-517.

Christy, D. P., (1984). Job Shop Scheduling with Stochastic Due Dates, Unpublished Ph.D. Dissertation, University of Georgia.

Christy, D. P., and J. J. Kanet, (1988). Open Order Rescheduling in Job Shops with Demand Uncertainty: a Simulation Study, Decision Sciences, Vol. 19, No. 4, pp. 801-818.

Christy, D. P., and J. J. Kanet, (1990). Manufacturing Systems with Forbidden Early Shipment: Implications for Choice of Scheduling Rules, International Journal of Production Research, Vol. 28, No. 1, pp. 91-100.

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

Farrington, P. A., (1991). Development of a Methodology for Selecting the Appropriate System Design Based on the Prevailing Characteristics of the Production Environment, Unpublished Ph.D. Dissertation, Oklahoma State University.

Flynn, B. B., (1987). Repetitive Lots: The Use of a Sequence-Dependent Set-up Time Scheduling Procedure in Group Technology and Traditional Shops, Journal of Operations Management, Vol. 7, Nos. 1 and 2, pp. 203-216.

Foo, F. C., and J. G. Wager, (1983). Set-up Times in Cyclic and Acyclic Group Technology Scheduling Systems, International Journal of Production Research, Vol. 21, No. 1, pp. 63-73.

Greene, T. J., and R. P. Sadowski, (1983). Cellular Manufacturing Control, Journal of Manufacturing Systems, Vol. 2, No. 2, pp. 137-144.

Greene, T. J., and R. P. Sadowski, (1984). A Review of Cellular Manufacturing Assumptions, Advantages and Design Techniques, Journal of Operations Management, Vol. 4, No. 2, pp. 85-97.

Ham, I., K. Hitomi, N. Nakamura, and T. Yoshida, (1979). Optimal Group Scheduling and Machining-Speed Decision under Due-Date Constraints, Journal of Engineering for Industry, Vol. 101, No. 5, pp. 128-134.

Hitomi, K., and I. Ham, (1976). Operations Scheduling for Group Technology Applications, Annals of the CIRP, Vol. 25, No. 1, pp. 419-422.

Hitomi, K., and I. Ham, (1977). Group Scheduling Technique for Multiproduct, Multistage Manufacturing Systems, Journal of Engineering for Industry, Vol. 99, No. 3, pp. 759-765.

Hitomi, K., and I. Ham, (1978). Machine Loading and Product-Mix Analysis for Group Technology, Journal of Engineering for Industry, Vol. 100, No. 3, pp. 370-374.

Hitomi, K., and I. Ham, (1982). Product-Mix and Machine-Loading Analysis of Multistage Production Systems for Group Technology, Journal of Engineering for Industry, Vol. 104, pp. 363-368.

Hyer, N. L., (1984). The Potential of Group Technology for U.S. Manufacturing, Journal of Operations Management, Vol. 4, No. 3, pp. 183-202.

Hyer, N. L., and U. Wemmerlov, (1982). MRP/GT: A Framework for_Production Planning and Control of Cellular Manufacturing, Decision Sciences, Vol. 13, No. 4, pp. 681-701.

Jacobs, F. R., and D. J. Bragg, (1988). Repetitive Lots: Flow-time Reductions Through Sequencing and Dynamic Batch Sizing, Decision Sciences, Vol. 19, No. 2, pp. 281-294.

Kanet, J. J., and D. P. Christy, (1984). Manufacturing Systems with Forbidden Early Order Departure, International Journal of Production Research, Vol. 22, No. 1, pp. 41-50.

Kanet, J. J., and D. P. Christy, (1989). Manufacturing Systems with Forbidden Early Shipments: Implications for Setting Manufacturing Lead Times, International Journal of Production Research, Vol. 27, No. 5, pp. 783-792.

Kelly, F. H., C. T. Mosier, and L. R. Taube, (1986). Impact of Cost Structure on Group Technology Implementation, Proceedings of New Developments and Integrative Opportunities in Information Systems and Operations Management, Greensboro, NC.

Kleijnen, J. P. C., (1987). Statistical Tools for Simulation Practitioners, Marcel Dekker Publishing, New York.

Law, A. M., and W. D. Kelton, (1991). Simulation Modeling and Analysis, McGraw-Hill Publishing, New York.

Lawrence, S. R., (1991). Scheduling a Single Machine to Maximize Net Present Value, International Journal of Production Research, Vol. 29, No. 6, pp. 1141-1160.

Lawrence, S. R., and T. E. Morton, (1986). PATRIARCH: Hierarchical Production Scheduling, Proceedings of Real-Time Optimization in Automated Manufacturing Facilities, NBS Special Publication 724, Gaithersburg, MD, pp. 87-97.

Lee, L. C., (1985). A Study of System Characteristics in a Manufacturing Cell, International Journal of Production Research, Vol. 23, No. 6, pp. 1101-1114.

Leu, B.-Y., T. J. Greene, and J. W. Nazemetz, (1992). A Review of Group Scheduling in Cellular Manufacturing, Working Paper, School of Industrial Engineering and Management, Oklahoma State University, December 1992.

Mahmoodi, F., (1989). An Investigation of Dynamic Group Scheduling and Releasing Heuristics in Cellular Manufacturing, Unpublished Ph.D. Dissertation, University of Minnesota.

Mahmoodi, F., and K. J. Dooley, (1991). A Comparison of Exhaustive and Non-exhaustive Group Scheduling Heuristics in a Manufacturing Cell, International Journal of Production Research, Vol. 29, No. 9, pp. 1923-1939.

Mahmoodi, F., and K. J. Dooley, (1992). Group Scheduling and Order Releasing: Review and Foundations for Research, Production Planning and Control, Vol. 3, No. 1, pp. 70-80.

Mahmoodi, F., K. J. Dooley, and P. J. Starr, (1988). A Comparison of Dynamic Scheduling Heuristics for Cellular Manufacturing, Proceedings of National Decision Sciences Institute, Las Vagas, NV, pp. 205-207.

Mahmoodi, F., K. J. Dooley, and P. J. Starr, (1990a). An Evaluation of Order Releasing and Due Date Assignment Heuristics in a Cellular Manufacturing System, Journal of Operations Management, Vol. 9, No. 4, pp. 548-573.

Mahmoodi, F., K. J. Dooley, and P. J. Starr, (1990b). An Investigation of Dynamic Group Scheduling Heuristics in a Job Shop Manufacturing Cell, International Journal of Production Research, Vol. 28, No. 9, pp. 1695-1711.

Mahmoodi, F., E. J. Tierney, and C. T. Mosier, (1992). Dynamic Group Scheduling Heuristics in a Flow-through Cell Environment, Decision Sciences, Vol. 23, No. 1, pp. 61-85.

Melnyk, S. A., P. L. Carter, D. M. Dilts, and D. M. Lyth, (1985). Shop Floor Control, Dow Jones-Irwin Publishing, Homewood, IL.

Melnyk, S. A., and G. L. Ragatz, (1989). Order Review/ Release: Research Issues and Perspectives, International Journal of Production Research, Vol. 27, No. 7, pp. 1081-1096.

Moily, J. P., and J. P. Stinson, (1989). Effects of Processing Time Variability and Routing Flexibility on Group Technology Implementation in Flow-Shop, Proceedings of Annual Meeting of the Decision Sciences Institute, New Orleans, LA, pp. 898-901.

Morris, J. S., and R. J. Tersine, (1990). A Simulation Analysis of Factors Influencing the Attractiveness of Group Technology Cellular Layouts, Management Science, Vol. 36, No. 12, pp. 1567-1578.

Morton, T. E., S. R. Lawrence, S. Rajagopolan, and S. Kekre, (1988). SCHED-STAR A Price-Based Shop Scheduling Module, Journal of Manufacturing and Operations Management, Vol. 1, No. 2, pp. 131-181.

Mosier, C. T., (1983). The Job Shop using Group Technology: A Simulation Investigation, Unpublished Ph.D. Dissertation, University of North Carolina at Chapel Hill.

Mosier, C. T., D. A. Elvers, and D. Kelly, (1984). Analysis of Group Technology Scheduling Heuristics, International Journal of Production Research, Vol. 22, No. 5, pp. 857-875.

Mosier, C. T., and L. Taube, (1985). The Facets of Group Technology and their Impacts on Implementation—— A State-of-the-Art Survey, OMEGA, Vol. 13, No. 5, pp. 381-391.

Nawaz, M., E. E. Enscore, and I. Ham, (1983). A Heuristic Algorithm for the m-machine, n-job Flow Shop Sequencing Problem, OMEGA, Vol. 11, No. 1, pp. 91-95.

Nobuto, N, T. Yoshida, and K. Hitomi, (1978). Group Production Scheduling for Minimum Total Tardiness, AIIE Transactions, Vol. 10, No. 2, pp. 157-162.

Nyman, L. R. (editor), (1992). Making Manufacturing Cells Work, The Society of Manufacturing Engineers.

Oral, M., and J. L. Malouin, (1973). Evaluation of the Shortest Processing Time Scheduling Rule with Truncation Process, AIIE Transactions, Vol. 5, No. 4, pp. 357-363.

Petrov, V. A., (1968). Flowline Group Production Planning, English Translation from Russian by E. Bishop, Business Publications, Ltd., London.

Pritsker, A. A. B., (1986). Introduction to Simulation and SLAM II, John Wiley and Sons Publishing, New York.

Pritsker, A. A. B., C. E. Sigal, and R. D. J. Hammesfahr, (1989). SLAM II Network Models for Decision Support, Prentice-Hall Publishing, Englewood Cliffs, NJ.

Radharamanan, R., (1986). A Heuristic Algorithm for Group Scheduling, Proceedings of the 1986 IIE Engineering Conference, Dallas, TX, pp. 229-236.

Ragatz, G. L., (1985). An Evaluation of Order Release Mechanisms for Job Shops, Unpublished Ph.D. Dissertation, Indiana University.

Ragatz, G. L., and J. R. Carter, (1988). Scheduling Against Due Dates in a Sequence Dependent Setup Environment, Working Paper, Michigan State University, East Lansing, MI, 1988.

Rajagopal, K., and K. Rathmill, (1980). The Performance Analysis of Batch Production Cells using Mathematical Modelling, Proceedings of the 21st International Machine Tool Design and Research Conference, MacMillan Press, Swansea, pp. 549-557.

Rohleder, T. R., and G. D. Scudder, (1992). Scheduling Rule Selection for Forbidden Early Shipment Environment: A Comparison of Economic Objectives, International Journal of Production Research, Vol. 30, No. 1, pp. 129-140.

Ruben, R. A., C. T. Mosier, and F. Mahmoodi, (1993). A Comprehensive Analysis of Group Scheduling Heuristics in a Job Shop Cell, International Journal of Production Research, (forthcoming).

Russell, G. R., and P. R. Philipoom, (1991). Sequencing Rules and Due Date Setting Procedures in Flow Line Cells with Family Setups, Journal of Operations Management, Vol. 10, No. 4, pp. 524-545.

SAS User's Guide: Basics, Version 5 Edition, (1985). SAS Institute Inc., Cary, NC.

SAS/STAT User's Guide, Release 6.03 Edition, (1988). SAS Institute Inc., Cary, NC.

Sassani, F., (1985). Computer Simulation of Scheduling and Service Rules in Group Technology Cells, Proceedings of the 13th International Conference on Modeling and Simulation, Lugano, pp. 256-259.

Sassani, F., (1990). A Simulation Study on Performance Improvement of Group Technology Cells, International Journal of Production Research, Vol. 28, No. 2, pp. 293-300.

Sawicki, J. D., (1973). The Problems of Tardiness and Saturation in a Multi-Class Queue with Sequence Dependent Setups, AIIE Transactions, Vol. 5, No. 4, pp. 250-255.

Scudder, G. D., and T. R. Hoffmann, (1987). The Use of Cost-Based Priorities in Random and Flow Shops, Journal of Operations Management, Vol. 7, Nos. 1 and 2, pp. 526-547.

Scudder, G. D., and T. R. Hoffmann, (1989). The Use of Value-Based Priorities When Early Shipments Are Forbidden, International Journal of Production Research, Vol. 27, No. 2, pp. 353-369.

Scudder, G. D., and D. E. Smith-Daniels, (1989). Application of the Net Present Value Criterion in Random and Flow Shop Scheduling, Decision Sciences, Vol. 20, No. 3, pp. 602-622.

Scudder, G. D., D. E. Smith-Daniels, and T. Rohleder, (1990). Use of the Net Present Criterion in a Random Job Shop Where Early Shipments Are Forbidden, Journal of Operations Management, Vol. 9, No. 4, pp. 526-547.

Sinha, R. K., and R. H. Hollier, (1984). A Review of Production Control Problems in Cellular Manufacturing, International Journal of Production Research, Vol. 22, No. 5, pp. 773-789.

Steel, R. G., and J. H. Torrie, (1980). Principles and Procedures of Statistics, McGraw-Hill Publishing, New York.

Sundaram, R. M., (1982). A Group Scheduling Algorithm for that Utilize Group Technology Manufacturing Concept, Proceedings of the 10th North American Manufacturing Research Conference, Ontario, Canada, pp. 524-528.

Sundaram, R. M., (1983). Some Scheduling for Group Technology Manufacturing System, in Computer Applications in Production and Engineering, Edited by E. A. Warman, North-Holland Publishing, Amsterdam, pp. 765-772.

Vaithiananthan, R., and K. L. McRoberts, (1982). On Scheduling in a GT Environment, Journal of Manufacturing Systems, Vol. 1, No. 2, pp. 149-155.

Vakharia, A. J., and Y.-L. Chang, (1990). A Simulated Annealing Approach to Scheduling a Manufacturing Cell, Naval Research Logistics, Vol. 37, No. 4, pp. 559-577.

Vepsalainen, A. P., and T. E. Morton, (1987). Priority Rules and Leadtime Estimation for Job Shops Scheduling with Weighted Tardiness Costs, Management Science, Vol. 33, No. 8, pp. 1036-1047.

Wemmerlov, U., (1989). Part Family Scheduling on a Single Machine, Proceedings of Annual Meeting of the Decision Sciences Institute, New Orleans, LA, pp. 197-199.

Wemmerlov, U., and N. L. Hyer, (1987). Research Issues in Cellular Manufacturing, International Journal of Production Research, Vol. 25, No. 3, pp. 413-431.

Wemmerlov, U., and N. L. Hyer, (1989). Cellular Manufacturing in the U.S. industry: a Survey of Users, International Journal of Production Research, Vol. 27, No. 9, pp. 1511-1530.

Wemmerlov, U., and A. J. Vakharia, (1991). Job and Family Scheduling of a Flow-line Manufacturing Cell: A Simulation Study, IIE Transactions, Vol. 23, No. 4, pp. 383-393.

Willey, P. C. T., and C. L. Ang, (1980). Computer Simulation of the Effects of Inter-Cell Workload Transfer on the Performance of GT Systems, Proceedings of the 21st International Machine Tool Design and Research Conference, Swansea, pp. 559-567.

APPENDIXES

APPENDIX A

SYSTEM DESCRIPTION PROGRAMS AND DATA

SAS PROGRAMS FOR GENERATING SYSTEM DATA

```
/*
   GENERATE ANNUAL DEMANDS FOR ALL PART TYPES
*/
DATA DEMAND;
   RETAIN SEED1 1613518295;
   DO I=1 to 30;
      DEM=ROUND(100+1401*RANUNI(SEED1));
   OUTPUT; END;
PROC PRINT DATA=DEMAND;
   VAR DEM;
   TITLE 'GENERATE ANNUAL DEMANDS FROM U(100,1500)';
PROC MEANS PRINT DATA=DEMAND;
   VAR DEM;
RUN;
;
;
/*
   GENERATE ANNUAL ORDERS FOR ALL PART TYPES
   -- FOR HIGH DEMAND PATTERN VARIABILITY
*/
DATA HIGH;
   RETAIN SEED1 1672548295;
   DO I=1 to 30;
      AOH=ROUND(10+21*RANUNI(SEED1));
   OUTPUT; END;
PROC PRINT DATA=HIGH;
   VAR AOH;
   TITLE 'GENERATE ANNUAL ORDERS FROM U(10,30)';
PROC MEANS PRINT DATA=HIGH;
   VAR AOH;
RUN;
;
;
/*
   GENERATE ANNUAL ORDERS FOR ALL PART TYPES
   -- FOR LOW DEMAND PATTERN VARIABILITY
*/
DATA LOW;
   RETAIN SEED1 1613578295;
   DO I=1 to 30;
      AOL=ROUND(30+21*RANUNI(SEED1));
   OUTPUT; END;
PROC PRINT DATA=LOW;
   VAR AOL;
   TITLE 'GENERATE ANNUAL ORDERS FROM U(30,50)';
PROC MEANS PRINT DATA=LOW;
   VAR AOL;
RUN;
;
;
```

```
/*
   GENERATE ANNUAL WORKLOADS FOR ALL WORKCENTERS
*/
DATA WORKLOAD;
   RETAIN SEED1 1463958721;
   DO I=1 to 9;
*** HIGH AAW (90%)
      AAPW=ROUND((1620+81.0*RANNOR(SEED1)));
      AASW=ROUND((180+9.0*RANNOR(SEED1)));
*** LOW AAW (80%)
*     AAPW=ROUND((1440+72.0*RANNOR(SEED1)));
*     AASW=ROUND((160+8.0*RANNOR(SEED1)));
      AAW=AAPW+AASW;
   OUTPUT; END;
PROC PRINT DATA=WORKLOAD;
   VAR AASW AAPW AAW;
   TITLE 'GENERATE ANNUAL WORKLOADS';
PROC MEANS PRINT DATA=WORKLOAD;
   VAR AASW AAPW AAW;
RUN;
;
;
/*
   GENERATE ROUTINGS FOR ALL PART TYPES
*/
DATA ROUTE;
   RETAIN SEED1 1673547295;
   DO I=1 to 30;
      ROUT=ROUND(3+3*RANUNI(SEED1));
   OUTPUT; END;
PROC PRINT DATA=ROUTE;
   VAR ROUT;
   TITLE 'GENERATE ROUTING TABLE FROM U(3,5)';
PROC MEANS PRINT DATA=ROUTE;
   VAR ROUT;
RUN;
;
;
```

TABLE OF DEMAND DEMANDS, ANNUAL ORDERS, AND AVERAGE ORDER SIZES

		High Demand Pattern Variability		Low Demand Pattern Variability	
Part Type	Annual Demand U(100,1500)	Annual Orders U(10,30)	Average Order Size	Annual Orders U(30,50)	Average Order Size
1	1135	22	51.59	31	36.61
2	548	11	49.82	46	11.91
3	118	17	6.94	32	3.69
4	1257	24	52.38	49	25.65
5	358	15	23.87	35	10.23
6	767	15	51.13	38	20.18
7	196	26	7.54	39	5.03
8	1025	18	56.94	42	24.40
9	248	19	13.05	49	5.06
10	1281	29	44.17	47	27.26
11	185	15	12.33	33	5.61
12	942	20	47.10	31	30.39
13	1043	19	54.89	47	22.19
14	100	10	10.00	48	2.08
15	1223	28	43.68	50	24.46
16	228	17	13.41	39	5.85
17	969	19	51.00	41	23.63
18	1156	27	42.81	46	25.13
19	899	25	35.96	36	24.97
20	280	10	28.00	42	6.67
21	336	29	11.59	34	9.88
22	675	16	42.19	37	18.24
23	937	28	33.46	43	21.79
24	1053	19	55.42	47	22.40
25	456	23	19.83	42	10.86
26	659	15	43.93	42	15.69
27	1173	26	45.12	39	30.08
28	687	30	22.90	33	20.82
29	475	23	20.65	31	15.32
30	517	24	21.54	38	13.61
Average	697.53	20.63	33.78	40.23	17.32

TABLE OF AVERAGE ANNUAL SETUP AND PROCESSING WORKLOADS, AND AVERAGE ANNUAL WORKLOADS

(1) Average annual workload across all workcenters is, on average, 90% of total capacity of the system.

Workcenter	Average Annual Setup Workload	Average Annual Processing Workload	Average Annual Workload
A1	185	1481	1666
A2	175	1632	1807
A3	173	1619	1792
A4	185	1796	1981
B1	174	1657	1831
B2	188	1608	1796
B3	197	1596	1793
B4	176	1635	1811
A5/B5	183	1704	1887
Average	181.78	1636.44	1818.22

(2) Average annual workload across all workcenters is, on average, 80% of total capacity of the system.

