1982

On scheduling and simulation within a GT cell

Muthuraj Vaithianathan
Iowa State University

Recommended Citation

Vaithianathan, Muthuraj, "On scheduling and simulation within a GT cell " (1982). *Retrospective Theses and Dissertations*. 7552.
https://lib.dr.iastate.edu/rtd/7552

INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.
2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.
3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.
4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.
5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University
Microfilms
International

300 N. Zeeb Road
Ann Arbor, MI 48106

Vaithianathan, Muthuraj

ON SCHEDULING AND SIMULATION WITHIN A GT CELL

Iowa State University PH.D. 1982

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

On scheduling and simulation within a GT cell

by

Muthuraj Vaithianathan

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Industrial Engineering

Major: Engineering Valuation

Approved:

In Charge of Major Work

For the Major Department

For the Graduate College

Iowa State University
Ames, Iowa

1982

TABLE OF CONTENTS

I. INTRODUCTION

A. Preamble

In recent times, much attention has been directed to the improvement of productivity. With almost religious fervor, federal, state and local governments, big and small industries, research and consulting institutions, and academia have industriously sought means of improving productivity. Productivity refers here to the reduction in manufacturing cost/unit either through increased output at the same expenditure level or increased output at a lower expenditure level. The concern for lagging productivity in the industry has prompted researchers both in the industry and the university to search for new improved ways of "doing business". Improved is used here in the narrowest definition of the "catch-all" phrase - either improving existing methodologies or looking at radically new approaches. In this narrow sense, much of the research effort has been directed to utilizing and integrating the capabilities of the computer with the main functional operations of the business. The meteoric rise in processing capabilities and plummeting costs for acquiring these capabilities have provided the incentive to promote wider use of the computer with increased productivity in mind.

In manufacturing, this effort has resulted in CAD/CAM (Computer aided design/computer aided manufacturing) systems, manufacturing resource planning systems (MRP), shop floor reporting systems, and production philosophies such as Group Technology (GT). Even skeptics of the present day developments in production systems cannot but recognize the gains in productivity that such systems have brought about. However, this gain is not without its shortcomings.

The developments behind the umbra of computer-aided manufacturing systems have brought to the limelight problems and concerns that researchers need to address in the near future. In the development of these systems, there has been a tendency to look at individual systems in isolation. This has resulted in difficulties in integrating these systems together to function harmoniously towards a common goal. Issues such as the structure of a common data base that can be used for both CAD and CAM have been brought to the forefront. This issue addresses the problems in integrating two "new" systems CAD and CAM. This integration is also a problem with old and new systems and philosophies. For example, how should MRP and GT function harmoniously towards a common goal? Besides, there are questions regarding the ways of performing traditional functions under new philosophies. For example, are existing ways of scheduling such as critical ratio techniques adequate for a GT cell? Or, could an improved method be developed that is more in harmony with the new philosophy?

These are rather important questions that need to be addressed immediately. There exists pressure to bolster manufacturing productivity through development, installation and use of manufacturing related systems such as CAD and CAM. This pressure tends to promote isolated individual approaches to manufacturing systems development. As a result of this, the interactions among such systems tend to be neglected. It was not long ago that forecasting, materials requirement planning, shop floor control and scheduling were treated as independent segments. Currently, much time and effort is directed to integrating these into a harmonious total system. Unfortunately, this past pitfall of developing isolated individual manufacturing related systems is currently not being avoided. Researchers should devote time to address the issues on a total systems concept such as in computer integrated manufacturing that the industry burdened by day to day production tasks does not find the resource to do.

This research work addresses to a small degree some of the concerns with Group Technology (GT), a production philosophy seeming to grow in popularity and widely in acceptance. A forecast of the future of the United States manufacturing environment, carried out both by the University of Michigan (14) and the International Institute for Production Engineering Research (30), predicted that approximately 50 to 75% of manufacturing industry will use group technology concepts in the period 1980-90. This forecast also predicts that the computer automated factory consisting of computer controlled cells will be a reality in many industries well before the end of the century. Specifically, the scheduling aspects of a GT cell are investigated in this work. Scheduling heuristics/procedures has been developed for the GT cell and their performance investigated via simulation of randomly generated job sets. The performance of these procedures is compared to the shortest processing time (SPT) a prevalently used priority assignment procedure. The comparison is made keeping in mind the environment and requirements of a GT cell.