Workcenter	Average Annual Setup Workload	Average Annual Processing Workload	Average Annual Workload
A1	164	1317	1481
A2	155	1451	1606
A3	154	1439	1593
A4	164	1597	1761
B1	155	1473	1628
B2	167	1429	1596
B3	175	1419	1594
B4	156	1454	1610
A5/B5	163	1515	1678
Average	161.44	1454.89	1616.33

TABLE OF ROUTINGS

Part Type	Number of Operations	Routing
1	4	A2 -> A3 -> A4 -> A5
2	5	A1 -> A2 -> A3 -> A4 -> A5
3	3	A1 -> A2 -> A3
4	4	A1 -> A2 -> A4 -> A5
5	4	A1 -> A2 -> A3 -> A5
6	5	A1 -> A2 -> A3 -> A4 -> A5
7	4	A1 -> A3 -> A4 -> A5
8	4	A2 -> A3 -> A4 -> A5
9	3	A1 -> A2 -> A4
10	5	A1 -> A2 -> A3 -> A4 -> A5
11	4	A1 -> A3 -> A4 -> A5
12	3	A2 -> A3 -> A5
13	5	A1 -> A2 -> A3 -> A4 -> A5
14	3	A1 -> A3 -> A4
15	4	A1 -> A2 -> A4 -> A5
16	3	B1 -> B3 -> B5
17	3	B2 -> B4 -> B5
18	5	B1 -> B2 -> B3 -> B4 -> B5
19	3	B1 -> B4 -> B5
20	4	B1 -> B2 -> B3 -> B4
21	4	B1 -> B2 -> B3 -> B5
22	5	B1 -> B2 -> B3 -> B4 -> B5
23	3	B1 -> B2 -> B4
24	5	B1 -> B2 -> B3 -> B4 -> B5
25	5	B1 -> B2 -> B3 -> B4 -> B5
26	3	B2 -> B3 -> B4
27	5	B1 -> B2 -> B3 -> B4 -> B5
28	3	B3 -> B4 -> B5
29	5	B1 -> B2 -> B3 -> B4 -> B5
30	4	B1 -> B2 -> B3 -> B5

APPENDIX B

SAS PROGRAM AND SAMPLE SAS OUTPUT OF ANALYSIS OF VARIANCE

SAS PROGRAM OF ANALYSIS OF VARIANCE

```
/*
       SAS PROGRAM OF ANOVA FOR SIMULATION OUTPUTS

       4 FACTORS => G: Group Scheduling Heuristic
                    A: Average Annual Workload
                    D: Demand Pattern Variability
                    C: Cell Transfer Batch
       10 PERFORMANCE MEASURES => Y1 - Y10
*/
OPTIONS PS=60 LS=80 NODATE;
DATA DISS;
   INFILE 'DISS.IN';
   INPUT A D C G REP Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10;
;
PROC ANOVA DATA=DISS;
   CLASS G A D C;
   MODEL Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10=
            G A D C
            G*A G*D G*C A*D A*C D*C
            G*A*D G*A*C G*D*C A*D*C
            G*A*D*C;
RUN;
```

SAMPLE SAS OUTPUT OF ANALYSIS OF VARIANCE

```
                 Class Level Information

                 Class    Levels    Values
                 G             5    1 2 3 4 5
                 A             2    1 2
                 D             2    1 2
                 C             2    1 2
         Number of observations in data set = 1000

               Analysis of Variance Procedure

Dependent Variable: Y1 (Time in System)

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

Model         39   2027627044.5      51990437.0    75.01   0.0001

Error        960    665413640.3        693139.2

Corrected
  Total      999   2693040684.8

    R-Square          C.V.          Root MSE           Y1 Mean
    0.752914      11.69971         832.54982         7115.9860

               Analysis of Variance Procedure

Dependent Variable: Y1 (Time in System)

Source        DF       Anova SS     Mean Square  F Value   Pr > F

G              4    130823727.7      32705931.9    47.19   0.0001
A              1   1186773709.5    1186773709.5  1712.17   0.0001
D              1    577490778.0     577490778.0   833.15   0.0001
C              1      3713103.5       3713103.5     5.36   0.0209
G*A            4     61647904.6      15411976.2    22.24   0.0001
G*D            4     11205493.8       2801373.4     4.04   0.0030
G*C            4      6043042.3       1510760.6     2.18   0.0694
A*D            1          192.2           192.2     0.00   0.9867
A*C            1      5592072.3       5592072.3     8.07   0.0046
D*C            1     15336687.4      15336687.4    22.13   0.0001
G*A*D          4     10821283.8       2705320.9     3.90   0.0038
G*A*C          4      2815230.3        703807.6     1.02   0.3983
G*D*C          4      2603866.1        650966.5     0.94   0.4404
A*D*C          1     12146151.5      12146151.5    17.52   0.0001
G*A*D*C        4       613801.4        153450.3     0.22   0.9266
```

APPENDIX C

EXAMPLES FOR GROUP SCHEDULING HEURISTICS

AN EXAMPLE FOR ADD/EDD, ACR/CR, AND ASLK/SLK

This example is designed to illustrate the mechanisms of the first three heuristics (i.e., ADD/EDD, ACR/CR, and ASLK/SLK). Assume that at time 30, there are 7, 3, and 5 jobs in the queues 1, 2, and 3 in front of the workcenter A3 (see Figure 4.2), respectively. The jobs' data are given below:

	Queue 1				Queue 2				Queue 3			
Job #	D_i	P_i	CR_i	S_i	D_i	P_i	CR_i	S_i	D_i	P_i	CR_i	S_i
1	94	51	1.25	13	67	36	1.03	1	84	37	1.46	17
2	82	72	.72	-20	83	69	.77	-16	56	60	.43	-34
3	59	27	1.07	2	90	32	1.88	28	89	29	2.03	30
4	83	60	.88	-7					91	58	1.05	3
5	91	45	1.36	16					61	27	1.15	4
6	75	35	1.29	10								
7	58	46	.61	-18								

Where:
D_i = Due date of job i
P_i = Remaining processing time of job i
CR_i = Critical ratio of job i
S_i = Slack of job i

Then, ADD, ACR, and ASLK for each queue can be calculated as shown below:

ADD for queue 1 = (58+59+75+82+83)/5 = 71.4 (minimum)
ADD for queue 2 = (67+83+90)/3 = 80.0
ADD for queue 3 = (56+61+84+89+91)/5 = 76.2

ACR for queue 1 = (.61+.72+.88+1.07+1.25)/5 = .906 (minimum)
ACR for queue 2 = (.77+1.03+1.88)/3 = 1.227
ACR for queue 3 = (.43+1.05+1.15+1.46+2.03)/5 = 1.224

ASLK for queue 1 = (-20-18-7+2+10)/5 = -6.60 (minimum)
ASLK for queue 2 = (-16+1+28)/3 = 4.33
ASLK for queue 3 = (-34+3+4+17+30)/5 = 4.00

(1) If we use ADD/EDD heuristic, we should select queue 1 and process job 7 within the queue 1 first, followed by jobs 3, 6, 2, and 4 within the queue 1.

(2) If we use ACR/CR heuristic, we should select queue 1 and process job 7 within the queue 1 first, followed by jobs 2, 4, 3, and 1 within the queue 1.

(3) If we use ASLK/SLK heuristic, we should select queue 1 and process job 2 within the queue 1 first, followed by jobs 7, 4, 3, and 6 within the queue 1.

AN EXAMPLE FOR NEQA/EQ AND NLQB/CR

This example is designed to illustrate the mechanisms of the last two heuristics (i.e., NEQA/EQ and NLQB/CR). Assume that the numbers of jobs waiting in the queues in front of workcenters in the cell A are given below:

Workcenter (WC) #	Queue Length		
	Queue 1	Queue 2	Queue 3
A1	6	0	4
A2	8	5	6
-> A3	7	3	5
A4	6	0	4
A5	0	0	7

Now, if we are going to schedule jobs at the workcenter A3, the number of empty queues and number of lengthy queues can be calculated as shown below:

\# of empty queues for queue 1 in front of workcenters A4 and A5 = 1
\# of empty queues for queue 2 in front of workcenters A4 and A5 = 2 (maximum)
\# of empty queues for queue 3 in front of workcenters A4 and A5 = 0

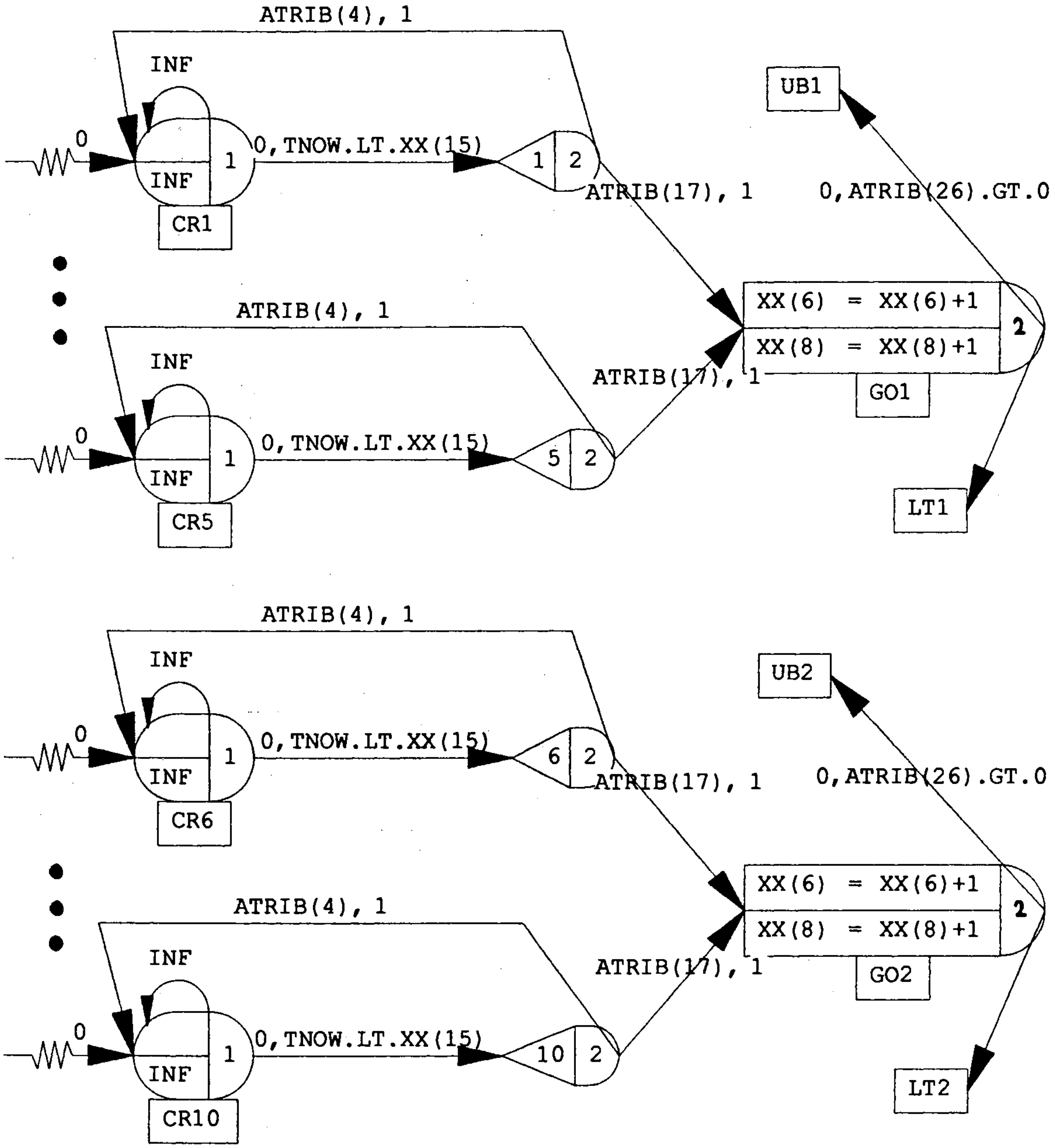
\# of lengthy queues for queue 1 in front of workcenters A1 and A2 = 2 (maximum)
\# of lengthy queues for queue 2 in front of workcenters A1 and A2 = 1
\# of lengthy queues for queue 3 in front of workcenters A1 and A2 = 1

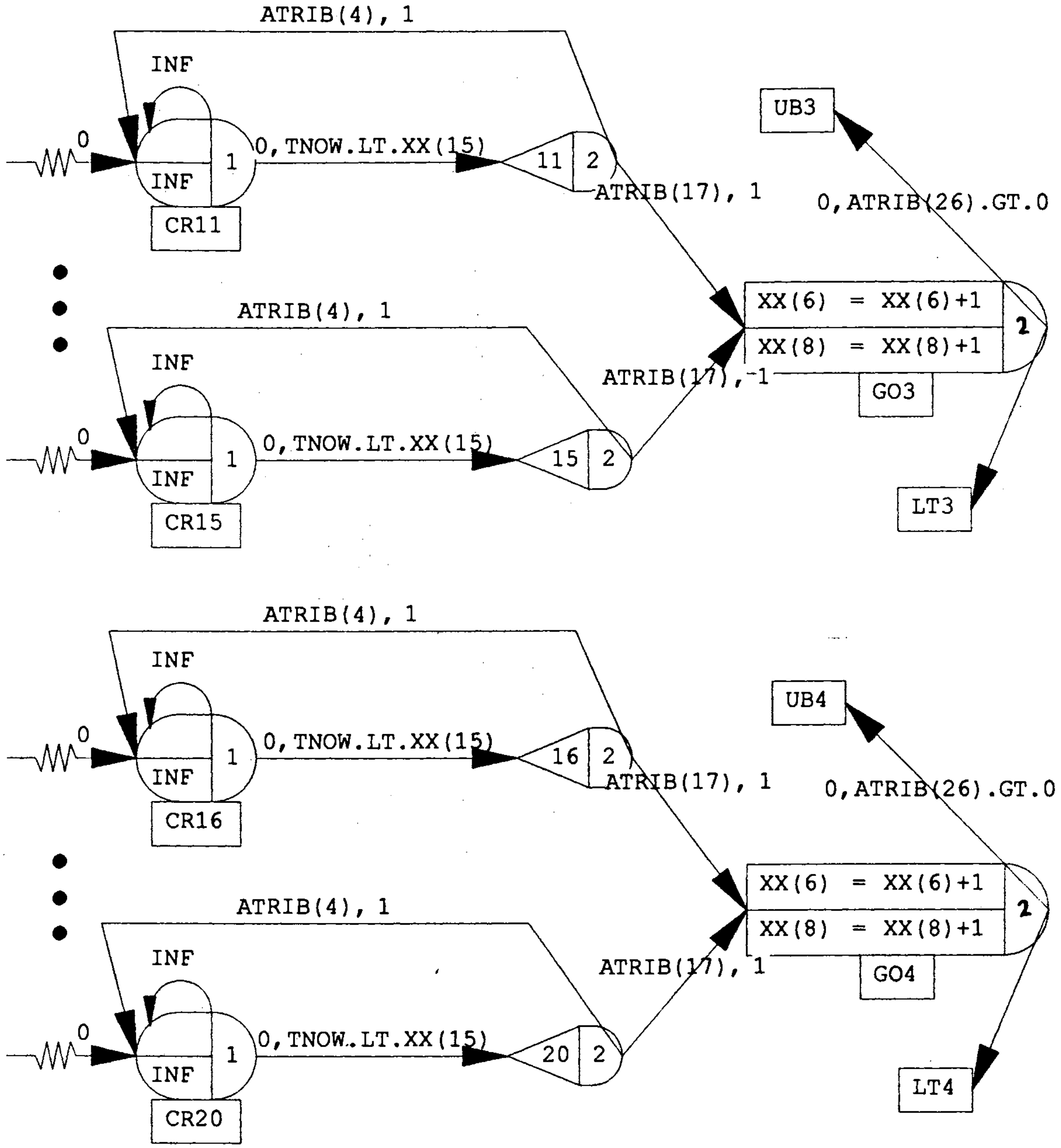
(1) If we use NEQA/EQ heuristic, we should select queue 2 and process all three jobs within the queue 2 which will go to the empty queues ahead (i.e., queue 2 in front of workcenters A4 and A5).

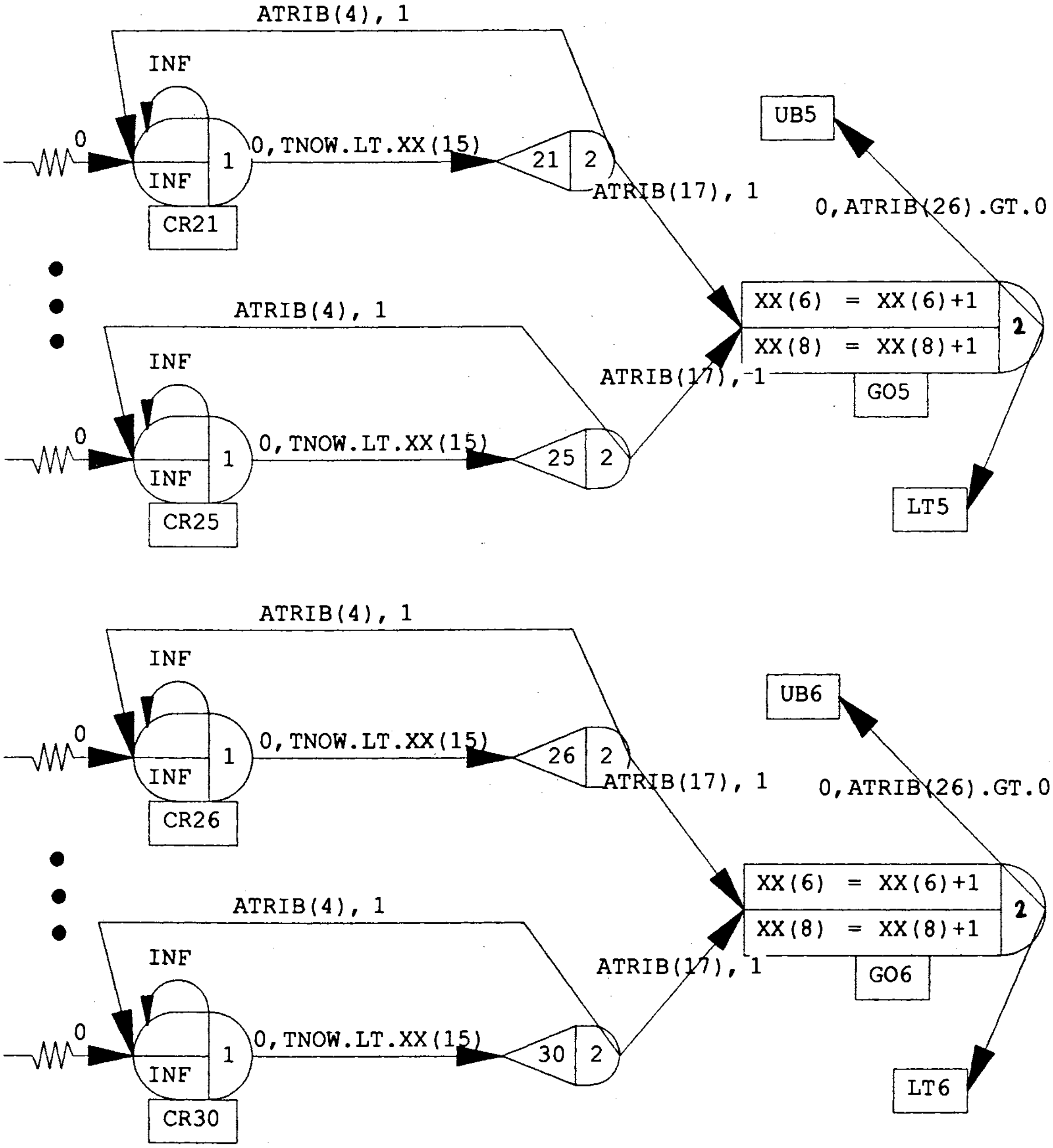
(2) If we use NLQB/CR heuristic, we should select queue 1 and process the five jobs within the queue 1 with the smallest critical ratios.

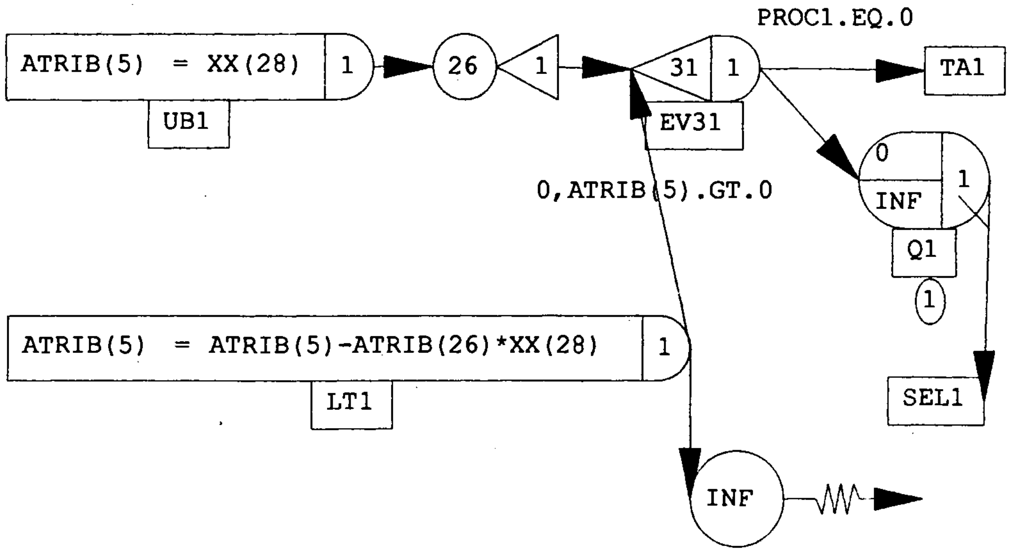
APPENDIX D

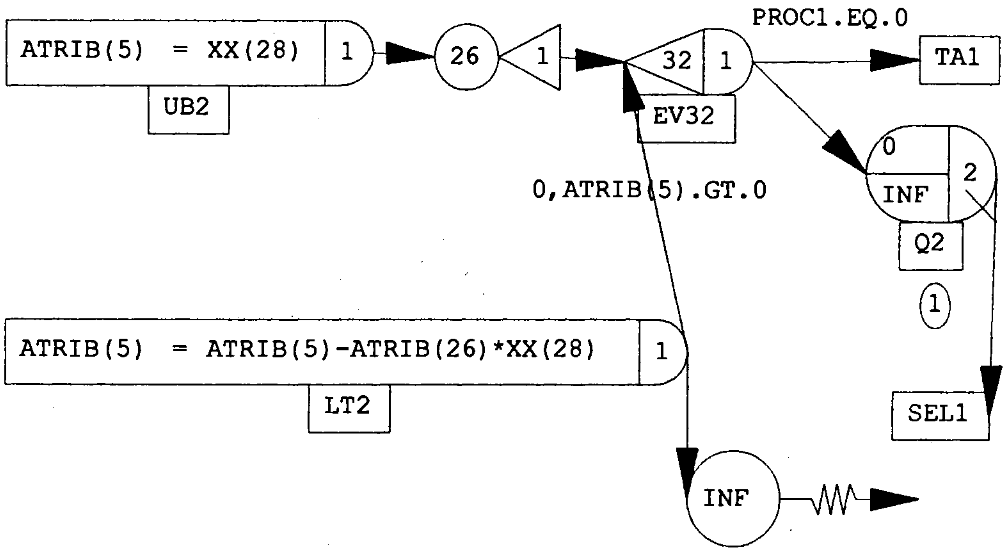
SLAM II NETWORK MODEL

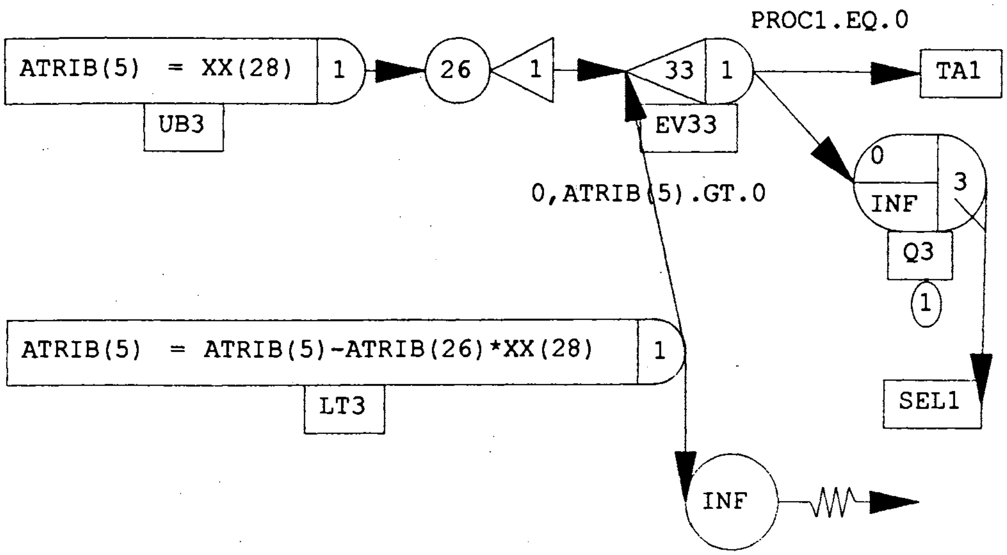

ATRIB(4), 1
INF
0
INF
1
CR1
0,TNOW.LT.XX(15)
1
2
ATRIB(17), 1
UB1
0,ATRIB(26).GT.0
XX(6) = XX(6)+1
XX(8) = XX(8)+1
2
GO1
LT1
ATRIB(4), 1
INF
0
INF
1
CR5
0,TNOW.LT.XX(15)
5
2
ATRIB(17), 1
ATRIB(4), 1
INF
0
INF
1
CR6
0,TNOW.LT.XX(15)
6
2
ATRIB(17), 1
UB2
0,ATRIB(26).GT.0
XX(6) = XX(6)+1
XX(8) = XX(8)+1
2
GO2
LT2
ATRIB(4), 1
INF
0
INF
1
CR10
0,TNOW.LT.XX(15)
10
2
ATRIB(17), 1

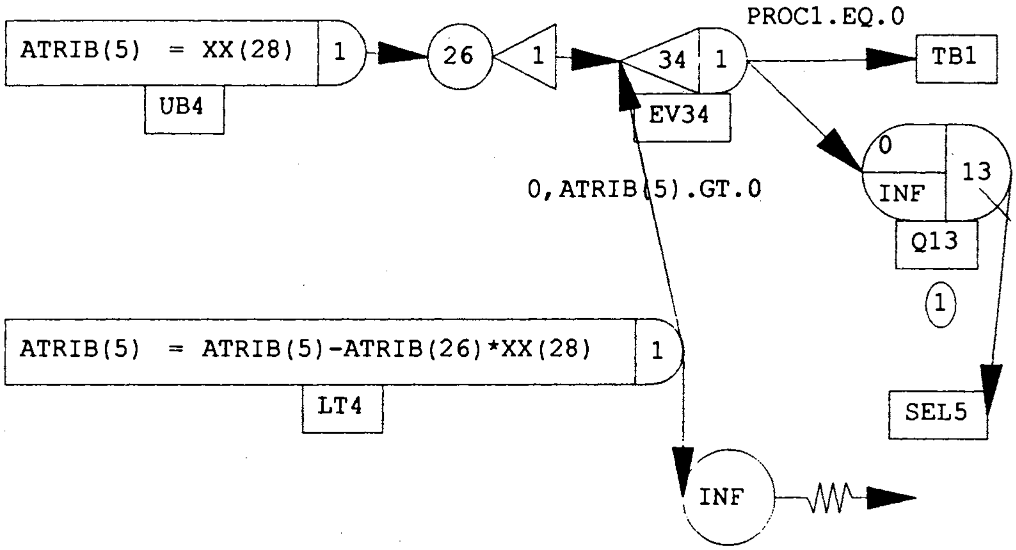

ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
11
2
CR11
ATRIB(17), 1
UB3
0,ATRIB(26).GT.0
XX(6) = XX(6)+1
XX(8) = XX(8)+1
2
GO3
ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
15
2
CR15
ATRIB(17), 1
LT3
ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
16
2
CR16
ATRIB(17), 1
UB4
0,ATRIB(26).GT.0
XX(6) = XX(6)+1
XX(8) = XX(8)+1
2
GO4
ATRIB(4), 1
INF
0
INF
1
0,TNOW..LT.XX(15)
20
2
CR20
ATRIB(17), 1
LT4

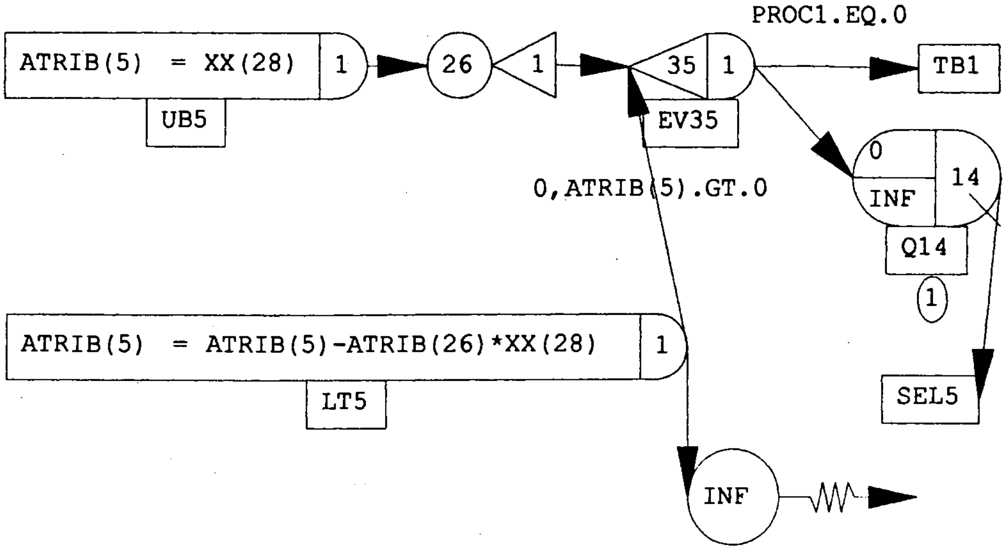

ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
21
2
CR21
ATRIB(17), 1
UB5
0,ATRIB(26).GT.0
XX(6) = XX(6)+1
XX(8) = XX(8)+1
2
GO5
ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
25
2
CR25
ATRIB(17), 1
LT5
ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
26
2
CR26
ATRIB(17), 1
UB6
0,ATRIB(26).GT.0
XX(6) = XX(6)+1
XX(8) = XX(8)+1
2
GO6
ATRIB(4), 1
INF
0
INF
1
0,TNOW.LT.XX(15)
30
2
CR30
ATRIB(17), 1
LT6

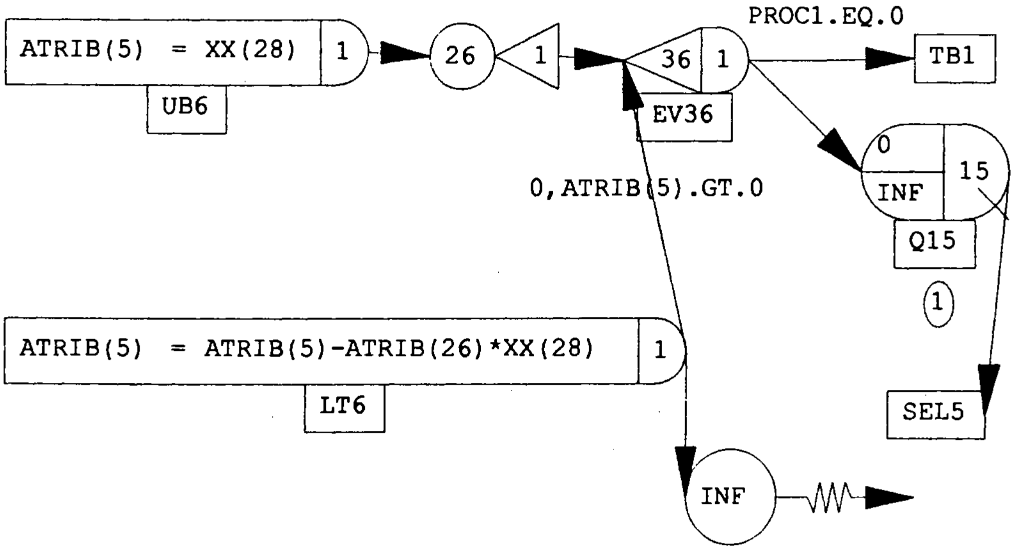

ATRIB(5) = XX(28)
1
UB1
26
1
31
1
EV31
PROC1.EQ.0
TA1
0
INF
1
Q1
1
0,ATRIB(5).GT.0
ATRIB(5) = ATRIB(5)-ATRIB(26)*XX(28)
1
LT1
SEL1
INF

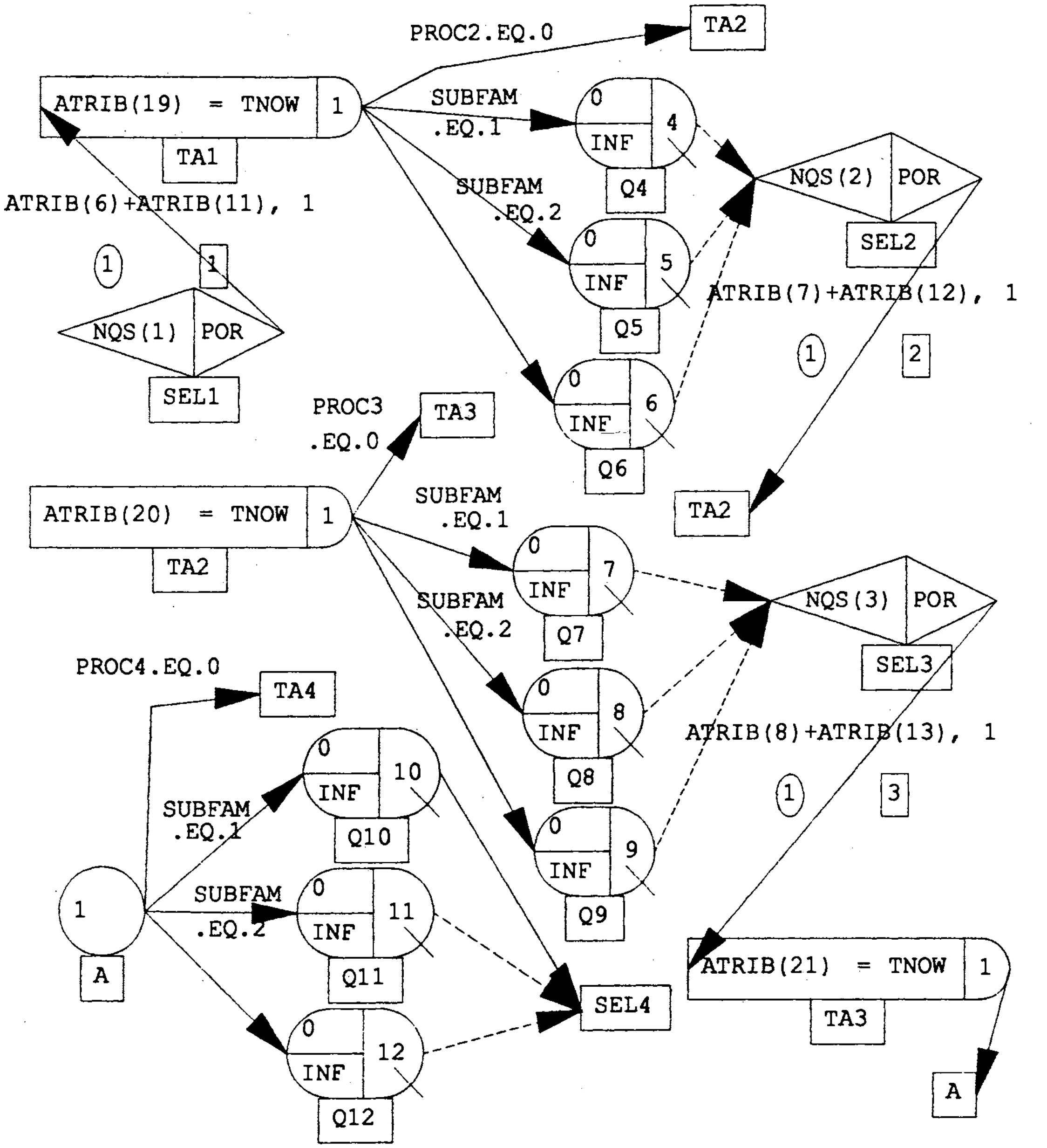

ATRIB(5) = XX(28)
1
UB2
26
1
32
1
EV32
PROC1.EQ.0
TA1
0
INF
2
Q2
1
0,ATRIB(5).GT.0
ATRIB(5) = ATRIB(5)-ATRIB(26)*XX(28)
1
LT2
SEL1
INF

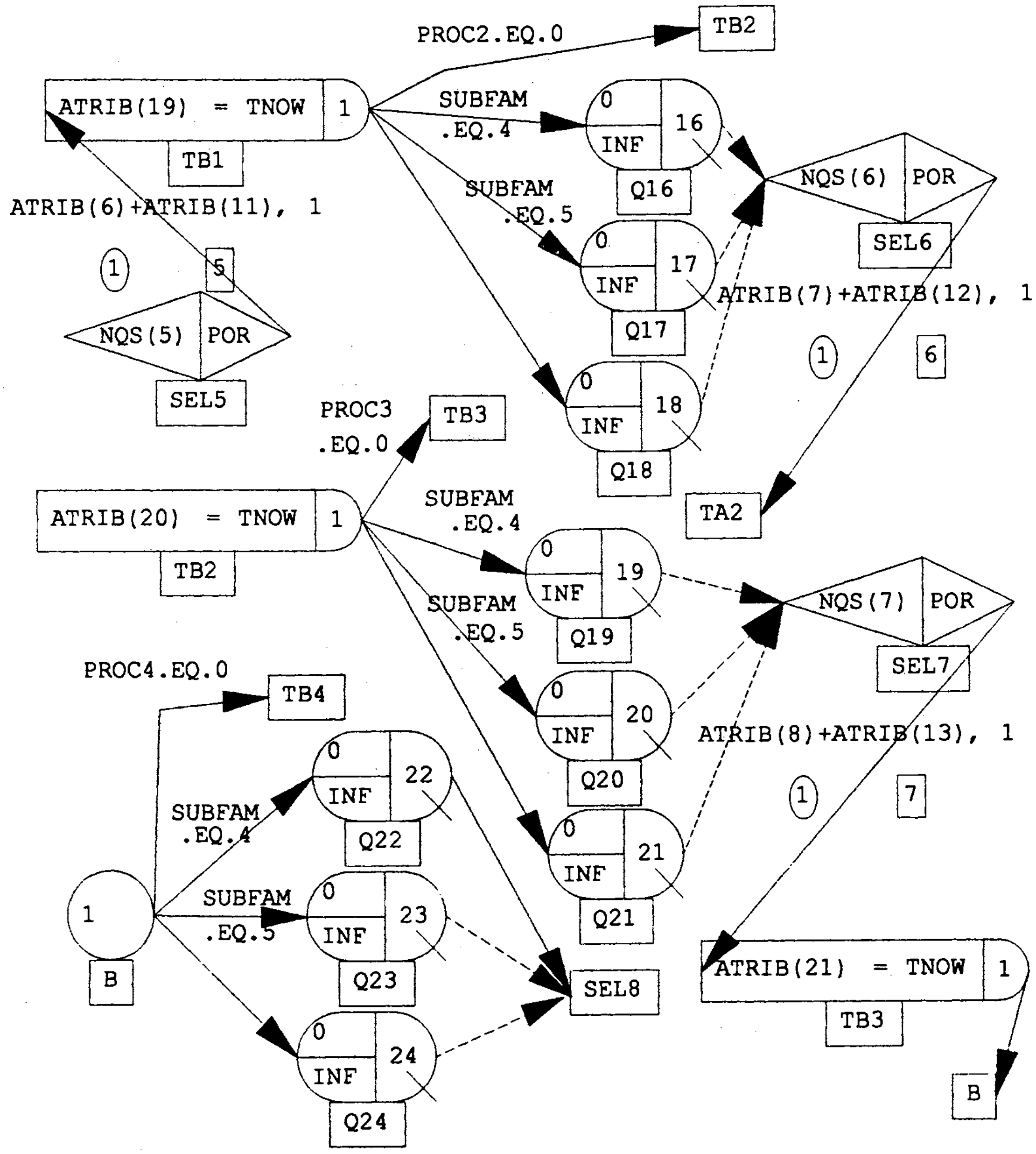

ATRIB(5) = XX(28)
1
UB3
26
1
33
1
EV33
PROC1.EQ.0
TA1
0
INF
3
Q3
1
SEL1
0,ATRIB(5).GT.0
ATRIB(5) = ATRIB(5)-ATRIB(26)*XX(28)
1
LT3
INF

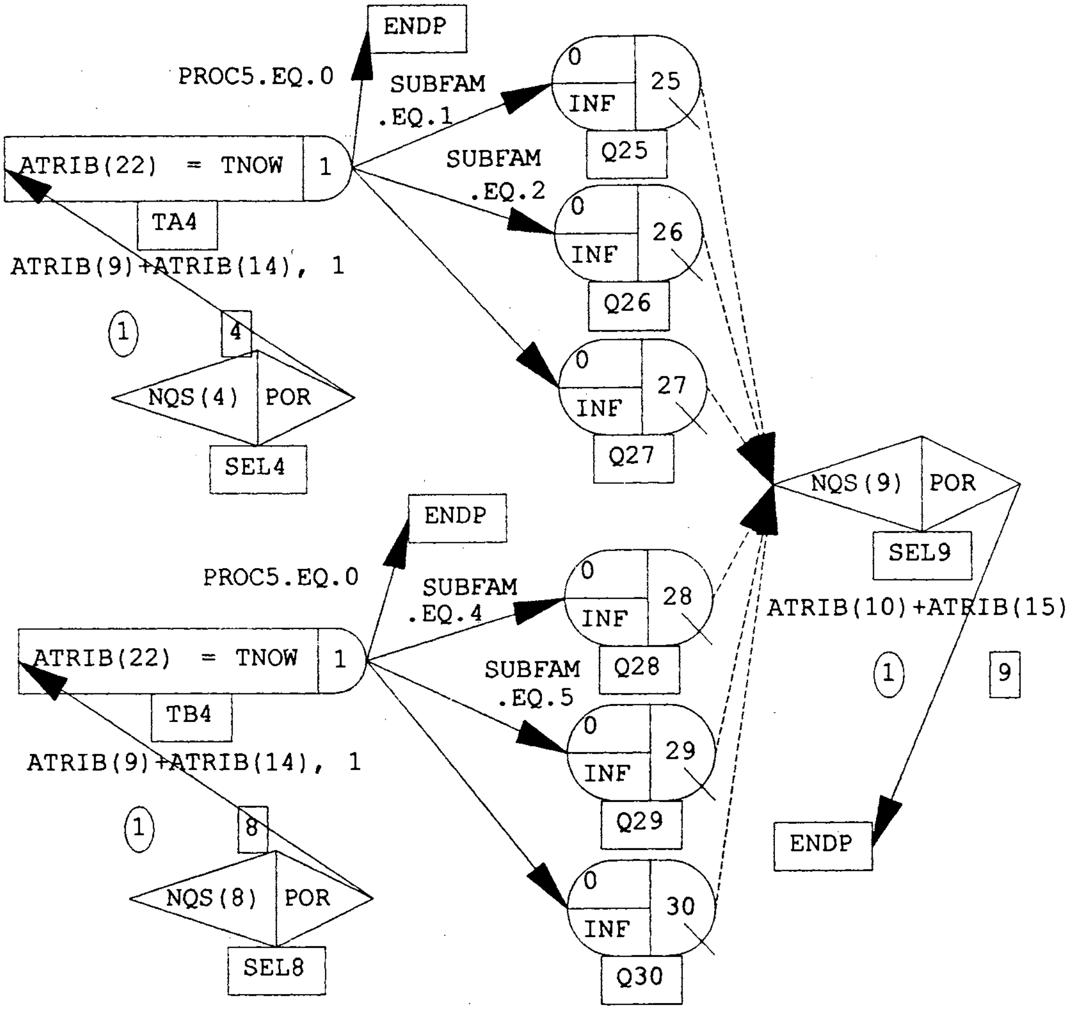

ATRIB(5) = XX(28)
1
UB4
26
1
34
1
EV34
PROC1.EQ.0
TB1
0
INF
13
Q13
1
SEL5
0,ATRIB(5).GT.0
ATRIB(5) = ATRIB(5)-ATRIB(26)*XX(28)
1
LT4
INF

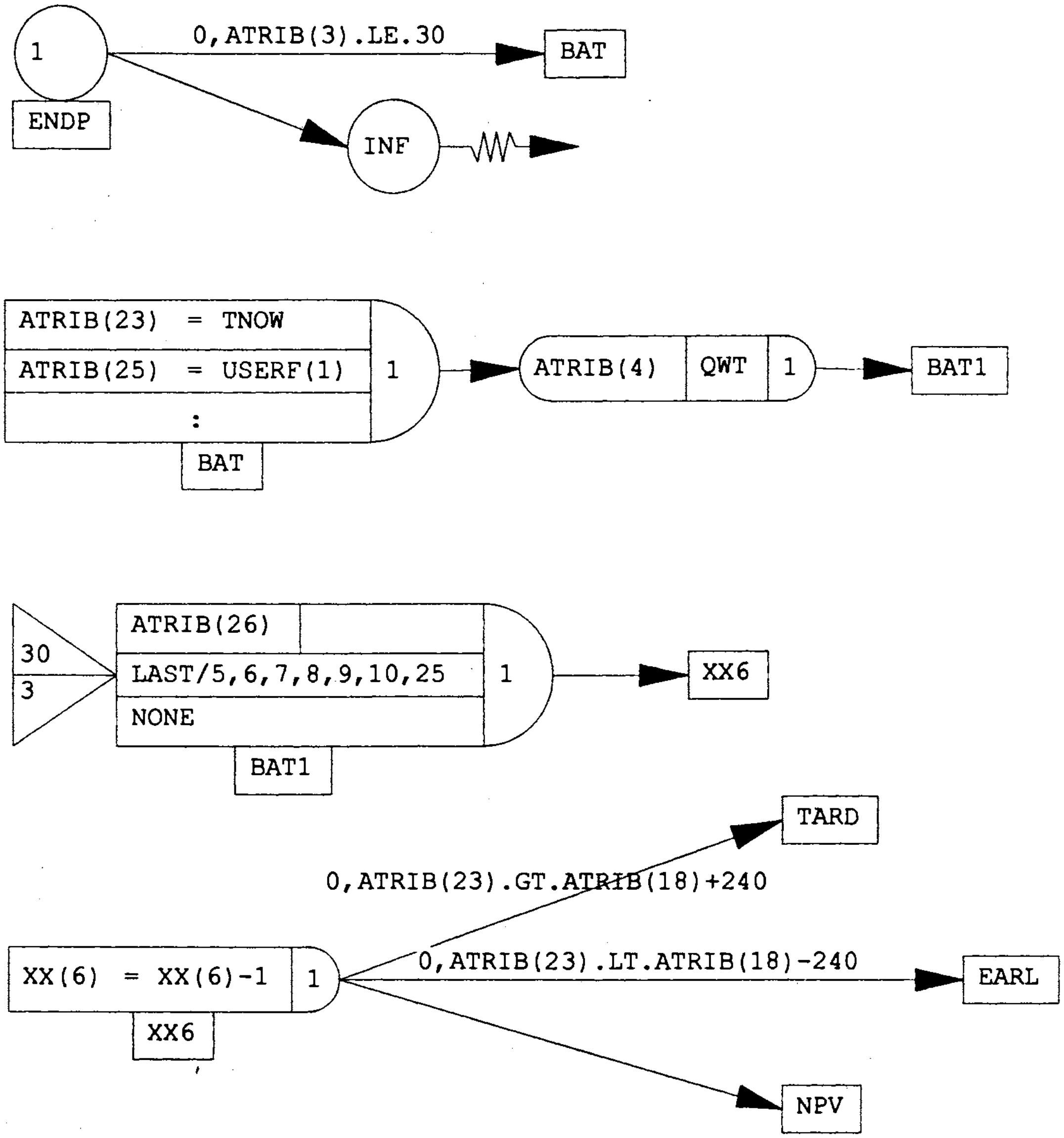

ATRIB(5) = XX(28)
1
UB5
26
1
35
1
EV35
PROC1.EQ.0
TB1
0
INF
14
Q14
1
0,ATRIB(5).GT.0
ATRIB(5) = ATRIB(5)-ATRIB(26)*XX(28)
1
LT5
SEL5
INF

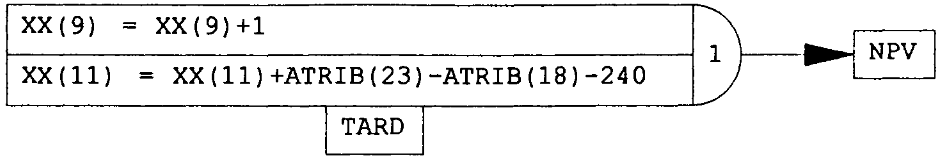

ATRIB(5) = XX(28)
1
UB6
26
1
36
1
EV36
PROC1.EQ.0
TB1
0
INF
15
Q15
1
0,ATRIB(5).GT.0
ATRIB(5) = ATRIB(5)-ATRIB(26)*XX(28)
1
LT6
SEL5
INF

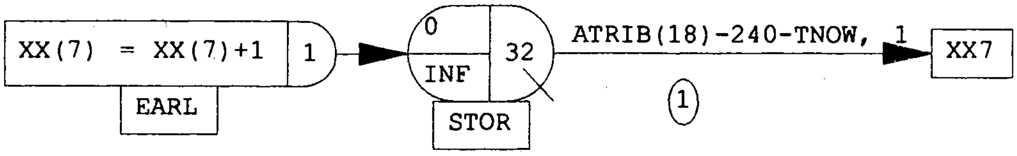

PROC2.EQ.0
TA2
ATRIB(19) = TNOW
1
TA1
SUBFAM
.EQ.1
0
INF
4
Q4
SUBFAM
.EQ.2
0
INF
5
Q5
NQS(2)
POR
SEL2
ATRIB(6)+ATRIB(11), 1
①
1
NQS(1)
POR
SEL1
ATRIB(7)+ATRIB(12), 1
①
2
0
INF
6
Q6
PROC3
.EQ.0
TA3
TA2
ATRIB(20) = TNOW
1
TA2
SUBFAM
.EQ.1
0
INF
7
Q7
SUBFAM
.EQ.2
NQS(3)
POR
SEL3
0
INF
8
Q8
PROC4.EQ.0
TA4
ATRIB(8)+ATRIB(13), 1
①
3
0
INF
10
Q10
SUBFAM
.EQ.1
0
INF
9
Q9
1
A
SUBFAM
.EQ.2
0
INF
11
Q11
ATRIB(21) = TNOW
1
TA3
SEL4
0
INF
12
Q12
A

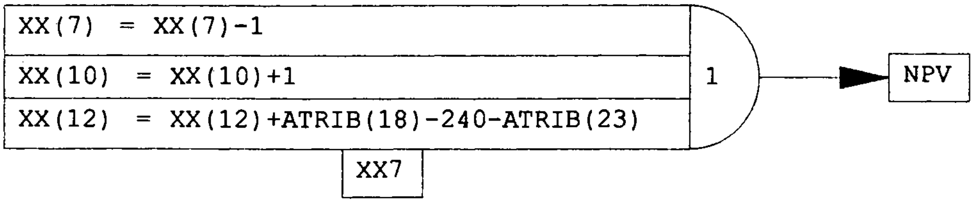

PROC2.EQ.0
TB2
ATRIB(19) = TNOW
1
TB1
SUBFAM
.EQ.4
0
16
INF
Q16
SUBFAM
.EQ.5
0
17
INF
Q17
0
18
INF
Q18
NQS(6)
POR
SEL6
ATRIB(6)+ATRIB(11), 1
1
5
NQS(5)
POR
SEL5
ATRIB(7)+ATRIB(12), 1
1
6
PROC3
.EQ.0
TB3
TA2
ATRIB(20) = TNOW
1
TB2
SUBFAM
.EQ.4
0
19
INF
Q19
SUBFAM
.EQ.5
0
20
INF
Q20
0
21
INF
Q21
NQS(7)
POR
SEL7
ATRIB(8)+ATRIB(13), 1
1
7
PROC4.EQ.0
TB4
SUBFAM
.EQ.4
0
22
INF
Q22
SUBFAM
.EQ.5
0
23
INF
Q23
0
24
INF
Q24
1
B
SEL8
ATRIB(21) = TNOW
1
TB3
B

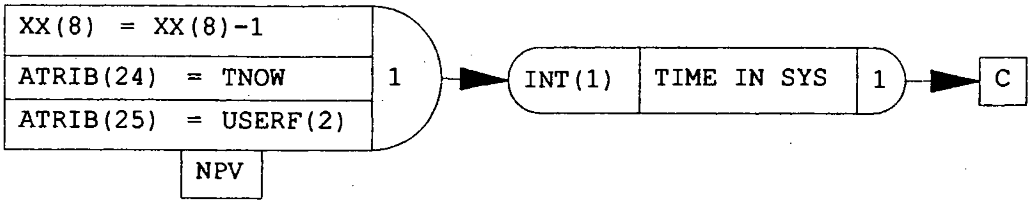

ENDP
PROC5.EQ.0
SUBFAM
.EQ.1
0
INF
25
Q25
ATRIB(22) = TNOW
1
SUBFAM
.EQ.2
0
INF
26
Q26
TA4
ATRIB(9)+ATRIB(14), 1
①
4
NQS(4)
POR
SEL4
0
INF
27
Q27
NQS(9)
POR
SEL9
ENDP
PROC5.EQ.0
SUBFAM
.EQ.4
0
INF
28
Q28
ATRIB(10)+ATRIB(15)
ATRIB(22) = TNOW
1
TB4
SUBFAM
.EQ.5
①
9
0
INF
29
Q29
ATRIB(9)+ATRIB(14), 1
①
8
NQS(8)
POR
SEL8
ENDP
0
INF
30
Q30

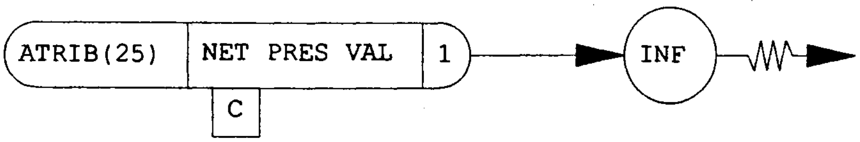

1
ENDP
0,ATRIB(3).LE.30
BAT
INF
ATRIB(23) = TNOW
ATRIB(25) = USERF(1)
:
1
BAT
ATRIB(4)
QWT
1
BAT1
30
3
ATRIB(26)
LAST/5,6,7,8,9,10,25
NONE
1
BAT1
XX6
TARD
0,ATRIB(23).GT.~~ATRIB~~(18)+240
XX(6) = XX(6)-1
1
0,ATRIB(23).LT.ATRIB(18)-240
EARL
XX6
NPV


XX(9) = XX(9)+1
XX(11) = XX(11)+ATRIB(23)-ATRIB(18)-240
1
NPV
TARD


XX(7) = XX(7)+1
1
EARL
0
INF
32
STOR
ATRIB(18)-240-TNOW, 1
①
XX7


XX(7) = XX(7)-1
XX(10) = XX(10)+1
XX(12) = XX(12)+ATRIB(18)-240-ATRIB(23)
1
NPV
XX7


XX(8) = XX(8)-1
ATRIB(24) = TNOW
ATRIB(25) = USERF(2)
1
NPV
INT(1)
TIME IN SYS
1
C


ATRIB(25)
NET PRES VAL
1
C
INF

APPENDIX E

LISTING OF SIMULATION PROGRAMS

```
;**********************************************************
;*                                                        *
;*      SLAM II NETWORK MODEL FOR GROUP SCHEDULING        *
;*                                                        *
;*      WRITTEN BY BOR-YUH JERRY LEU, NOVEMBER 1993       *
;**********************************************************
;
GEN,B. J. LEU,_ DISSERTATION,11/1/1993,25,N,N,Y/N,N,N,72;
;GEN,B. J. LEU,_DISSERTATION,11/1/1993,1,N,N,Y/N,N,Y,72;
LIMIT,32,26,32000;
PRIORITY/32,LVF(18);    [ ATRIB(18)=DUEDATE ]
;
;***** EXPERIMENTAL FACTORS =>
;*********************************************
; XX(5)=EXP. COND. (AAW_DPV_CTB), XX(1):GSH
;       1-8                               1-5
INTLC,XX(5)=1, XX(1)=1;
;*********************************************
;
;***** RUN# TO BEGIN PRT'G REPORT (REP & OA AVG) =>
INTLC,XX(18)=01,XX(19)=25;
;
;***** STOP TIME, ENTRY JOBS AT TIME 0? =>
;    (1 YEAR=120000 MINUTES)
INTLC,XX(15)=600000,XX(24)=1;
;
;***** CTB_H, CTB_L, Q TRUNC VAL, DELAY REL?, K (TWK) =>
INTLC,XX(26)=99,XX(27)=10,XX(53)=5,XX(25)=1,XX(52)=10;
;
;***** ORDER DUE DATES =>
INTLC,XX(65)=11,XX(66)=20;
;
;***** CV_OS, CV_PT, DISS.TIM? =>
INTLC,XX(50)=0.25,XX(51)=0.25,XX(23)=1;
;
;***** COST PARAMETERS FOR CALC NPV =>
; R,H,PI,PROFM, UPROC,VSTP
INTLC,XX(56)=0.15,XX(57)=0.2,XX(58)=1.0,XX(59)=0.95;
INTLC,XX(60)=9,XX(61)=15;
;
;***** RANDOM NUMBER STREAMS =>
; IS=1:INTERARRIVAL TIMES, IS=2:PROCESSING TIMES
; IS=3:ORDER SIZES,        IS=4:DUE DATES
SEEDS,5567143(1),6367787(2),9358319(3),6224571(4);
;
;***** TIME NOTATIONS =>
;
;     |<               ARRDUE             >|
;--------------------------------------------------- TIME
;     |<    'DELAY'    >|<      ALLOW     >|
;  ARRIVAL        'RELDATE'        'DUEDATE'
;
```

```
;***** ATTRIBUTES =>
;
EQUIVALENCE/ATRIB(1),RELDATE;
EQUIVALENCE/ATRIB(2),SUBFAM;
EQUIVALENCE/ATRIB(3),PTYPE;
EQUIVALENCE/ATRIB(4),INTARRT;
EQUIVALENCE/ATRIB(5),BATCH;
EQUIVALENCE/ATRIB(6),PROC1;
EQUIVALENCE/ATRIB(7),PROC2;
EQUIVALENCE/ATRIB(8),PROC3;
EQUIVALENCE/ATRIB(9),PROC4;
EQUIVALENCE/ATRIB(10),PROC5;
EQUIVALENCE/ATRIB(11),STP1;
EQUIVALENCE/ATRIB(12),STP2;
EQUIVALENCE/ATRIB(13),STP3;
EQUIVALENCE/ATRIB(14),STP4;
EQUIVALENCE/ATRIB(15),STP5;
EQUIVALENCE/ATRIB(16),DISPAT;
EQUIVALENCE/ATRIB(17),DELAY;
EQUIVALENCE/ATRIB(18),DUEDATE;
EQUIVALENCE/ATRIB(19),TWC1;
EQUIVALENCE/ATRIB(20),TWC2;
EQUIVALENCE/ATRIB(21),TWC3;
EQUIVALENCE/ATRIB(22),TWC4;
EQUIVALENCE/ATRIB(23),TCOMP;
EQUIVALENCE/ATRIB(24),TSHIP;
EQUIVALENCE/ATRIB(25),JNPV;
EQUIVALENCE/ATRIB(26),CONTN;
;
;***** GLOBAL VARIABLES =>
;
EQUIVALENCE/XX(1),GSH;
EQUIVALENCE/XX(2),AAW;
EQUIVALENCE/XX(3),DPV;
EQUIVALENCE/XX(4),CTB;
;XX(5),EXPERIMENTAL CONDITION (COMB. OF AAW_DPV_CTB)
;XX(6),# OF ORDERS IN PROCESS (WIP)
;XX(7),# OF ORDERS IN STORAGE
;XX(8),# OF ORDERS IN SYSTEM
;XX(9), TOTAL # OF TARDY ORDERS SHIPPED
;XX(10),TOTAL # OF EARLY ORDERS SHIPPED
;XX(11),TOTAL TARDINESS (TIME)
;XX(12),TOTAL EARLINESS (TIME)
;XX(15),STOPPING TIME
;XX(16),TOTAL # OF ORDERS ARRIVED
;XX(18),RUN# TO BEG PRINTING REP (CURRENT RUN)
;XX(19),RUN# TO BEG PRINTING REP (OVERALL AVG)
;XX(21),# OF JOBS IN CELL A
;XX(22),# OF JOBS IN CELL B
;XX(23),OUTPUT PT AND FST TO "DISS.TIM"? (Y=1,N=0)
;XX(24),ENTRY JOBS AT TIME 0? (Y=1,N=0)
;XX(25),DELAYED RELEASE? (Y=1,N=0)
;XX(26),CELL TRANSFER BATCH_HIGH LEVEL
```

```
;XX(27),CELL TRANSFER BATCH_LOW LEVEL
;XX(28),CELL TRANSFER BATCH (= XX(26) OR XX(27))
;XX(31),XX(41),PREVIOUS SUBFAM AT W.C. 1
;XX(32),XX(42),PREVIOUS SUBFAM AT W.C. 2
;XX(33),XX(43),PREVIOUS SUBFAM AT W.C. 3
;XX(34),XX(44),PREVIOUS SUBFAM AT W.C. 4
;XX(35),XX(45),PREVIOUS SUBFAM AT W.C. 5
;XX(36),XX(46),PREVIOUS SUBFAM AT W.C. 6
;XX(37),XX(47),PREVIOUS SUBFAM AT W.C. 7
;XX(38),XX(48),PREVIOUS SUBFAM AT W.C. 8
;XX(39),XX(49),PREVIOUS SUBFAM AT W.C. 9
;XX(50),CV FOR ORDER SIZES
;XX(51),CV FOR PROCESSING TIMES
;XX(52),K (ORDER ALLOWANCE LEVEL) FOR AAW_H
;XX(53),QUEUE TRUNCATION VALUE
;XX(56)-XX(59) FOR R,H,PI,F (COST PARAMETERS)
;XX(60)-XX(61) FOR UPROC,VSTP (COST PARAMETERS)
;XX(65)-XX(66) FOR DUE DATES
;
;***** TIME-PERSISTENT VARIABLES =>
;
TIMST,XX(6),WORK IN PROCESS;
TIMST,XX(8),N ORDERS I SYST;
TIMST,XX(7),N ORDERS I STOR;
;
;***** OBSERVATION BASED VARIABLES =>
;
STAT,4,PERCENT TARDY;
STAT,5,PERCENT EARLY;
STAT,6,PERCENT ON TM;
STAT,7,TARDINESS;
STAT,8,EARLINESS;
STAT,9,N ORDERS SHIP;
STAT,10,N ORDERS ARRIVE;
STAT,11,UTILIZATION;
STAT,12,QUEUE LENGTH;
STAT,21,TIME IN SYST_OA;
STAT,22,QUE WAIT TIM_OA;
STAT,23,NET PRES VAL_OA;
STAT,24,PERCEN TARDY_OA;
STAT,25,PERCEN EARLY_OA;
STAT,26,PERCEN ON TM_OA;
STAT,27,TARDINESS____OA;
STAT,28,EARLINESS____OA;
STAT,29,N ORDERS SHP_OA;
STAT,30,N ORDERS ARR_OA;
STAT,31,UTILIZATION__OA;
STAT,32,QUEUE LENGTH_OA;
STAT,33,DELAY________OA;
STAT,41,WORK IN PROC_OA;
STAT,42,N ORDERS I S_OA;
;
```

```
;************** NETWORK BEGINS. *****************
;
NETWORK;
;
;***** CREATE PART TYPES 1 - 5 (SUBFAM 1)
;
CR1   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,1,2;
        ACT,INTARRT,,CR1;
        ACT,DELAY,,GO1;
CR2   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,2,2;
        ACT,INTARRT,,CR2;
        ACT,DELAY,,GO1;
CR3   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,3,2;
        ACT,INTARRT,,CR3;
        ACT,DELAY,,GO1;
CR4   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,4,2;
        ACT,INTARRT,,CR4;
        ACT,DELAY,,GO1;
CR5   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,5,2;
        ACT,INTARRT,,CR5;
        ACT,DELAY,,GO1;
;
;*** UNBATCH ORDERS INTO STANDARD CONTAINERS
GO1   ASSIGN,XX(6)=XX(6)+1,XX(8)=XX(8)+1,2;
        ACT,,CONTN.GT.0,UB1;
        ACT,,,LT1;
UB1   ASSIGN,BATCH=XX(28);
      UNBATCH,26;
        ACT,,,EV31;
LT1   ASSIGN,BATCH=BATCH-CONTN*XX(28),1;
        ACT,,BATCH.LE.0,TME;
        ACT;
EV31  EVENT,31,1;
        ACT,,PROC1.EQ.0,TA1;
        ACT,,,Q1;
;
;***** CREATE PART TYPES 6 - 10 (SUBFAM 2)
;
CR6   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,6,2;
        ACT,INTARRT,,CR6;
        ACT,DELAY,,GO2;
```

```
CR7   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,7,2;
        ACT,INTARRT,,CR7;
        ACT,DELAY,,GO2;
CR8   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,8,2;
        ACT,INTARRT,,CR8;
        ACT,DELAY,,GO2;
CR9   CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,9,2;
        ACT,INTARRT,,CR9;
        ACT,DELAY,,GO2;
CR10  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,10,2;
        ACT,INTARRT,,CR10;
        ACT,DELAY,,GO2;
;
;*** UNBATCH ORDERS INTO STANDARD CONTAINERS
GO2   ASSIGN,XX(6)=XX(6)+1,XX(8)=XX(8)+1,2;
        ACT,,CONTN.GT.0,UB2;
        ACT,,,LT2;
UB2   ASSIGN,BATCH=XX(28);
      UNBATCH,26;
        ACT,,,EV32;
LT2   ASSIGN,BATCH=BATCH-CONTN*XX(28),1;
        ACT,,BATCH.LE.0,TME;
        ACT;
EV32  EVENT,32,1;
        ACT,,PROC1.EQ.0,TA1;
        ACT,,,Q2;
;
;***** CREATE PART TYPES 11 - 15 (SUBFAM 3)
;
CR11  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,11,2;
        ACT,INTARRT,,CR11;
        ACT,DELAY,,GO3;
CR12  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,12,2;
        ACT,INTARRT,,CR12;
        ACT,DELAY,,GO3;
CR13  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,13,2;
        ACT,INTARRT,,CR13;
        ACT,DELAY,,GO3;
CR14  CREATE;
```

```
          ACT,,TNOW.LT.XX(15);
        EVENT,14,2;
          ACT,INTARRT,,CR14;
          ACT,DELAY,,GO3;
CR15    CREATE;
          ACT,,TNOW.LT.XX(15);
        EVENT,15,2;
          ACT,INTARRT,,CR15;
          ACT,DELAY,,GO3;
;
;*** UNBATCH ORDERS INTO STANDARD CONTAINERS
GO3     ASSIGN,XX(6)=XX(6)+1,XX(8)=XX(8)+1,2;
          ACT,,CONTN.GT.0,UB3;
          ACT,,,LT3;
UB3     ASSIGN,BATCH=XX(28);
        UNBATCH,26;
          ACT,,,EV33;
LT3     ASSIGN,BATCH=BATCH-CONTN*XX(28),1;
          ACT,,BATCH.LE.0,TME;
          ACT;
EV33    EVENT,33,1;
          ACT,,PROC1.EQ.0,TA1;
          ACT,,,Q3;
;
;***** CREATE PART TYPES 16 - 20 (SUBFAM 4)
;
CR16    CREATE;
          ACT,,TNOW.LT.XX(15);
        EVENT,16,2;
          ACT,INTARRT,,CR16;
          ACT,DELAY,,GO13;
CR17    CREATE;
          ACT,,TNOW.LT.XX(15);
        EVENT,17,2;
          ACT,INTARRT,,CR17;
          ACT,DELAY,,GO13;
CR18    CREATE;
          ACT,,TNOW.LT.XX(15);
        EVENT,18,2;
          ACT,INTARRT,,CR18;
          ACT,DELAY,,GO13;
CR19    CREATE;
          ACT,,TNOW.LT.XX(15);
        EVENT,19,2;
          ACT,INTARRT,,CR19;
          ACT,DELAY,,GO13;
CR20    CREATE;
          ACT,,TNOW.LT.XX(15);
        EVENT,20,2;
          ACT,INTARRT,,CR20;
          ACT,DELAY,,GO13;
;
```

```
;*** UNBATCH ORDERS INTO STANDARD CONTAINERS
GO13  ASSIGN,XX(6)=XX(6)+1,XX(8)=XX(8)+1,2;
        ACT,,CONTN.GT.0,UB13;
        ACT,,,LT13;
UB13  ASSIGN,BATCH=XX(28);
      UNBATCH,26;
        ACT,,,EV34;
LT13  ASSIGN,BATCH=BATCH-CONTN*XX(28),1;
        ACT,,BATCH.LE.0,TME;
        ACT;
EV34  EVENT,34,1;
        ACT,,PROC1.EQ.0,TB1;
        ACT,,,Q13;
;
;***** CREATE PART TYPES 21 - 25 (SUBFAM 5)
;
CR21  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,21,2;
        ACT,INTARRT,,CR21;
        ACT,DELAY,,GO14;
CR22  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,22,2;
        ACT,INTARRT,,CR22;
        ACT,DELAY,,GO14;
CR23  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,23,2;
        ACT,INTARRT,,CR23;
        ACT,DELAY,,GO14;
CR24  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,24,2;
        ACT,INTARRT,,CR24;
        ACT,DELAY,,GO14;
CR25  CREATE;
        ACT,,TNOW.LT.XX(15);
      EVENT,25,2;
        ACT,INTARRT,,CR25;
        ACT,DELAY,,GO14;
;
;*** UNBATCH ORDERS INTO STANDARD CONTAINERS
GO14  ASSIGN,XX(6)=XX(6)+1,XX(8)=XX(8)+1,2;
        ACT,,CONTN.GT.0,UB14;
        ACT,,,LT14;
UB14  ASSIGN,BATCH=XX(28);
      UNBATCH,26;
        ACT,,,EV35;
LT14  ASSIGN,BATCH=BATCH-CONTN*XX(28),1;
        ACT,,BATCH.LE.0,TME;
        ACT;
EV35  EVENT,35,1;
```

```
          ACT,,PROC1.EQ.0,TB1;
          ACT,,,Q14;
;
;***** CREATE PART TYPES 26 - 30 (SUBFAM 6)
;
CR26   CREATE;
          ACT,,TNOW.LT.XX(15);
       EVENT,26,2;
          ACT,INTARRT,,CR26;
          ACT,DELAY,,GO15;
CR27   CREATE;
          ACT,,TNOW.LT.XX(15);
       EVENT,27,2;
          ACT,INTARRT,,CR27;
          ACT,DELAY,,GO15;
CR28   CREATE;
          ACT,,TNOW.LT.XX(15);
       EVENT,28,2;
          ACT,INTARRT,,CR28;
          ACT,DELAY,,GO15;
CR29   CREATE;
          ACT,,TNOW.LT.XX(15);
       EVENT,29,2;
          ACT,INTARRT,,CR29;
          ACT,DELAY,,GO15;
CR30   CREATE;
          ACT,,TNOW.LT.XX(15);
       EVENT,30,2;
          ACT,INTARRT,,CR30;
          ACT,DELAY,,GO15;
;
;*** UNBATCH ORDERS INTO STANDARD CONTAINERS
GO15   ASSIGN,XX(6)=XX(6)+1,XX(8)=XX(8)+1,2;
          ACT,,CONTN.GT.0,UB15;
          ACT,,,LT15;
UB15   ASSIGN,BATCH=XX(28);
       UNBATCH,26;
          ACT,,,EV36;
LT15   ASSIGN,BATCH=BATCH-CONTN*XX(28),1;
          ACT,,BATCH.LE.0,TME;
          ACT;
EV36   EVENT,36,1;
          ACT,,PROC1.EQ.0,TB1;
          ACT,,,Q15;
;
;***** CELL A NETWORK BEGINS.
;
;*** WORKCENTER 1 (A1)
Q1     QUEUE(1),,,,SEL1;
Q2     QUEUE(2),,,,SEL1;
Q3     QUEUE(3),,,,SEL1;
SEL1   SELECT,NQS(1),,,Q1,Q2,Q3;
          ACT/1,PROC1+STP1;
```

```
TA1   ASSIGN,TWC1=TNOW,1;
        ACT,,PROC2.EQ.0,TA2;
        ACT,,SUBFAM.EQ.1,Q4;
        ACT,,SUBFAM.EQ.2,Q5;
        ACT,,SUBFAM.EQ.3,Q6;
;
;*** WORKCENTER 2 (A2)
Q4    QUEUE(4),,,,SEL2;
Q5    QUEUE(5),,,,SEL2;
Q6    QUEUE(6),,,,SEL2;
SEL2  SELECT,NQS(2),,,Q4,Q5,Q6;
        ACT/2,PROC2+STP2;
TA2   ASSIGN,TWC2=TNOW,1;
        ACT,,PROC3.EQ.0,TA3;
        ACT,,SUBFAM.EQ.1,Q7;
        ACT,,SUBFAM.EQ.2,Q8;
        ACT,,SUBFAM.EQ.3,Q9;
;
;*** WORKCENTER 3 (A3)
Q7    QUEUE(7),,,,SEL3;
Q8    QUEUE(8),,,,SEL3;
Q9    QUEUE(9),,,,SEL3;
SEL3  SELECT,NQS(3),,,Q7,Q8,Q9;
        ACT/3,PROC3+STP3;
TA3   ASSIGN,TWC3=TNOW,1;
        ACT,,PROC4.EQ.0,TA4;
        ACT,,SUBFAM.EQ.1,Q10;
        ACT,,SUBFAM.EQ.2,Q11;
        ACT,,SUBFAM.EQ.3,Q12;
;
;*** WORKCENTER 4 (A4)
Q10   QUEUE(10),,,,SEL4;
Q11   QUEUE(11),,,,SEL4;
Q12   QUEUE(12),,,,SEL4;
SEL4  SELECT,NQS(4),,,Q10,Q11,Q12;
        ACT/4,PROC4+STP4;
TA4   ASSIGN,TWC4=TNOW,XX(23)=XX(23)+1,1;
        ACT,,PROC5.EQ.0,ENDP;
        ACT,,SUBFAM.EQ.1,Q25;
        ACT,,SUBFAM.EQ.2,Q26;
        ACT,,SUBFAM.EQ.3,Q27;
;
;***** CELL B NETWORK BEGINS.
;
;*** WORKCENTER 5 (B1)
Q13   QUEUE(13),,,,SEL5;
Q14   QUEUE(14),,,,SEL5;
Q15   QUEUE(15),,,,SEL5;
SEL5  SELECT,NQS(5),,,Q13,Q14,Q15;
        ACT/5,PROC1+STP1;
TB1   ASSIGN,TWC1=TNOW,1;
        ACT,,PROC2.EQ.0,TB2;
        ACT,,SUBFAM.EQ.4,Q16;
```

```
          ACT,,SUBFAM.EQ.5,Q17;
          ACT,,SUBFAM.EQ.6,Q18;
;
;*** WORKCENTER 6 (B2)
Q16    QUEUE(16),,,,SEL6;
Q17    QUEUE(17),,,,SEL6;
Q18    QUEUE(18),,,,SEL6;
SEL6   SELECT,NQS(6),,,Q16,Q17,Q18;
          ACT/6,PROC2+STP2;
TB2    ASSIGN,TWC2=TNOW,1;
          ACT,,PROC3.EQ.0,TB3;
          ACT,,SUBFAM.EQ.4,Q19;
          ACT,,SUBFAM.EQ.5,Q20;
          ACT,,SUBFAM.EQ.6,Q21;
;
;*** WORKCENTER 7 (B3)
Q19    QUEUE(19),,,,SEL7;
Q20    QUEUE(20),,,,SEL7;
Q21    QUEUE(21),,,,SEL7;
SEL7   SELECT,NQS(7),,,Q19,Q20,Q21;
          ACT/7,PROC3+STP3;
TB3    ASSIGN,TWC3=TNOW,1;
          ACT,,PROC4.EQ.0,TB4;
          ACT,,SUBFAM.EQ.4,Q22;
          ACT,,SUBFAM.EQ.5,Q23;
          ACT,,SUBFAM.EQ.6,Q24;
;
;*** WORKCENTER 8 (B4)
Q22    QUEUE(22),,,,SEL8;
Q23    QUEUE(23),,,,SEL8;
Q24    QUEUE(24),,,,SEL8;
SEL8   SELECT,NQS(8),,,Q22,Q23,Q24;
          ACT/8,PROC4+STP4;
TB4    ASSIGN,TWC4=TNOW,XX(24)=XX(24)+1,1;
          ACT,,PROC5.EQ.0,ENDP;
          ACT,,SUBFAM.EQ.4,Q28;
          ACT,,SUBFAM.EQ.5,Q29;
          ACT,,SUBFAM.EQ.6,Q30;
;
;***** SHARED WORKCENTER 9 (A5/B5)
;
Q25    QUEUE(25),,,,SEL9;
Q26    QUEUE(26),,,,SEL9;
Q27    QUEUE(27),,,,SEL9;
Q28    QUEUE(28),,,,SEL9;
Q29    QUEUE(29),,,,SEL9;
Q30    QUEUE(30),,,,SEL9;
SEL9   SELECT,NQS(9),,,Q25,Q26,Q27,Q28,Q29,Q30;
          ACT/9,PROC5+STP5;
ENDP   GOON,1;
          ACT,,PTYPE.LE.30,PASS;
          ACT;
       COLCT,ATRIB(3),INITIAL JOBS;
```

```
      TERM;
PASS  GOON,1;
        ACT,,SUBFAM.LE.3,XX21;
        ACT,,SUBFAM.GE.4,XX22;
XX21  ASSIGN,XX(21)=XX(21)-1;
        ACT,,,BAT;
XX22  ASSIGN,XX(22)=XX(22)-1;
;
;***** BATCH STANDARD CONTAINERS INTO ORDERS
;
BAT   ASSIGN,TCOMP=TNOW,JNPV=USERF(1);
      ASSIGN,ATRIB(26)=ATRIB(26)+ATRIB(24);
      ASSIGN,ATRIB(4)=TNOW-RELDATE-ATRIB(6)-ATRIB(7)-
          ATRIB(8)-ATRIB(9)-ATRIB(10)-ATRIB(11)-
          ATRIB(12)-ATRIB(13)-ATRIB(14)-ATRIB(15);
      COLCT(2),ATRIB(4),QUE WAIT TIME;
      BATCH,30/3,ATRIB(26),,LAST/5,6,7,8,9,10,25;
      ASSIGN,XX(6)=XX(6)-1,1;
        ACT,,TCOMP.GT.DUEDATE+240,TARD;
        ACT,,TCOMP.LT.DUEDATE-240,EARL;
        ACT,,,NPV;
;
;***** TARDY BRANCH
;
TARD  ASSIGN,XX(9)=XX(9)+1;
      ASSIGN,XX(11)=XX(11)+TCOMP-DUEDATE-240;
        ACT,,,NPV;
;
;***** EARLY BRANCH
;
EARL  ASSIGN,XX(7)=XX(7)+1;
STOR  QUEUE(32);
        ACT,DUEDATE-240-TNOW;
      ASSIGN,XX(7)=XX(7)-1,XX(10)=XX(10)+1;
      ASSIGN,XX(12)=XX(12)+DUEDATE-240-TCOMP;
;
;***** COLLECT STATISTICS
;
NPV   ASSIGN,TSHIP=TNOW,JNPV=USERF(2),XX(8)=XX(8)-1;
      COLCT(1),INT(1),TIME IN SYS;
      COLCT(3),JNPV,NET PRES VAL;
      COLCT(13),DELAY,DELAY B REL;
TME   TERM;
;
      ENDNET;
;
;************** NETWORK ENDS. *******************
;
;***** INIT: JJCLR=Y/21 => CLEAR STAT ARRAYS BET RUNS?
;               CLEAR VAR TYPES 1 - 20
;               CUMULATE VAR TYPES FROM 21
INIT,0,,Y/21;
;
```

```
;***** RANDOM NUMBER STREAMS FOR RUNS 2 - 25
;
SIM;
SEEDS,7498797(1),1328413(2),6785355(3),1649781(4);
SIM;
SEEDS,8452461(1),2122437(2),7823661(3),4617649(4);
SIM;
SEEDS,7974355(1),9905247(2),5785195(3),4545375(4);
SIM;
SEEDS,6196569(1),5099389(2),1245769(3),8812961(4);
SIM;
SEEDS,3624681(1),6057883(2),2896981(3),1991139(4);
SIM;
SEEDS,5683129(1),9124081(2),7114317(3),9814817(4);
SIM;
SEEDS,7576887(1),7945841(2),5846367(3),8257261(4);
SIM;
SEEDS,2855897(1),3524963(2),6778633(3),3981825(4);
SIM;
SEEDS,2292817(1),3898065(2),9452545(3),6534895(4);
SIM;
SEEDS,9383211(1),1075025(2),7693821(3),2735961(4);
SIM;
SEEDS,1634765(1),3624727(2),9525638(3),7948247(4);
SIM;
SEEDS,5967585(1),7099469(2),3569469(3),1253965(4);
SIM;
SEEDS,4147823(1),9963849(2),7442563(3),5741729(4);
SIM;
SEEDS,2919379(1),7205515(2),8418927(3),3224619(4);
SIM;
SEEDS,4778589(1),2403865(2),6284381(3),1649113(4);
SIM;
SEEDS,2929865(1),7497641(2),1369741(3),8136413(4);
SIM;
SEEDS,7665431(1),3553471(2),3212271(3),8115765(4);
SIM;
SEEDS,9389861(1),5367621(2),3746155(3),2618471(4);
SIM;
SEEDS,8172627(1),7481567(2),2789747(3),5898783(4);
SIM;
SEEDS,4342773(1),4524685(2),6621617(3),8517591(4);
SIM;
SEEDS,5928337(1),7650947(2),1238343(3),3999723(4);
SIM;
SEEDS,1651165(1),1968909(2),7473889(3),3256669(4);
SIM;
SEEDS,6283683(1),4275135(2),1917567(3),7464179(4);
SIM;
SEEDS,2496481(1),1194621(2),4859797(3),1681455(4);
;
FIN;
;
```

```
C
C***********************************************************
C*                                                         *
C*        FORTRAN SUBPROGRAMS FOR GROUP SCHEDULING:        *
C*          INTLC, OTPUT, EVENT, USERF, NQS, SELJOB        *
C*          (INITNQS, SELNQS, COMPA)                       *
C*                                                         *
C*        WRITTEN BY BOR-YUH JERRY LEU, NOVEMBER 1993      *
C***********************************************************
C
      PROGRAM MAIN
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      COMMON QSET(1000000)
      DIMENSION NSET(1000000)
      EQUIVALENCE(NSET(1),QSET(1))
      NNSET=1000000
      NCRDR=5
      NPRNT=6
      NTAPE=7
C*** OUTPUT PROCESSING AND SETUP TIMES
      OPEN (60,FILE='DISS.TIM',STATUS='NEW')
C*** OUTPUT ALL MEASURES FOR STATISTICAL ANALYSIS
      OPEN (70,FILE='DISS.SAS',STATUS='NEW')
      OPEN (80,FILE='DISS.VIP',STATUS='NEW')
      CALL SLAM
      STOP
      END
C
C***********************************************************
C*            SUBROUTINE INTLC                             *
C*            -- SET INITIAL CONDITIONS AT THE BEGINNING   *
C*               OF EACH RUN (INCLUDING TO GENERATE        *
C*               PROCESSING AND SETUP TIMES, AND TO ENTRY  *
C*               JOBS AT TIME 0)                           *
C***********************************************************
C
      SUBROUTINE INTLC
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/AD(30),AOH(30),AOL(30),PT(30,5),FST(6,5)
C
      EQUIVALENCE(ATRIB(1),RELDATE),(ATRIB(2),SUBFAM)
      EQUIVALENCE(ATRIB(3),PTYPE),(ATRIB(4),INTARRT)
      EQUIVALENCE(ATRIB(5),BATCH),(ATRIB(6),PROC1)
      EQUIVALENCE(ATRIB(7),PROC2),(ATRIB(8),PROC3)
      EQUIVALENCE(ATRIB(9),PROC4),(ATRIB(10),PROC5)
      EQUIVALENCE(ATRIB(11),STP1),(ATRIB(12),STP2)
      EQUIVALENCE(ATRIB(13),STP3),(ATRIB(14),STP4)
      EQUIVALENCE(ATRIB(15),STP5),(ATRIB(16),DISPAT)
      EQUIVALENCE(ATRIB(17),DELAY),(ATRIB(18),DUEDATE)
```

```
      EQUIVALENCE(ATRIB(19),TWC1),(ATRIB(20),TWC2)
      EQUIVALENCE(ATRIB(21),TWC3),(ATRIB(22),TWC4)
      EQUIVALENCE(ATRIB(23),TCOMP),(ATRIB(24),TSHIP)
      EQUIVALENCE(ATRIB(25),JNPV),(ATRIB(26),CONTN)
      EQUIVALENCE(XX(1),GSH),(XX(2),AAW),(XX(3),DPV)
      EQUIVALENCE(XX(4),CTB)
      REAL INTARRT,JNPV
C
      REAL AAPW(9),AAPWH(9),AAPWL(9),IAPT(30),IPT(30,5),
     1     IPW(30,5),TIPW(9)
      REAL AASW(9),AASWH(9),AASWL(9),ISW(30,5),TISW(9),
     1     ASW(30,5),AC(30),ST(30,5)
      DIMENSION NPART(6,5),NCPOH(30),NCPOL(30),AT(100)
C
      DATA XX(6),XX(7),XX(8),XX(9),XX(10),XX(11),XX(12),
     1     XX(16),XX(21),XX(22) /10*0./
      DATA XX(31),XX(32),XX(33),XX(34),XX(35),XX(36),XX(37),
     1     XX(38),XX(39) /9*1./
      DATA XX(41),XX(42),XX(43),XX(44),XX(45),XX(46),XX(47),
     1     XX(48),XX(49) /9*0./
C*** HIGH LEVEL OF AAW
C* AAW:90%
      DATA AAPWH /1481.,1632.,1619.,1796.,1657.,1608.,1596.,
     1            1635.,1704./
      DATA AASWH /185.,175.,173.,185.,174.,188.,197.,176.,
     1            183./
C*** LOW LEVEL OF AAW
C* AAW:80%
      DATA AAPWL /1317.,1451.,1439.,1597.,1473.,1429.,1419.,
     1            1454.,1515./
      DATA AASWL /164.,155.,154.,164.,155.,167.,175.,156.,
     1            163./
C*** AD, AOH, & AOL
      DATA AD /1135.,548.,118.,1257.,358.,767.,196.,1025.,
     1         248.,1281.,185.,942.,1043.,100.,1223.,228.,
     2         969.,1156.,899.,280.,336.,675.,937.,1053.,
     3         456.,659.,1173.,687.,475.,517./
      DATA AOH /22.,11.,17.,24.,15.,15.,26.,18.,19.,29.,
     1          15.,20.,19.,10.,28.,17.,19.,27.,25.,10.,
     2          29.,16.,28.,19.,23.,15.,26.,30.,23.,24./
      DATA AOL /31.,46.,32.,49.,35.,38.,39.,42.,49.,47.,
     1          33.,31.,47.,48.,50.,39.,41.,46.,36.,42.,
     2          34.,37.,43.,47.,42.,42.,39.,33.,31.,38./
C*** (ROUTING-RELATED)
      DATA ((NPART(K,J),J=1,5),K=1,6) /4,5,4,3,4, 4,4,4,5,4,
     1  4,3,4,4,4, 4,3,3,4,4, 5,5,4,4,4, 3,4,5,4,4/
C
      IS2=2
      CVPT=XX(51)
C
C***** DEFINE XX(5): COMB. OF AAW_DPV_CTB
C
      IF(XX(5).EQ.1.OR.XX(5).EQ.2.OR.XX(5).EQ.3.OR.
```

```
     1     XX(5).EQ.4) AAW=1.
      IF(XX(5).EQ.5.OR.XX(5).EQ.6.OR.XX(5).EQ.7.OR.
     1     XX(5).EQ.8) AAW=2.
      IF(XX(5).EQ.1.OR.XX(5).EQ.2.OR.XX(5).EQ.5.OR.
     1     XX(5).EQ.6) DPV=1.
      IF(XX(5).EQ.3.OR.XX(5).EQ.4.OR.XX(5).EQ.7.OR.
     1     XX(5).EQ.8) DPV=2.
      IF(XX(5).EQ.1.OR.XX(5).EQ.3.OR.XX(5).EQ.5.OR.
     1     XX(5).EQ.7) CTB=1.
      IF(XX(5).EQ.2.OR.XX(5).EQ.4.OR.XX(5).EQ.6.OR.
     1     XX(5).EQ.8) CTB=2.
C
C##### CHOOSE A LEVEL OF "CTB"
C
      IF(CTB.EQ.1) XX(28)=XX(26)
      IF(CTB.EQ.2) XX(28)=XX(27)
C
C***** GENERATE PROCESSING TIMES
C      WHICH DEPEND ON AAW ONLY.
C
C### CHOOSE A LEVEL OF "AAW" TO SPECIFY AAPW
      DO 5 I=1,9
        IF (AAW.EQ.1.0) AAPW(I)=AAPWH(I)
        IF (AAW.EQ.2.0) AAPW(I)=AAPWL(I)
    5 CONTINUE
C*** COMPUTE IAPT(30) & IPT(30,5)
      DO 20 I=1,30
        IAPT(I)=UNFRM(5.,15.,IS2)
      DO 20 J=1,5
   10   IPT(I,J)=RNORM(IAPT(I),IAPT(I)*CVPT,IS2)
        IF(IPT(I,J).LE.0) GO TO 10
   20 CONTINUE
C*** IPT()=0 (ROUTING-RELATED)
      IPT(1,1)=0.0
      IPT(8,1)=0.0
      IPT(12,1)=0.0
      IPT(7,2)=0.0
      IPT(11,2)=0.0
      IPT(14,2)=0.0
      IPT(4,3)=0.0
      IPT(9,3)=0.0
      IPT(15,3)=0.0
      IPT(3,4)=0.0
      IPT(5,4)=0.0
      IPT(12,4)=0.0
      IPT(17,1)=0.0
      IPT(26,1)=0.0
      IPT(28,1)=0.0
      IPT(16,2)=0.0
      IPT(19,2)=0.0
      IPT(28,2)=0.0
      IPT(17,3)=0.0
      IPT(19,3)=0.0
```

```
      IPT(23,3)=0.0
      IPT(16,4)=0.0
      IPT(21,4)=0.0
      IPT(30,4)=0.0
      IPT(3,5)=0.0
      IPT(9,5)=0.0
      IPT(14,5)=0.0
      IPT(20,5)=0.0
      IPT(23,5)=0.0
      IPT(26,5)=0.0
C*** COMPUTE IPW(30,5)
      DO 35 I=1,30
      DO 35 J=1,5
        IPW(I,J)=IPT(I,J)*AD(I)
  35  CONTINUE
C*** COMPUTE TIPW(9)
      DO 40 J=1,4
        TIPW(J)=0.
      DO 40 I=1,15
        TIPW(J)=TIPW(J)+IPW(I,J)
  40  CONTINUE
      DO 50 J=5,8
        TIPW(J)=0.
      DO 50 I=16,30
        TIPW(J)=TIPW(J)+IPW(I,J-4)
  50  CONTINUE
        TIPW(9)=0.
      DO 60 I=1,30
        TIPW(9)=TIPW(9)+IPW(I,5)
  60  CONTINUE
C*** COMPUTE PT(30,5)
      DO 70 I=1,15
      DO 70 J=1,4
        PT(I,J)=IPT(I,J)*AAPW(J)/TIPW(J)*60.
  70  CONTINUE
      DO 80 I=16,30
      DO 80 J=1,4
        PT(I,J)=IPT(I,J)*AAPW(J+4)/TIPW(J+4)*60.
  80  CONTINUE
      DO 90 I=1,30
        PT(I,5)=IPT(I,5)*AAPW(9)/TIPW(9)*60.
  90  CONTINUE
C
C***** GENERATE MAJOR (FAMILY) SETUP TIMES
C      WHICH DEPEND ON AAW & CTB ONLY.
C
C### CHOOSE A LEVEL OF "AAW" TO SPECIFY AASW
      DO 95 I=1,9
        IF (AAW.EQ.1.0) AASW(I)=AASWH(I)
        IF (AAW.EQ.2.0) AASW(I)=AASWL(I)
  95  CONTINUE
C*** COMPUTE ISW(30,5)
      DO 100 I=1,15
```

```
      DO 100 J=1,4
        ISW(I,J)=AASW(J)*UNFRM(0.2,1.0,IS2)
 100  CONTINUE
      DO 110 I=16,30
      DO 110 J=1,4
        ISW(I,J)=AASW(J+4)*UNFRM(0.2,1.0,IS2)
 110  CONTINUE
      DO 120 I=1,30
        ISW(I,5)=AASW(9)*UNFRM(0.2,1.0,IS2)
 120  CONTINUE
C*** ISW()=0 (ROUTING-RELATED)
      ISW(1,1)=0.0
      ISW(8,1)=0.0
      ISW(12,1)=0.0
      ISW(7,2)=0.0
      ISW(11,2)=0.0
      ISW(14,2)=0.0
      ISW(4,3)=0.0
      ISW(9,3)=0.0
      ISW(15,3)=0.0
      ISW(3,4)=0.0
      ISW(5,4)=0.0
      ISW(12,4)=0.0
      ISW(17,1)=0.0
      ISW(26,1)=0.0
      ISW(28,1)=0.0
      ISW(16,2)=0.0
      ISW(19,2)=0.0
      ISW(28,2)=0.0
      ISW(17,3)=0.0
      ISW(19,3)=0.0
      ISW(23,3)=0.0
      ISW(16,4)=0.0
      ISW(21,4)=0.0
      ISW(30,4)=0.0
      ISW(3,5)=0.0
      ISW(9,5)=0.0
      ISW(14,5)=0.0
      ISW(20,5)=0.0
      ISW(23,5)=0.0
      ISW(26,5)=0.0
C*** COMPUTE TISW(9)
      DO 140 J=1,4
        TISW(J)=0.
      DO 140 I=1,15
        TISW(J)=TISW(J)+ISW(I,J)
 140  CONTINUE
      DO 150 J=5,8
        TISW(J)=0.
      DO 150 I=16,30
        TISW(J)=TISW(J)+ISW(I,J-4)
 150  CONTINUE
        TISW(9)=0.
```

```
      DO 160 I=1,30
        TISW(9)=TISW(9)+ISW(I,5)
 160  CONTINUE
C*** COMPUTE ASW(30,5)
      DO 170 I=1,15
      DO 170 J=1,4
        ASW(I,J)=ISW(I,J)*AASW(J)/TISW(J)
 170  CONTINUE
      DO 180 I=16,30
      DO 180 J=1,4
        ASW(I,J)=ISW(I,J)*AASW(J+4)/TISW(J+4)
 180  CONTINUE
      DO 190 I=1,30
        ASW(I,5)=ISW(I,5)*AASW(9)/TISW(9)
 190  CONTINUE
C*** COMPUTE ST(30,5) FOR EACH PART TYPE,
C    AC(I) OBTAINED BY AVERAGING DPV_H & DPV_L,
C    ("ST" & "FST" ARE NOT CHANGED W/ LEVEL OF DPV)
      DO 200 I=1,30
        NCPOH(I)=AD(I)/AOH(I)/XX(28)+1
        NCPOL(I)=AD(I)/AOL(I)/XX(28)+1
        AC(I)=(AOH(I)*NCPOH(I)+AOL(I)*NCPOL(I))/2.
      DO 200 J=1,5
        ST(I,J)=ASW(I,J)/AC(I)*60.
 200  CONTINUE
C*** COMPUTE FST(6,5) FOR EACH SUBFAMILY
      DO 220 J=1,5
      DO 220 K=1,6
        FST(K,J)=0.0
        DO 210 I=5*K-4,5*K
          FST(K,J)=FST(K,J)+ST(I,J)
 210    CONTINUE
        FST(K,J)=FST(K,J)/NPART(K,J)
 220  CONTINUE
C
C***** OUTPUT PT AND FST TO "DISS.TIM"
C
      IF(XX(23).EQ.1) THEN
        WRITE(60,*) '***** RUN = ',NNRUN
        WRITE(60,*) 'PT(30,5)='
        WRITE(60,290) (I,(PT(I,J),J=1,5),I=1,30)
        WRITE(60,*) 'FST(6,5)='
        WRITE(60,290) (I,(FST(I,J),J=1,5),I=1,6)
 290    FORMAT(1X,I3,5F12.3/)
      ENDIF
C
C***** ENTRY JOBS TO QUEUES AT TIME 0
C      BASED ON E.C.=1, GSH=2, RUN LENGTH=6-MONTH
C
      IF(XX(24).EQ.1) THEN
        AT(3)=99
        AT(5)=XX(54)
        AT(18)=1600
```

```
          DO 300 I=6,10
            AT(I)=200
 300      CONTINUE
C
          DO 305 I=1,2
          AT(2)=2
 305      CALL FILEM(5,AT)
          DO 307 I=1,1
          AT(2)=1
 307      CALL FILEM(7,AT)
          DO 308 I=1,5
          AT(2)=2
 308      CALL FILEM(8,AT)
          DO 310 I=1,2
          AT(2)=1
 310      CALL FILEM(10,AT)
          DO 311 I=1,6
          AT(2)=2
 311      CALL FILEM(11,AT)
          DO 312 I=1,10
          AT(2)=3
 312      CALL FILEM(12,AT)
          DO 313 I=1,5
          AT(2)=4
 313      CALL FILEM(13,AT)
          DO 315 I=1,8
          AT(2)=6
 315      CALL FILEM(15,AT)
          DO 318 I=1,5
          AT(2)=6
 318      CALL FILEM(18,AT)
          DO 319 I=1,3
          AT(2)=4
 319      CALL FILEM(19,AT)
          DO 320 I=1,5
          AT(2)=5
 320      CALL FILEM(20,AT)
          DO 321 I=1,9
          AT(2)=6
 321      CALL FILEM(21,AT)
        ENDIF
        RETURN
        END
C
C************************************************************
C*            SUBROUTINE OTPUT                             *
C*            -- END-OF-RUN PROCESSING AT THE END OF       *
C*               EACH RUN (I.E., OUTPUT STAT TO FILES)     *
C************************************************************
C
      SUBROUTINE OTPUT
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
```

```
     2  SSL(100),TNEXT,TNOW,XX(100)
C
C***** COLLECT STATISTICS
C
      TOTSHP=CCNUM(1)
      TOTARR=XX(16)
      IF(TOTSHP.EQ.0) GO TO 5
      PTARDY=XX(9)/TOTSHP*100.
      PEARLY=XX(10)/TOTSHP*100.
      PONTM=100-PTARDY-PEARLY
      TARDY=XX(11)/XX(9)
      EARLY=XX(12)/XX(10)
C*** PERCENT TARDY
    5 CALL COLCT(PTARDY,4)
C*** PERCENT EARLY
      CALL COLCT(PEARLY,5)
C*** PERCENT ON TIME
      CALL COLCT(PONTM,6)
C*** AVG TARDINESS
      CALL COLCT(TARDY,7)
C*** AVG EARLINESS
      CALL COLCT(EARLY,8)
C*** # OF ORDERS SHIPPED
      CALL COLCT(TOTSHP,9)
C*** # OF ORDERS ARRIVED
      CALL COLCT(TOTARR,10)
C*** AVG MACHINE UTILIZATION
      SUMU=0
      DO 10 I=1,9
        SUMU=SUMU+AAAVG(I)
   10 CONTINUE
      AVGU=SUMU/9.0*100.
      CALL COLCT(AVGU,11)
C*** AVG QUEUE LENGTH
      SUMQL=0
      DO 20 I=1,30
        SUMQL=SUMQL+FFAVG(I)
   20 CONTINUE
      AVGQL=SUMQL/30.0
      CALL COLCT(AVGQL,12)
C
C*** COLLECT WIHIN A RUN ===>
      GRAND1=CCAVG(1)
      GRAND2=CCAVG(2)
      GRAND3=CCAVG(3)
      GRAND4=CCAVG(4)
      GRAND5=CCAVG(5)
      GRAND6=CCAVG(6)
      GRAND7=CCAVG(7)
      GRAND8=CCAVG(8)
      GRAND9=CCAVG(9)
      GRAND10=CCAVG(10)
      GRAND11=CCAVG(11)
```

```
      GRAND12=CCAVG(12)
      GRAND13=CCAVG(13)
      GRANDT1=TTAVG(1)
      GRANDT2=TTAVG(2)
C*** COLLECT AMONG RUNS (OVERALL) ===>
      CALL COLCT(GRAND1,21)
      CALL COLCT(GRAND2,22)
      CALL COLCT(GRAND3,23)
      CALL COLCT(GRAND4,24)
      CALL COLCT(GRAND5,25)
      CALL COLCT(GRAND6,26)
      CALL COLCT(GRAND7,27)
      CALL COLCT(GRAND8,28)
      CALL COLCT(GRAND9,29)
      CALL COLCT(GRAND10,30)
      CALL COLCT(GRAND11,31)
      CALL COLCT(GRAND12,32)
      CALL COLCT(GRAND13,33)
      CALL COLCT(GRANDT1,41)
      CALL COLCT(GRANDT2,42)
C
C***** OUTPUT CURRENT RUN (REP) TO "DISS.SAS"
C
      IF(NNRUN.GE.XX(18)) THEN
        IF(NNRUN.EQ.1) THEN
          WRITE(70,50) XX(5),XX(1),TNOW
  50      FORMAT(1X,'***** EXP/GSH = ',F2.0,'/',F2.0,
     1             ', TNOW = ',F8.0,' ===>')
        ENDIF
C
C*** [ ]: MEASURES INCLUDED IN DISSERTATION.....
C    [1] TIME IN SYSTEM         [2] QUE WAIT TM P TRAN BAT
C    [3] NET PRESENT VALUE      [4] WORK-IN-PROCESS
C    [5] # OF ORDERS IN SYS.    [6] % TARDY
C    [7] % EARLY                [8] % ON TIME
C    [9] TARDINESS              [10] EARLINESS
C    <11> # OF ORDERS SHIPPED   <12> UTILIZATION
C    <13> DELAY BEF RELEASE
C
        WRITE(70,55)
     1    CCAVG(1),CCAVG(2),CCAVG(3),TTAVG(1),TTAVG(2),
     2    CCAVG(4),CCAVG(5),CCAVG(6),CCAVG(7),CCAVG(8),
     3    CCAVG(9),CCAVG(11),CCAVG(13)
  55    FORMAT(10F10.2,3F10.2)
      ENDIF
C
C***** OUPUT OVERALL AVERAGE (ACROSS RUNS) TO "DISS.VIP"
C
      IF(NNRUN.GE.XX(19)) THEN
        WRITE(80,*)
     1     '-----------------------------------------------'
        WRITE(80,60) XX(5),XX(1),NNRUN,TNOW
  60    FORMAT(1X,'EXP/GSH = ',F2.0,'/',F2.0,', NNRUN = ',I2,
```

```
     1   ', TNOW = ',F8.0)
       WRITE(80,61) CCAVG(21),CCSTD(21)
  61   FORMAT(1X,'TIME IN SYSTE_OA ',2F15.2)
       WRITE(80,62) CCAVG(22),CCSTD(22)
  62   FORMAT(1X,'Q W TIM P T B_OA ',2F15.2)
       WRITE(80,63) CCAVG(23),CCSTD(23)
  63   FORMAT(1X,'NET PRES VALU_OA ',2F15.2)
       WRITE(80,64) CCAVG(41),CCSTD(41)
  64   FORMAT(1X,'WORK IN PROC_OA  ',2F15.2)
       WRITE(80,65) CCAVG(42),CCSTD(42)
  65   FORMAT(1X,'N ORDERS I S _OA ',2F15.2)
       WRITE(80,*) '-----------------'
       WRITE(80,66) CCAVG(24),CCSTD(24)
  66   FORMAT(1X,'PERCENT TARDY_OA ',2F15.2)
       WRITE(80,67) CCAVG(25),CCSTD(25)
  67   FORMAT(1X,'PERCENT EARLY_OA ',2F15.2)
       WRITE(80,68) CCAVG(26),CCSTD(26)
  68   FORMAT(1X,'PERCENT ON TM_OA ',2F15.2)
       WRITE(80,69) CCAVG(27),CCSTD(27)
  69   FORMAT(1X,'TARDINESS____OA  ',2F15.2)
       WRITE(80,70) CCAVG(28),CCSTD(28)
  70   FORMAT(1X,'EARLINESS____OA  ',2F15.2)
      ENDIF
      RETURN
      END
C
C************************************************************
C*            SUBROUTINE EVENT                              *
C*            -- ASSIGN BASIC DATA TO AN ARRIVING ORDER     *
C*               OR A JOB (TRANSFER BATCH)                  *
C*                                                          *
C************************************************************
C
      SUBROUTINE EVENT(N)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/AD(30),AOH(30),AOL(30),PT(30,5),FST(6,5)
C
      EQUIVALENCE(ATRIB(1),RELDATE),(ATRIB(2),SUBFAM)
      EQUIVALENCE(ATRIB(3),PTYPE),(ATRIB(4),INTARRT)
      EQUIVALENCE(ATRIB(5),BATCH),(ATRIB(6),PROC1)
      EQUIVALENCE(ATRIB(7),PROC2),(ATRIB(8),PROC3)
      EQUIVALENCE(ATRIB(9),PROC4),(ATRIB(10),PROC5)
      EQUIVALENCE(ATRIB(11),STP1),(ATRIB(12),STP2)
      EQUIVALENCE(ATRIB(13),STP3),(ATRIB(14),STP4)
      EQUIVALENCE(ATRIB(15),STP5),(ATRIB(16),DISPAT)
      EQUIVALENCE(ATRIB(17),DELAY),(ATRIB(18),DUEDATE)
      EQUIVALENCE(ATRIB(19),TWC1),(ATRIB(20),TWC2)
      EQUIVALENCE(ATRIB(21),TWC3),(ATRIB(22),TWC4)
      EQUIVALENCE(ATRIB(23),TCOMP),(ATRIB(24),TSHIP)
      EQUIVALENCE(ATRIB(25),JNPV),(ATRIB(26),CONTN)
      EQUIVALENCE(XX(1),GSH),(XX(2),AAW),(XX(3),DPV)
```

```
      EQUIVALENCE(XX(4),CTB)
      REAL INTARRT,JNPV
C
      IF(N.GT.30) GO TO 100
C
      IS1=1
      IS3=3
      IS4=4
      CVOS=XX(50)
      TWK=XX(52)
      DISPAT=0.
      XX(16)=XX(16)+1
C
C***** ASSIGN PART TYPE AND SUBFAMILY NO. TO AN ORDER
      IF(N.GE.1.AND.N.LE.5)   SUBFAM=1.
      IF(N.GE.6.AND.N.LE.10)  SUBFAM=2.
      IF(N.GE.11.AND.N.LE.15) SUBFAM=3.
      IF(N.GE.16.AND.N.LE.20) SUBFAM=4.
      IF(N.GE.21.AND.N.LE.25) SUBFAM=5.
      IF(N.GE.26.AND.N.LE.30) SUBFAM=6.
      PTYPE=N
      IPTYPE=N
C##### CHOOSE A LEVEL OF "DPV" (TO DECIDE AORDER) &
C       COMPUTE INTERARRIVAL TIME AND ORDER SIZE
      IF(DPV.EQ.1) AORDER=AOH(IPTYPE)
      IF(DPV.EQ.2) AORDER=AOL(IPTYPE)
      ARRT=120000./AORDER
      INTARRT=EXPON(ARRT,IS1)
      AOSIZE=AD(IPTYPE)/AORDER
   10 IBATCH=RNORM(AOSIZE,AOSIZE*CVOS,IS3)
      IF(IBATCH.LE.0) GO TO 10
      BATCH=IBATCH+1
C***** CALCULATE # OF TRANSFER BATCHES REQ FOR THIS ORDER
      CONT1=BATCH/XX(28)
      ICONT1=CONT1
      CONTN=ICONT1
      IF(CONTN.EQ.CONT1) THEN
        ATRIB(24)=0
      ELSE
        ATRIB(24)=1
      ENDIF
C***** CALCULATE ORDER ALLOWANCE FOR DELAYED RELEASE
      IF(XX(25).EQ.1) THEN
        TPROC=0
        DO 20 I=1,5
   20   TPROC=TPROC+PT(IPTYPE,I)*BATCH
        ALLOW=TWK*TPROC
      ENDIF
C***** CALCULATE ORDER DUE DATE
      XX65=XX(65)*480.0
      XX66=XX(66)*480.0
      ARRDUE=UNFRM(XX65,XX66,IS4)
      DUEDATE=TNOW+ARRDUE
```

```
C***** CALCULATE ORDER RELEASE TIME AND DELAY BEFORE RELEASE
      IF(XX(25).EQ.0) THEN
        RELDATE=TNOW
        DELAY=0
      ELSE
        IF(ARRDUE.GT.ALLOW) THEN
          RELDATE=DUEDATE-ALLOW
          DELAY=ARRDUE-ALLOW
        ELSE
          RELDATE=TNOW
          DELAY=0
        ENDIF
      ENDIF
      RETURN
C
C
100   IF(SUBFAM.LE.3) XX(21)=XX(21)+1
      IF(SUBFAM.GE.4) XX(22)=XX(22)+1
C***** INITIALLY SET SETUP TIMES AT W.C. 1 - 5
      STP1=0.
      STP2=0.
      STP3=0.
      STP4=0.
      STP5=0.
C***** COMPUTE PROCESSING TIMES AT W.C. 1 - 5
      IPTYPE=PTYPE
      PROC1=PT(IPTYPE,1)*BATCH
      PROC2=PT(IPTYPE,2)*BATCH
      PROC3=PT(IPTYPE,3)*BATCH
      PROC4=PT(IPTYPE,4)*BATCH
      PROC5=PT(IPTYPE,5)*BATCH
      RETURN
      END
C
C************************************************************
C*             FUNCTION USERF                               *
C*             -- CALCULATE NET PRESENT VALUE               *
C*                (PER ORDER)                               *
C************************************************************
C
      FUNCTION USERF(N)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
C
      EQUIVALENCE(ATRIB(1),RELDATE),(ATRIB(2),SUBFAM)
      EQUIVALENCE(ATRIB(3),PTYPE),(ATRIB(4),INTARRT)
      EQUIVALENCE(ATRIB(5),BATCH),(ATRIB(6),PROC1)
      EQUIVALENCE(ATRIB(7),PROC2),(ATRIB(8),PROC3)
      EQUIVALENCE(ATRIB(9),PROC4),(ATRIB(10),PROC5)
      EQUIVALENCE(ATRIB(11),STP1),(ATRIB(12),STP2)
      EQUIVALENCE(ATRIB(13),STP3),(ATRIB(14),STP4)
      EQUIVALENCE(ATRIB(15),STP5),(ATRIB(16),DISPAT)
```

```
      EQUIVALENCE(ATRIB(17),DELAY),(ATRIB(18),DUEDATE)
      EQUIVALENCE(ATRIB(19),TWC1),(ATRIB(20),TWC2)
      EQUIVALENCE(ATRIB(21),TWC3),(ATRIB(22),TWC4)
      EQUIVALENCE(ATRIB(23),TCOMP),(ATRIB(24),TSHIP)
      EQUIVALENCE(ATRIB(25),JNPV),(ATRIB(26),CONTN)
      EQUIVALENCE(XX(1),GSH),(XX(2),AAW),(XX(3),DPV)
      EQUIVALENCE(XX(4),CTB)
      REAL INTARRT,JNPV
C
C***** THE PARAMETERS USED TO CALCULATE AN ORDER'S "NPV"
C      ARE ADOPTED FROM A PAPER (ROHLEDER AND SCUDDER 1992).
C       R: ANNUAL INTEREST RATE (CONTINUOUS COMPOUNDING)
C       H: ANNUAL HOLDING COST RATE (CONTINUOUS COMPOUNDING)
C       PI: UNIT PENALTY COST (% OF REVENUE PER YEAR)
C       F: PROFIT MARGIN
C       UPRO: UNIT PROCESSING COST ($ PER HOUR)
C       VSTP: UNIT SETUP COST ($ PER HOUR)
C
      R=XX(56)
      H=XX(57)
      PI=XX(58)
      F=XX(59)
      UPRO=XX(60)
      VSTP=XX(61)
C
C***** UNIT MATERIAL COST ($ PER PART):
C           SAMPLING FROM U(50,100)
C       WMAT = (UNIT MATERIAL COST) * BATCH
C
      IF(SUBFAM.EQ.1) WMAT=89.0*BATCH
      IF(SUBFAM.EQ.2) WMAT=74.0*BATCH
      IF(SUBFAM.EQ.3) WMAT=65.0*BATCH
      IF(SUBFAM.EQ.4) WMAT=51.0*BATCH
      IF(SUBFAM.EQ.5) WMAT=77.0*BATCH
      IF(SUBFAM.EQ.6) WMAT=96.0*BATCH
C
      GO TO (100,200),N
C
 100  AWC1=(TWC1-PROC1-STP1-RELDATE)/120000.0
      AWC2=(TWC2-PROC2-STP2-RELDATE)/120000.0
      AWC3=(TWC3-PROC3-STP3-RELDATE)/120000.0
      AWC4=(TWC4-PROC4-STP4-RELDATE)/120000.0
      ACOMP=(TCOMP-PROC5-STP5-RELDATE)/120000.0
      ADUEDATE=(DUEDATE-RELDATE)/120000.0
      CWC1=(PROC1*UPRO+STP1*VSTP)/60.0
      CWC2=(PROC2*UPRO+STP2*VSTP)/60.0
      CWC3=(PROC3*UPRO+STP3*VSTP)/60.0
      CWC4=(PROC4*UPRO+STP4*VSTP)/60.0
      CWC5=(PROC5*UPRO+STP5*VSTP)/60.0
C
C***** PV1
C
      PV1=WMAT+CWC1*EXP(-R*AWC1)+CWC2*EXP(-R*AWC2)
```

```
     1         +CWC3*EXP(-R*AWC3)+CWC4*EXP(-R*AWC4)
     2         +CWC5*EXP(-R*ACOMP)
C
C***** PV2
C
      PV2=(
     1 (WMAT*EXP(H*ADUEDATE)-WMAT)
     2+(CWC1*EXP(H*(DUEDATE-TWC1-PROC1-STP1)/120000.)-CWC1)
     3+(CWC2*EXP(H*(DUEDATE-TWC2-PROC2-STP2)/120000.)-CWC2)
     4+(CWC3*EXP(H*(DUEDATE-TWC3-PROC3-STP3)/120000.)-CWC3)
     5+(CWC4*EXP(H*(DUEDATE-TWC4-PROC4-STP4)/120000.)-CWC4)
     6+(CWC5*EXP(H*(DUEDATE-TCOMP-PROC5-STP5)/120000.)-CWC5)
     7    )*EXP(-R*ADUEDATE)
C
C***** A JOB'S "NPV" (PV1 & PV2 ONLY) EQUALS:
C
      USERF=-PV1-PV2
      RETURN
C
 200  ASHIP=(TSHIP-RELDATE)/120000.0
C
C***** PV3
C
C***** TUCOST: TOTAL UNDISCOUNTED COST OF A ORDER
C      TUREVE: TOTAL UNDISCOUNTED REVENUE OF A ORDER
C                   = (1+PROF_MARG) * COST
C      TARDINESS PENALTY OF A ORDER
C                   = PI * REVENUE * TARDINESS
C
      TUCOST=WMAT+(PROC1+PROC2+PROC3+PROC4+PROC5)/60.*UPRO
      TUREVE=(1.0+F)*TUCOST
      IF(TCOMP.GT.(DUEDATE+240)) THEN
        PV3=PI*TUREVE*(TCOMP-(DUEDATE+240))/120000.*
     1      EXP(-R*ASHIP)
      ELSE
        PV3=0
      ENDIF
C
C
C***** PV4
C
      PV4=TUREVE*EXP(-R*ASHIP)
C
C***** TO SUM UP, AN ORDER'S "NPV" CAN BE CALCULATED AS:
C
      USERF=JNPV-PV3+PV4
      RETURN
      END
C
```

```
C************************************************************
C*            FUNCTION NQS                                  *
C*            -- APPLY ONE OF THE FIVE GROUP SCHEDULING     *
C*               HEURISTICS, GSH = 1: ADD/EDD               *
C*                                 2: ACR/CR                *
C*                                 3: ASLK/SLK              *
C*                                 4: NEQA/EQ               *
C*                                 5: NLQB/CR               *
C************************************************************
C
      FUNCTION NQS(N)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/AD(30),AOH(30),AOL(30),PT(30,5),FST(6,5)
C
      COMMON QSET(1000000)
      DIMENSION NSET(1000000)
      EQUIVALENCE(NSET(1),QSET(1))
C
      EQUIVALENCE(ATRIB(1),RELDATE),(ATRIB(2),SUBFAM)
      EQUIVALENCE(ATRIB(3),PTYPE),(ATRIB(4),INTARRT)
      EQUIVALENCE(ATRIB(5),BATCH),(ATRIB(6),PROC1)
      EQUIVALENCE(ATRIB(7),PROC2),(ATRIB(8),PROC3)
      EQUIVALENCE(ATRIB(9),PROC4),(ATRIB(10),PROC5)
      EQUIVALENCE(ATRIB(11),STP1),(ATRIB(12),STP2)
      EQUIVALENCE(ATRIB(13),STP3),(ATRIB(14),STP4)
      EQUIVALENCE(ATRIB(15),STP5),(ATRIB(16),DISPAT)
      EQUIVALENCE(ATRIB(17),DELAY),(ATRIB(18),DUEDATE)
      EQUIVALENCE(ATRIB(19),TWC1),(ATRIB(20),TWC2)
      EQUIVALENCE(ATRIB(21),TWC3),(ATRIB(22),TWC4)
      EQUIVALENCE(ATRIB(23),TCOMP),(ATRIB(24),TSHIP)
      EQUIVALENCE(ATRIB(25),JNPV),(ATRIB(26),CONTN)
      EQUIVALENCE(XX(1),GSH),(XX(2),AAW),(XX(3),DPV)
      EQUIVALENCE(XX(4),CTB)
      REAL INTARRT,JNPV
      DIMENSION A(6),AT(100)
C
C##### CHOOSE A QUEUE SELECTION RULE FROM FOLLOWING "NQSR":
C      NQSR = 5      7      9      12     13
C             ADD    ACR    ASLK   NEQA   NLQB
C
      IF(GSH.EQ.1.) NQSR=5
      IF(GSH.EQ.2.) NQSR=7
      IF(GSH.EQ.3.) NQSR=9
      IF(GSH.EQ.4.) NQSR=12
      IF(GSH.EQ.5.) NQSR=13
C
      GO TO (100,200,300,300,300,400,400,400,400,
     1       100,100,500,600),NQSR
C
```

```
C**************************************************
C        ADD QUEUE SELECTION RULE
C**************************************************
 300  CALL INITNQS(N,NQS,INDEX)
      IF(INDEX.EQ.1) RETURN
C
      GO TO (310,320,330,340,350,360,370,380,390),N
C
C***** WORKCENTER 1
 310  DO 315 I=1,3
        IF(NNQ(I).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 313 J=1,NNQ(I)
            CALL COPY(J,I,AT)
            A(I)=A(I)+AT(18)
 313      CONTINUE
          A(I)=A(I)/NNQ(I)
        ENDIF
 315  CONTINUE
      GO TO 399
C***** WORKCENTER 2
 320  DO 325 I=1,3
        IF(NNQ(I+3).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 323 J=1,NNQ(I+3)
            CALL COPY(J,I+3,AT)
            A(I)=A(I)+AT(18)
 323      CONTINUE
          A(I)=A(I)/NNQ(I+3)
        ENDIF
 325  CONTINUE
      GO TO 399
C***** WORKCENTER 3
 330  DO 335 I=1,3
        IF(NNQ(I+6).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 333 J=1,NNQ(I+6)
            CALL COPY(J,I+6,AT)
            A(I)=A(I)+AT(18)
 333      CONTINUE
          A(I)=A(I)/NNQ(I+6)
        ENDIF
 335  CONTINUE
      GO TO 399
C***** WORKCENTER 4
 340  DO 345 I=1,3
        IF(NNQ(I+9).EQ.0) THEN
```

```
            A(I)=9999999.
          ELSE
            A(I)=0.
            DO 343 J=1,NNQ(I+9)
              CALL COPY(J,I+9,AT)
              A(I)=A(I)+AT(18)
 343        CONTINUE
            A(I)=A(I)/NNQ(I+9)
          ENDIF
 345    CONTINUE
        GO TO 399
C***** WORKCENTER 5
 350    DO 355 I=1,3
          IF(NNQ(I+12).EQ.0) THEN
            A(I)=9999999.
          ELSE
            A(I)=0.
            DO 353 J=1,NNQ(I+12)
              CALL COPY(J,I+12,AT)
              A(I)=A(I)+AT(18)
 353        CONTINUE
            A(I)=A(I)/NNQ(I+12)
          ENDIF
 355    CONTINUE
        GO TO 399
C***** WORKCENTER 6
 360    DO 365 I=1,3
          IF(NNQ(I+15).EQ.0) THEN
            A(I)=9999999.
          ELSE
            A(I)=0.
            DO 363 J=1,NNQ(I+15)
              CALL COPY(J,I+15,AT)
              A(I)=A(I)+AT(18)
 363        CONTINUE
            A(I)=A(I)/NNQ(I+15)
          ENDIF
 365    CONTINUE
        GO TO 399
C***** WORKCENTER 7
 370    DO 375 I=1,3
          IF(NNQ(I+18).EQ.0) THEN
            A(I)=9999999.
          ELSE
            A(I)=0.
            DO 373 J=1,NNQ(I+18)
              CALL COPY(J,I+18,AT)
              A(I)=A(I)+AT(18)
 373        CONTINUE
            A(I)=A(I)/NNQ(I+18)
          ENDIF
 375    CONTINUE
        GO TO 399
```

```
C***** WORKCENTER 8
 380  DO 385 I=1,3
        IF(NNQ(I+21).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 383 J=1,NNQ(I+21)
            CALL COPY(J,I+21,AT)
            A(I)=A(I)+AT(18)
 383      CONTINUE
          A(I)=A(I)/NNQ(I+21)
        ENDIF
 385  CONTINUE
      GO TO 399
C***** WORKCENTER 9
 390  DO 395 I=1,6
        IF(NNQ(I+24).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 393 J=1,NNQ(I+24)
            CALL COPY(J,I+24,AT)
            A(I)=A(I)+AT(18)
 393      CONTINUE
          A(I)=A(I)/NNQ(I+24)
        ENDIF
 395  CONTINUE
C
C
 399  CALL SELNQS(N,NQS,A)
      CALL SELJOB(N,NQS)
      RETURN
C
C***************************************************
C        ACR,ASLK QUEUE SELECTION RULES
C***************************************************
 400  CALL INITNQS(N,NQS,INDEX)
      IF(INDEX.EQ.1) RETURN
C
      GO TO (410,420,430,440,450,460,470,480,490),N
C
C***** WORKCENTER 1
 410  DO 415 I=1,3
        IF(NNQ(I).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 413 J=1,NNQ(I)
            CALL COPY(J,I,AT)
            IF(NQSR.EQ.7) THEN
              DISP=(AT(18)-TNOW)/
     1            (AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
            ELSEIF(NQSR.EQ.9) THEN
```

```
              DISP=(AT(18)-TNOW)-
     1           (AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
            ENDIF
            A(I)=A(I)+DISP
 413      CONTINUE
          A(I)=A(I)/NNQ(I)
        ENDIF
 415  CONTINUE
      GO TO 499
C***** WORKCENTER 2
 420  DO 425 I=1,3
        IF(NNQ(I+3).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 423 J=1,NNQ(I+3)
            CALL COPY(J,I+3,AT)
            IF(NQSR.EQ.7) THEN
              DISP=(AT(18)-TNOW)/(AT(7)+AT(8)+AT(9)+AT(10))
            ELSEIF(NQSR.EQ.9) THEN
              DISP=(AT(18)-TNOW)-(AT(7)+AT(8)+AT(9)+AT(10))
            ENDIF
            A(I)=A(I)+DISP
 423      CONTINUE
          A(I)=A(I)/NNQ(I+3)
        ENDIF
 425  CONTINUE
      GO TO 499
C***** WORKCENTER 3
 430  DO 435 I=1,3
        IF(NNQ(I+6).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 433 J=1,NNQ(I+6)
            CALL COPY(J,I+6,AT)
            IF(NQSR.EQ.7) THEN
              DISP=(AT(18)-TNOW)/(AT(8)+AT(9)+AT(10))
            ELSEIF(NQSR.EQ.9) THEN
              DISP=(AT(18)-TNOW)-(AT(8)+AT(9)+AT(10))
            ENDIF
            A(I)=A(I)+DISP
 433      CONTINUE
          A(I)=A(I)/NNQ(I+6)
        ENDIF
 435  CONTINUE
      GO TO 499
C***** WORKCENTER 4
 440  DO 445 I=1,3
        IF(NNQ(I+9).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
```

```
            DO 443 J=1,NNQ(I+9)
              CALL COPY(J,I+9,AT)
              IF(NQSR.EQ.7) THEN
                DISP=(AT(18)-TNOW)/(AT(9)+AT(10))
              ELSEIF(NQSR.EQ.9) THEN
                DISP=(AT(18)-TNOW)-(AT(9)+AT(10))
              ENDIF
              A(I)=A(I)+DISP
 443        CONTINUE
            A(I)=A(I)/NNQ(I+9)
          ENDIF
 445    CONTINUE
        GO TO 499
C***** WORKCENTER 5
 450    DO 455 I=1,3
          IF(NNQ(I+12).EQ.0) THEN
            A(I)=9999999.
          ELSE
            A(I)=0.
            DO 453 J=1,NNQ(I+12)
              CALL COPY(J,I+12,AT)
              IF(NQSR.EQ.7) THEN
                DISP=(AT(18)-TNOW)/
     1              (AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
              ELSEIF(NQSR.EQ.9) THEN
                DISP=(AT(18)-TNOW)-
     1              (AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
              ENDIF
              A(I)=A(I)+DISP
 453        CONTINUE
            A(I)=A(I)/NNQ(I+12)
          ENDIF
 455    CONTINUE
        GO TO 499
C***** WORKCENTER 6
 460    DO 465 I=1,3
          IF(NNQ(I+15).EQ.0) THEN
            A(I)=9999999.
          ELSE
            A(I)=0.
            DO 463 J=1,NNQ(I+15)
              CALL COPY(J,I+15,AT)
              IF(NQSR.EQ.7) THEN
                DISP=(AT(18)-TNOW)/(AT(7)+AT(8)+AT(9)+AT(10))
              ELSEIF(NQSR.EQ.9) THEN
                DISP=(AT(18)-TNOW)-(AT(7)+AT(8)+AT(9)+AT(10))
              ENDIF
              A(I)=A(I)+DISP
 463        CONTINUE
            A(I)=A(I)/NNQ(I+15)
          ENDIF
 465    CONTINUE
        GO TO 499
```

```
C***** WORKCENTER 7
 470  DO 475 I=1,3
        IF(NNQ(I+18).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 473 J=1,NNQ(I+18)
            CALL COPY(J,I+18,AT)
            IF(NQSR.EQ.7) THEN
              DISP=(AT(18)-TNOW)/(AT(8)+AT(9)+AT(10))
            ELSEIF(NQSR.EQ.9)
              THEN DISP=(AT(18)-TNOW)-(AT(8)+AT(9)+AT(10))
            ENDIF
            A(I)=A(I)+DISP
 473      CONTINUE
          A(I)=A(I)/NNQ(I+18)
        ENDIF
 475  CONTINUE
      GO TO 499
C***** WORKCENTER 8
 480  DO 485 I=1,3
        IF(NNQ(I+21).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 483 J=1,NNQ(I+21)
            CALL COPY(J,I+21,AT)
            IF(NQSR.EQ.7) THEN
              DISP=(AT(18)-TNOW)/(AT(9)+AT(10))
            ELSEIF(NQSR.EQ.9) THEN
              DISP=(AT(18)-TNOW)-(AT(9)+AT(10))
            ENDIF
            A(I)=A(I)+DISP
 483      CONTINUE
          A(I)=A(I)/NNQ(I+21)
        ENDIF
 485  CONTINUE
      GO TO 499
C***** WORKCENTER 9
 490  DO 495 I=1,6
        IF(NNQ(I+24).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.
          DO 493 J=1,NNQ(I+24)
            CALL COPY(J,I+24,AT)
            IF(NQSR.EQ.7) THEN
              DISP=(AT(18)-TNOW)/AT(10)
            ELSEIF(NQSR.EQ.9) THEN
              DISP=(AT(18)-TNOW)-AT(10)
            ENDIF
            A(I)=A(I)+DISP
 493      CONTINUE
```

```
          A(I)=A(I)/NNQ(I+24)
        ENDIF
 495  CONTINUE
C
C
 499  CALL SELNQS(N,NQS,A)
      CALL SELJOB(N,NQS)
      RETURN
C
C***************************************************
C        NEQA QUEUE SELECTION RULE
C***************************************************
 500  CALL INITNQS(N,NQS,INDEX)
      IF(INDEX.EQ.1) RETURN
C
      GO TO (510,520,530,540,550,560,570,580,590),N
C
C***** WORKCENTER 1
 510  DO 515 I=1,3
        IF(NNQ(I).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+3).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+6).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+9).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+24).EQ.0) A(I)=A(I)+1
          A(I)=(NNQ(I+3)+NNQ(I+6)+NNQ(I+9)+NNQ(24))/A(I)
        ENDIF
 515  CONTINUE
      GO TO 599
C***** WORKCENTER 2
 520  DO 525 I=1,3
        IF(NNQ(I+3).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+6).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+9).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+24).EQ.0) A(I)=A(I)+1
          A(I)=(NNQ(I+6)+NNQ(I+9)+NNQ(24))/A(I)
        ENDIF
 525  CONTINUE
      GO TO 599
C***** WORKCENTER 3
 530  DO 535 I=1,3
        IF(NNQ(I+6).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+9).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+24).EQ.0) A(I)=A(I)+1
          A(I)=(NNQ(I+9)+NNQ(24))/A(I)
```

```
        ENDIF
 535  CONTINUE
      GO TO 599
C***** WORKCENTER 4
 540  DO 545 I=1,3
        IF(NNQ(I+9).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+24).EQ.0) A(I)=A(I)+1
          A(I)=NNQ(24)/A(I)
        ENDIF
 545  CONTINUE
      GO TO 599
C***** WORKCENTER 5
 550  DO 555 I=1,3
        IF(NNQ(I+12).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+15).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+18).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+21).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+27).EQ.0) A(I)=A(I)+1
          A(I)=(NNQ(I+15)+NNQ(I+18)+NNQ(I+21)+NNQ(27))/A(I)
        ENDIF
 555  CONTINUE
      GO TO 599
C***** WORKCENTER 6
 560  DO 565 I=1,3
        IF(NNQ(I+15).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+18).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+21).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+27).EQ.0) A(I)=A(I)+1
          A(I)=(NNQ(I+18)+NNQ(I+21)+NNQ(27))/A(I)
        ENDIF
 565  CONTINUE
      GO TO 599
C***** WORKCENTER 7
 570  DO 575 I=1,3
        IF(NNQ(I+18).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+21).EQ.0) A(I)=A(I)+1
          IF(NNQ(I+27).EQ.0) A(I)=A(I)+1
          A(I)=(NNQ(I+21)+NNQ(27))/A(I)
        ENDIF
 575  CONTINUE
      GO TO 599
```

```
C***** WORKCENTER 8
 580  DO 585 I=1,3
        IF(NNQ(I+21).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0.1
          IF(NNQ(I+27).EQ.0) A(I)=A(I)+1
          A(I)=NNQ(27)/A(I)
        ENDIF
 585  CONTINUE
      GO TO 599
C***** WORKCENTER 9
 590  DO 595 I=1,6
        IF(NNQ(I+24).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=1./(NNQ(I+24)+0.1)
        ENDIF
 595  CONTINUE
C
C
 599  CALL SELNQS(N,NQS,A)
      CALL SELJOB(N,NQS)
      RETURN
C
C***************************************************
C        NLQB QUEUE SELECTION RULE
C***************************************************
 600  CALL INITNQS(N,NQS,INDEX)
      IF(INDEX.EQ.1) RETURN
      GO TO (610,620,630,640,650,660,670,680,690),N
C
C***** WORKCENTER 1
 610  DO 615 I=1,3
        IF(NNQ(I).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=1./(NNQ(I)+0.1)
        ENDIF
 615  CONTINUE
      GO TO 699
C***** WORKCENTER 2
 620  DO 625 I=1,3
        IF(NNQ(I+3).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0
          IF(NNQ(I).GE.XX(53)) A(I)=A(I)+1
          A(I)=A(I)*NNQ(I)
          A(I)=1./(A(I)+0.1)
        ENDIF
 625  CONTINUE
      GO TO 699
```

```
C***** WORKCENTER 3
 630  DO 635 I=1,3
        IF(NNQ(I+6).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0
          IF(NNQ(I).GE.XX(53)) A(I)=A(I)+1
          IF(NNQ(I+3).GE.XX(53)) A(I)=A(I)+1
          A(I)=A(I)*(NNQ(I)+NNQ(I+3))
          A(I)=1./(A(I)+0.1)
        ENDIF
 635  CONTINUE
      GO TO 699
C***** WORKCENTER 4
 640  DO 645 I=1,3
        IF(NNQ(I+9).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0
          IF(NNQ(I).GE.XX(53)) A(I)=A(I)+1
          IF(NNQ(I+3).GE.XX(53)) A(I)=A(I)+1
          IF(NNQ(I+6).GE.XX(53)) A(I)=A(I)+1
          A(I)=A(I)*(NNQ(I)+NNQ(I+3)+NNQ(I+6))
          A(I)=1./(A(I)+0.1)
        ENDIF
 645  CONTINUE
      GO TO 699
C***** WORKCENTER 5
 650  DO 655 I=1,3
        IF(NNQ(I+12).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=1./(NNQ(I+12)+0.1)
        ENDIF
 655  CONTINUE
      GO TO 699
C***** WORKCENTER 6
 660  DO 665 I=1,3
        IF(NNQ(I+15).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0
          IF(NNQ(I+12).GE.XX(53)) A(I)=A(I)+1
          A(I)=A(I)*NNQ(I+12)
          A(I)=1./(A(I)+0.1)
        ENDIF
 665  CONTINUE
      GO TO 699
C***** WORKCENTER 7
 670  DO 675 I=1,3
        IF(NNQ(I+18).EQ.0) THEN
          A(I)=9999999.
        ELSE
```

```
          A(I)=0
          IF(NNQ(I+12).GE.XX(53)) A(I)=A(I)+1
          IF(NNQ(I+15).GE.XX(53)) A(I)=A(I)+1
          A(I)=A(I)*(NNQ(I+12)+NNQ(I+15))
          A(I)=1./(A(I)+0.1)
        ENDIF
 675  CONTINUE
      GO TO 699
C***** WORKCENTER 8
 680  DO 685 I=1,3
        IF(NNQ(I+21).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0
          IF(NNQ(I+12).GE.XX(53)) A(I)=A(I)+1
          IF(NNQ(I+15).GE.XX(53)) A(I)=A(I)+1
          IF(NNQ(I+18).GE.XX(53)) A(I)=A(I)+1
          A(I)=A(I)*(NNQ(I+12)+NNQ(I+15)+NNQ(I+18))
          A(I)=1./(A(I)+0.1)
        ENDIF
 685  CONTINUE
      GO TO 699
C***** WORKCENTER 9
 690  DO 695 I=1,6
        IF(NNQ(I+24).EQ.0) THEN
          A(I)=9999999.
        ELSE
          A(I)=0
          IF(I.LE.3) THEN
            IF(NNQ(I).GE.XX(53)) A(I)=A(I)+1
            IF(NNQ(I+3).GE.XX(53)) A(I)=A(I)+1
            IF(NNQ(I+6).GE.XX(53)) A(I)=A(I)+1
            IF(NNQ(I+9).GE.XX(53)) A(I)=A(I)+1
            A(I)=A(I)*(NNQ(I)+NNQ(I+3)+NNQ(I+6)+NNQ(I+9))
          ELSE
            IF(NNQ(I+12).GE.XX(53)) A(I)=A(I)+1
            IF(NNQ(I+15).GE.XX(53)) A(I)=A(I)+1
            IF(NNQ(I+18).GE.XX(53)) A(I)=A(I)+1
            IF(NNQ(I+21).GE.XX(53)) A(I)=A(I)+1
            A(I)=A(I)*(NNQ(I+12)+NNQ(I+15)+NNQ(I+18)+
     1                 NNQ(I+21))
          ENDIF
          A(I)=1./(A(I)+0.1)
        ENDIF
 695  CONTINUE
C
C
 699  CALL SELNQS(N,NQS,A)
      CALL SELJOB(N,NQS)
      RETURN
      END
C
```

```
C************************************************************
C*             SUBROUTINE INITNQS                           *
C*             -- CALLED BY FUNCTION "NQS"                  *
C*                FOR INITIAL CHECKING                      *
C************************************************************
C
      SUBROUTINE INITNQS(N,NQS,INDEX)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
C
      NQS=0
      INDEX=1
      GO TO (10,20,30,40,50,60,70,80,90),N
C
C***** WORKCENTER 1
   10 IF(NNQ(1).EQ.0.AND.NNQ(2).EQ.0.AND.NNQ(3).EQ.0) RETURN
      DO 15 I=1,3
        IF(XX(41).EQ.I.AND.NNQ(I).GT.0) THEN
          NQS=I
          XX(31)=XX(31)+1
          IF(XX(31).GT.XX(53)) GO TO 15
          RETURN
        ENDIF
   15 CONTINUE
      INDEX=2
      XX(31)=1
      RETURN
C***** WORKCENTER 2
   20 IF(NNQ(4).EQ.0.AND.NNQ(5).EQ.0.AND.NNQ(6).EQ.0)RETURN
      DO 25 I=1,3
        IF(XX(42).EQ.I.AND.NNQ(I+3).GT.0) THEN
          NQS=I+3
          XX(32)=XX(32)+1
          IF(XX(32).GT.XX(53)) GO TO 25
          RETURN
        ENDIF
   25 CONTINUE
      INDEX=2
      XX(32)=1
      RETURN
C***** WORKCENTER 3
   30 IF(NNQ(7).EQ.0.AND.NNQ(8).EQ.0.AND.NNQ(9).EQ.0)RETURN
      DO 35 I=1,3
        IF(XX(43).EQ.I.AND.NNQ(I+6).GT.0) THEN
          NQS=I+6
          XX(33)=XX(33)+1
          IF(XX(33).GT.XX(53)) GO TO 35
          RETURN
        ENDIF
   35 CONTINUE
      INDEX=2
      XX(33)=1
```

```
      RETURN
C***** WORKCENTER 4
   40 IF(NNQ(10).EQ.0.AND.NNQ(11).EQ.0.AND.NNQ(12).EQ.0)RETURN
      DO 45 I=1,3
        IF(XX(44).EQ.I.AND.NNQ(I+9).GT.0) THEN
          NQS=I+9
          XX(34)=XX(34)+1
          IF(XX(34).GT.XX(53)) GO TO 45
          RETURN
        ENDIF
   45 CONTINUE
      INDEX=2
      XX(34)=1
      RETURN
C***** WORKCENTER 5
   50 IF(NNQ(13).EQ.0.AND.NNQ(14).EQ.0.AND.NNQ(15).EQ.0)RETURN
      DO 55 I=1,3
        IF(XX(45).EQ.I.AND.NNQ(I+12).GT.0) THEN
          NQS=I+12
          XX(35)=XX(35)+1
          IF(XX(35).GT.XX(53)) GO TO 55
          RETURN
        ENDIF
   55 CONTINUE
      INDEX=2
      XX(35)=1
      RETURN
C***** WORKCENTER 6
   60 IF(NNQ(16).EQ.0.AND.NNQ(17).EQ.0.AND.NNQ(18).EQ.0)RETURN
      DO 65 I=1,3
        IF(XX(46).EQ.I.AND.NNQ(I+15).GT.0) THEN
          NQS=I+15
          XX(36)=XX(36)+1
          IF(XX(36).GT.XX(53)) GO TO 65
          RETURN
        ENDIF
   65 CONTINUE
      INDEX=2
      XX(36)=1
      RETURN
C***** WORKCENTER 7
   70 IF(NNQ(19).EQ.0.AND.NNQ(20).EQ.0.AND.NNQ(21).EQ.0)RETURN
      DO 75 I=1,3
        IF(XX(47).EQ.I.AND.NNQ(I+18).GT.0) THEN
          NQS=I+18
          XX(37)=XX(37)+1
          IF(XX(37).GT.XX(53)) GO TO 75
          RETURN
        ENDIF
   75 CONTINUE
      INDEX=2
      XX(37)=1
      RETURN
```

```
C***** WORKCENTER 8
  80  IF(NNQ(22).EQ.0.AND.NNQ(23).EQ.0.AND.NNQ(24).EQ.0)RETURN
      DO 85 I=1,3
        IF(XX(48).EQ.I.AND.NNQ(I+21).GT.0) THEN
          NQS=I+21
          XX(38)=XX(38)+1
          IF(XX(38).GT.XX(53)) GO TO 85
          RETURN
        ENDIF
  85  CONTINUE
      INDEX=2
      XX(38)=1
      RETURN
C***** WORKCENTER 9
  90  IF(NNQ(25).EQ.0.AND.NNQ(26).EQ.0.AND.NNQ(27).EQ.0.AND.
     1 NNQ(28).EQ.0.AND.NNQ(29).EQ.0.AND.NNQ(30).EQ.0)RETURN
      DO 95 I=1,6
        IF(XX(49).EQ.I.AND.NNQ(I+24).GT.0) THEN
          NQS=I+24
          XX(39)=XX(39)+1
          IF(XX(39).GT.XX(53)) GO TO 95
          RETURN
        ENDIF
  95  CONTINUE
      INDEX=2
      XX(39)=1
      RETURN
      END
C
C************************************************************
C*           SUBROUTINE SELNQS                             *
C*           -- CALLED BY FUNCTION "NQS"                   *
C*              FOR SELECTING A QUEUE                      *
C************************************************************
C
      SUBROUTINE SELNQS(N,NQS,A)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      DIMENSION A(6)
      EQUIVALENCE(XX(1),GSH)
C
      GO TO (10,20,30,40,50,60,70,80,90),N
C
C***** WORKCENTER 1
  10  CALL COMPA(N,A,MINA)
      NQS=MINA
      XX(41)=MINA
      RETURN
C***** WORKCENTER 2
  20  CALL COMPA(N,A,MINA)
      NQS=MINA+3
      XX(42)=MINA
```

```
      RETURN
C***** WORKCENTER 3
  30  CALL COMPA(N,A,MINA)
      NQS=MINA+6
      XX(43)=MINA
      RETURN
C***** WORKCENTER 4
  40  CALL COMPA(N,A,MINA)
      NQS=MINA+9
      XX(44)=MINA
      RETURN
C***** WORKCENTER 5
  50  CALL COMPA(N,A,MINA)
      NQS=MINA+12
      XX(45)=MINA+3
      RETURN
C***** WORKCENTER 6
  60  CALL COMPA(N,A,MINA)
      NQS=MINA+15
      XX(46)=MINA+3
      RETURN
C***** WORKCENTER 7
  70  CALL COMPA(N,A,MINA)
      NQS=MINA+18
      XX(47)=MINA+3
      RETURN
C***** WORKCENTER 8
  80  CALL COMPA(N,A,MINA)
      NQS=MINA+21
      XX(48)=MINA+3
      RETURN
C***** WORKCENTER 9
  90  CALL COMPA(N,A,MINA)
      NQS=MINA+24
      XX(49)=MINA
      RETURN
      END
C
C************************************************************
C*             SUBROUTINE COMPA                             *
C*             -- CALLED BY SUBROUTINE "SELNQS"             *
C*                FOR FINDIND A MIN A()                     *
C*                AND DEFINING CELL SELECTION RULES         *
C************************************************************
C
      SUBROUTINE COMPA(N,A,MINA)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      DIMENSION A(6)
      EQUIVALENCE(XX(1),GSH)
C
C***** DEFINE CELL SELECTION RULES FOR W.C. 5 (SHARED W.C.)
```

```
C      NCSR = 1: NONE
C             2: LARGEST # OF JOBS IN A CELL
C
      NCSR=2
C
      IF(N.LT.9) GO TO 30
      IF(NCSR.EQ.1) GO TO 50
      IF(NCSR.EQ.2) THEN
        IF(XX(21).GE.XX(22)) GO TO 70
        IF(XX(21).LT.XX(22)) GO TO 90
      ENDIF
C
  30  IF(A(1).LE.A(2).AND.A(1).LE.A(3)) MINA=1
      IF(A(2).LE.A(1).AND.A(2).LE.A(3)) MINA=2
      IF(A(3).LE.A(1).AND.A(3).LE.A(2)) MINA=3
      RETURN
C
  50  IF(A(1).LE.A(2).AND.A(1).LE.A(3).AND.A(1).LE.A(4)
     1     .AND.A(1).LE.A(5).AND.A(1).LE.A(6)) MINA=1
        IF(A(2).LE.A(1).AND.A(2).LE.A(3).AND.A(2).LE.A(4)
     1     .AND.A(2).LE.A(5).AND.A(2).LE.A(6)) MINA=2
        IF(A(3).LE.A(1).AND.A(3).LE.A(2).AND.A(3).LE.A(4)
     1     .AND.A(3).LE.A(5).AND.A(3).LE.A(6)) MINA=3
        IF(A(4).LE.A(1).AND.A(4).LE.A(2).AND.A(4).LE.A(3)
     1     .AND.A(4).LE.A(5).AND.A(4).LE.A(6)) MINA=4
        IF(A(5).LE.A(1).AND.A(5).LE.A(2).AND.A(5).LE.A(3)
     1     .AND.A(5).LE.A(4).AND.A(5).LE.A(6)) MINA=5
        IF(A(6).LE.A(1).AND.A(6).LE.A(2).AND.A(6).LE.A(3)
     1     .AND.A(6).LE.A(4).AND.A(6).LE.A(5)) MINA=6
        RETURN
C
  70  IF(NNQ(25).EQ.0.AND.NNQ(26).EQ.0.AND.NNQ(27).EQ.0.)
     1  GO TO 90
      IF(A(1).LE.A(2).AND.A(1).LE.A(3)) MINA=1
      IF(A(2).LE.A(1).AND.A(2).LE.A(3)) MINA=2
      IF(A(3).LE.A(1).AND.A(3).LE.A(2)) MINA=3
      RETURN
C
  90  IF(NNQ(28).EQ.0.AND.NNQ(29).EQ.0.AND.NNQ(30).EQ.0)
     1  GO TO 70
      IF(A(4).LE.A(5).AND.A(4).LE.A(6)) MINA=4
      IF(A(5).LE.A(4).AND.A(5).LE.A(6)) MINA=5
      IF(A(6).LE.A(4).AND.A(6).LE.A(5)) MINA=6
      RETURN
      END
C
C***********************************************************
C*            SUBROUTINE SELJOB                            *
C*            -- CALLED BY FUNCTION "NQS"                  *
C*               FOR CHOOSING ONE OF THE JOB               *
C*               DISPATCHING RULES                         *
C***********************************************************
C
```

```
      SUBROUTINE SELJOB(N,NQS)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1  MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),
     2  SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/AD(30),AOH(30),AOL(30),PT(30,5),FST(6,5)
C
      COMMON QSET(1000000)
      DIMENSION NSET(1000000)
      EQUIVALENCE(NSET(1),QSET(1))
C
      EQUIVALENCE(ATRIB(1),RELDATE),(ATRIB(2),SUBFAM)
      EQUIVALENCE(ATRIB(3),PTYPE),(ATRIB(4),INTARRT)
      EQUIVALENCE(ATRIB(5),BATCH),(ATRIB(6),PROC1)
      EQUIVALENCE(ATRIB(7),PROC2),(ATRIB(8),PROC3)
      EQUIVALENCE(ATRIB(9),PROC4),(ATRIB(10),PROC5)
      EQUIVALENCE(ATRIB(11),STP1),(ATRIB(12),STP2)
      EQUIVALENCE(ATRIB(13),STP3),(ATRIB(14),STP4)
      EQUIVALENCE(ATRIB(15),STP5),(ATRIB(16),DISPAT)
      EQUIVALENCE(ATRIB(17),DELAY),(ATRIB(18),DUEDATE)
      EQUIVALENCE(ATRIB(19),TWC1),(ATRIB(20),TWC2)
      EQUIVALENCE(ATRIB(21),TWC3),(ATRIB(22),TWC4)
      EQUIVALENCE(ATRIB(23),TCOMP),(ATRIB(24),TSHIP)
      EQUIVALENCE(ATRIB(25),JNPV),(ATRIB(26),CONTN)
      EQUIVALENCE(XX(1),GSH),(XX(2),AAW),(XX(3),DPV)
      EQUIVALENCE(XX(4),CTB)
      REAL INTARRT,JNPV
      DIMENSION AT(100)
C
C##### CHOOSE A JOB DISPATCHING RULE FROM FOLLOWING "NJDR":
C       NJDR = 3      5     6     7
C              EDD    CR    SLK   EQ
C
      IF(GSH.EQ.1.) NJDR=3
      IF(GSH.EQ.2.) NJDR=5
      IF(GSH.EQ.3.) NJDR=6
      IF(GSH.EQ.4.) NJDR=7
      IF(GSH.EQ.5.) NJDR=5
C
C*************************************************
C         SPT/EDD/SI/CR/SLK/EQ JOB DISPATCHING RULES
C*************************************************
C*** FILE 31 IS USED AS A BUFFER TO SORT ENTITIES IN
C     A QUEUE FILE BASED ON A JOB DISPATCHING RULE.
C
   50 IF(NNQ(31).EQ.0) GO TO 100
      DO 60 I=1,NNQ(31)
        CALL RMOVE(I,31,AT)
   60 CONTINUE
C
  100 GO TO (110,120,130,140,150,160,170,180,190),N
C
C***** WORKCENTER 1
C
```

```
 110  DO 112 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
C
C*** EDD RULE
        IF(NJDR.EQ.3) DISP=AT(18)
C*** CR RULE
        IF(NJDR.EQ.5)
     1   DISP=(AT(18)-TNOW)/(AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
C*** SLK RULE
        IF(NJDR.EQ.6)
     1   DISP=(AT(18)-TNOW)-(AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
C*** EQ RULE
        IF(NJDR.EQ.7) THEN
          DISP=1
          NEXT=0
          IF(AT(7).NE.0) THEN
            NEXT=1
          ELSEIF(AT(8).NE.0) THEN
            NEXT=2
          ELSEIF(AT(9).NE.0) THEN
            NEXT=3
          ELSEIF(AT(10).NE.0) THEN
            NEXT=8
          ENDIF
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
C*** SAVE AN ENTITY'S PRIORITY INFORM. (DISP) TO ATR(16)
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 112  CONTINUE
C*** SORT A QUEUE FILE (NQS) BY PRIORITY INFORM: ATR(16)
C         PROCESS: FILE(NQS) -> FILE(31) -> FILE (NQS)
C         HERE FILE 31 IS A BUFFER.
      DO 114 I=1,NNQ(NQS)
        NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 114
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 114  CONTINUE
      DO 116 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 116
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(11)=FST(NQS,1)
        CALL FFILE(NQS,AT)
 116  CONTINUE
      RETURN
C
C***** WORKCENTER 2
C
 120  DO 122 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
C
        IF(NJDR.EQ.3) DISP=AT(18)
```

```
        IF(NJDR.EQ.5)
     1    DISP=(AT(18)-TNOW)/(AT(7)+AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.6)
     1    DISP=(AT(18)-TNOW)-(AT(7)+AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.7) THEN
          DISP=1
          NEXT=0
          IF(AT(8).NE.0) THEN
            NEXT=1
          ELSEIF(AT(9).NE.0) THEN
            NEXT=2
          ELSEIF(AT(10).NE.0) THEN
            NEXT=7
          ENDIF
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
C
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 122  CONTINUE
C
      DO 124 I=1,NNQ(NQS)
        NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 124
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 124  CONTINUE
      DO 126 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 126
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(12)=FST(NQS-3,2)
        CALL FFILE(NQS,AT)
 126  CONTINUE
      RETURN
C
C***** WORKCENTER 3
C
 130  DO 132 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
C
        IF(NJDR.EQ.3)DISP=AT(18)
        IF(NJDR.EQ.5)DISP=(AT(18)-TNOW)/(AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.6)DISP=(AT(18)-TNOW)-(AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.7) THEN
          DISP=1
          NEXT=0
          IF(AT(9).NE.0) THEN
            NEXT=1
          ELSEIF(AT(10).NE.0) THEN
            NEXT=6
          ENDIF
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
```

```
C
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 132  CONTINUE
C
      DO 134 I=1,NNQ(NQS)
        NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 134
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 134  CONTINUE
      DO 136 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 136
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(13)=FST(NQS-6,3)
        CALL FFILE(NQS,AT)
 136  CONTINUE
      RETURN
C
C***** WORKCENTER 4
C
 140  DO 142 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
        IF(NJDR.EQ.3) DISP=AT(18)
        IF(NJDR.EQ.5) DISP=(AT(18)-TNOW)/(AT(9)+AT(10))
        IF(NJDR.EQ.6) DISP=(AT(18)-TNOW)-(AT(9)+AT(10))
        IF(NJDR.EQ.7) THEN
          DISP=1
          NEXT=0
          IF(AT(10).NE.0) NEXT=5
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
C
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 142  CONTINUE
C
      DO 144 I=1,NNQ(NQS)
        NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 144
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 144  CONTINUE
      DO 146 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 146
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(14)=FST(NQS-9,4)
        CALL FFILE(NQS,AT)
 146  CONTINUE
      RETURN
C
C***** WORKCENTER 5
C
```

```
 150  DO 152 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
C
        IF(NJDR.EQ.3) DISP=AT(18)
        IF(NJDR.EQ.5)
     1   DISP=(AT(18)-TNOW)/(AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.6)
     1   DISP=(AT(18)-TNOW)-(AT(6)+AT(7)+AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.7) THEN
          DISP=1
          NEXT=0
          IF(AT(7).NE.0) THEN
            NEXT=1
          ELSEIF(AT(8).NE.0) THEN
            NEXT=2
          ELSEIF(AT(9).NE.0) THEN
            NEXT=3
          ELSEIF(AT(10).NE.0) THEN
            NEXT=5
          ENDIF
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
C
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 152  CONTINUE
C
      DO 154 I=1,NNQ(NQS)
        NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 154
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 154  CONTINUE
      DO 156 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 156
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(11)=FST(NQS-9,1)
        CALL FFILE(NQS,AT)
 156  CONTINUE
      RETURN
C
C***** WORKCENTER 6
C
 160  DO 162 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
C
        IF(NJDR.EQ.3) DISP=AT(18)
        IF(NJDR.EQ.5)
     1    DISP=(AT(18)-TNOW)/(AT(7)+AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.6)
     1    DISP=(AT(18)-TNOW)-(AT(7)+AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.7) THEN
          DISP=1
```

```
          NEXT=0
          IF(AT(8).NE.0) THEN
            NEXT=1
          ELSEIF(AT(9).NE.0) THEN
            NEXT=2
          ELSEIF(AT(10).NE.0) THEN
            NEXT=4
          ENDIF
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
C
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 162  CONTINUE
C
      DO 164 I=1,NNQ(NQS)
        NRA6K=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 164
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 164  CONTINUE
      DO 166 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 166
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(12)=FST(NQS-12,2)
        CALL FFILE(NQS,AT)
 166  CONTINUE
      RETURN
C
C***** WORKCENTER 7
C
 170  DO 172 I=1,NNQ(NQS)
        CALL COPY(I,NQS,AT)
C
        IF(NJDR.EQ.3)DISP=AT(18)
        IF(NJDR.EQ.5)DISP=(AT(18)-TNOW)/(AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.6)DISP=(AT(18)-TNOW)-(AT(8)+AT(9)+AT(10))
        IF(NJDR.EQ.7) THEN
          DISP=1
          NEXT=0
          IF(AT(9).NE.0) THEN
            NEXT=1
          ELSEIF(AT(10).NE.0) THEN
            NEXT=3
          ENDIF
          IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
        ENDIF
C
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 172  CONTINUE
C
      DO 174 I=1,NNQ(NQS)
```

```
          NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
          IF(NRANK.EQ.0) GO TO 174
          CALL RMOVE(NRANK,NQS,AT)
          CALL FFILE(31,AT)
 174    CONTINUE
        DO 176 I=1,NNQ(31)
          IF(NNQ(31).EQ.0) GO TO 176
          CALL RMOVE(1,31,AT)
          IF(I.EQ.1) AT(13)=FST(NQS-15,3)
          CALL FFILE(NQS,AT)
 176    CONTINUE
        RETURN
C
C***** WORKCENTER 8
C
 180    DO 182 I=1,NNQ(NQS)
          CALL COPY(I,NQS,AT)
C
          IF(NJDR.EQ.3) DISP=AT(18)
          IF(NJDR.EQ.5) DISP=(AT(18)-TNOW)/(AT(9)+AT(10))
          IF(NJDR.EQ.6) DISP=(AT(18)-TNOW)-(AT(9)+AT(10))
          IF(NJDR.EQ.7) THEN
            DISP=1
            NEXT=0
            IF(AT(10).NE.0) NEXT=2
            IF(NEXT.EQ.0.OR.NNQ(NQS+NEXT*3).EQ.0) DISP=0
          ENDIF
C
          NTRY=LOCAT(I,NQS)
          QSET(NTRY+16)=DISP
 182    CONTINUE
C
        DO 184 I=1,NNQ(NQS)
          NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
          IF(NRANK.EQ.0) GO TO 184
          CALL RMOVE(NRANK,NQS,AT)
          CALL FFILE(31,AT)
 184    CONTINUE
        DO 186 I=1,NNQ(31)
          IF(NNQ(31).EQ.0) GO TO 186
          CALL RMOVE(1,31,AT)
          IF(I.EQ.1) AT(14)=FST(NQS-18,4)
          CALL FFILE(NQS,AT)
 186    CONTINUE
        RETURN
C
C***** WORKCENTER 9
C
 190    DO 192 I=1,NNQ(NQS)
          CALL COPY(I,NQS,AT)
          IF(NJDR.EQ.3) DISP=AT(18)
          IF(NJDR.EQ.5.OR.NJDR.EQ.7)DISP=(AT(18)-TNOW)/AT(10)
          IF(NJDR.EQ.6) DISP=(AT(18)-TNOW)-AT(10)
```

```
        NTRY=LOCAT(I,NQS)
        QSET(NTRY+16)=DISP
 192  CONTINUE
C
      DO 194 I=1,NNQ(NQS)
        NRANK=NFIND(1,NQS,16,1,-1.0E+6,0.)
        IF(NRANK.EQ.0) GO TO 194
        CALL RMOVE(NRANK,NQS,AT)
        CALL FFILE(31,AT)
 194  CONTINUE
      DO 196 I=1,NNQ(31)
        IF(NNQ(31).EQ.0) GO TO 196
        CALL RMOVE(1,31,AT)
        IF(I.EQ.1) AT(15)=FST(NQS-24,5)
        CALL FFILE(NQS,AT)
 196  CONTINUE
      RETURN
      END
```

VITA 2

Bor-Yuh Leu

Candidate for the Degree of

Doctor of Philosophy

Thesis: AN INVESTIGATION OF GROUP SCHEDULING HEURISTICS IN A FLOW SHOP CELLULAR SYSTEM WITH WORKCENTER SHARING FOR THE FORBIDDEN EARLY SHIPMENT ENVIRONMENT

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Taichung, Taiwan, ROC, August 31, 1957, the son of Chi-Chung and Chang-Yen Leu. Married Tsao-Jean Leu on June 13, 1987. Father of twins, Chia-Luen (Alan) and Chia-Tien (Amy), born on August 22, 1988.

Education: Received Bachelor of Science degree in Engineering Science from National Cheng Kung University in 1981; received Master of Science degree in Mechanical Engineering from National Cheng Kung University in 1983; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1994.

Professional Experience: Teaching Assistant, Department of Mechanical Engineering, National Cheng Kung University, 1981-1983; Assistant Researcher, Aero Industry Development Center, Chung Shan Institute of Science and Technology, 1984-1990; Teaching Associate, School of Industrial Engineering and Management, Oklahoma State University, 1991-1994; Members of Alpha Pi Mu, Tau Beta Pi, the Institute of Industrial Engineers, the Society for Computer Simulation, and the Decision Sciences Institute.