B. Scheduling Background

The importance of efficient scheduling and sequencing in a manufacturing environment, even if only from a management perspective, has been long recognized. To quote F. W. Taylor (48):

> "In most case, for instance, of a machine shop doing miscellaneous work, in order to assign daily to each man a carefully measured task, a special planning department is required to layout all of the work at least one day ahead."

There are several broad classes of scheduling problems such as project scheduling, line scheduling and shop scheduling problems. In this research, discussion is restricted to shop scheduling. What is the shop scheduling problem? There are a host of problems variously called scheduling, sequencing and dispatching problems that fall under this domain (18). There are basically two levels of scheduling in a manufacturing environment--top level and detailed level scheduling.

At the top level, emphasis is on the scheduling of production and plant operations over an extended period of time. The objective here is to plan production quantities of the various products to accomplish a predetermined target such as a monthly forecast. This schedule is more a tentative plan of manufacture and is the vehicle to estimate, plan for, and procure labor, material, and machine oriented resources. This is sometimes referred to as the master schedule. The master schedule in conjunction with other production systems determines the release time of job orders.

Detailed scheduling concerns itself with the actual sequencing of job orders in machines that have already been released to the shop floor. These job orders have an establishd due date. The time period of the total schedule is much shorter than in top level scheduling.

The detailed level scheduling problem in a job shop can be defined as the determination of an "optimal" way of sequencing n jobs through m machines subject to operation procedure constraints. The solution of this problem has posed a formidable challenge to researchers. Barankin (6) underscores this very fact when he states:

> "Among the problems that econometrics and engineering has put to mathematics, the scheduling problem is one of the most interesting and challenging ...."

This challenge has lured several researchers into the field of scheduling. However, there has been limited success in the solution of the generalized n-jobs X m-machines problem.

The first formal documented "scheduling" procedure is the time progress chart named after its developer Gantt (49). The Gantt chart is not only an effective pedagogical tool but also a simplistic, and reasonably efficient tool for small sized n X m problems. However, it becomes unwieldy for medium and large sized problems. Since the Gantt chart, a considerable amount of research has been conducted in the development of optimal and optimal-tending heuristic algorithms for special cases of the generalized scheduling problems.

These theoretical and quasi theoretical approaches have been guided by formal mathematical elegance more than practical considerations of scheduling in a real manufacturing environment. As a result of this, an ever expanding rift has been forged between practitioners and theoreticians (19). All things considered, it would be totally unfair to single out the theoreticians for this rift; the inherent combinatorial explosion of the n X m ($n!^m$ schedules) problem poses computational difficulties primarily due to the size.

Almost all the research has been directed towards studying either the job shop environment or the flow shop environment. Tables 1.1 and 1.2 detail the difference between a job and flow shop with respect to routings. In a job shop, the jobs do not follow a set unidirectional order in machine visitation sequences. Further revisits are also allowed. Revisits refers to visiting a machine more than once as part of the production sequence. In a flow shop, the jobs need not have identical routes, but all the jobs follow unidirectional order in machine visitation sequences.

This research concerns itself with scheduling within a GT cell that could possess both job and flow shop flavors.

C. Group Technology

What is group technology and what is a GT cell? Group technology is a technique for identifying and bringing together related or similar components in a production process in order to take advantage of their similarities by making use of, for example, the inherent economies of similar setups and flow production methods. The applicability of GT in a manufacturing organization depends on the number and variety of different products manufactured and the manufacturing process required by them.

Table 1.1 Sample Routing for Flow Shop. Matrix Values Correspond to Machine Number

Operator # Job #	1	2	3	4	5	....	m
1	m	9	m-1	1	2		4
2	m	9	1	2			
3	m	9	1	2	4		
4	m	9	m-1	1	2		4
•							
•							
•							
•							
•							
•							
•							
n							

Table 1.2 Sample Routing of the Job Shop. Matrix Values Correspond to the Machine Numbers

Operator # Job #	1	2	3	4	5	....	m
1	m	9	m-1	1	2		4
2	4	9	m	1	m		9
3	m-1	8	7	8	1		3
•							
•							
•							
•							
•							
•							
•							
n							

One approach for implementing GT is as follows. First, the components are given a part code. This part code is generally of the block type in nature. Each block might consist of a few digits (between 2 and 4). Further, each block is used to define a characteristic of the component. This could include for example, the type of material, and the shape of the part. Within each block the digits provide further subclassification, such as the shape block which could not only identify the component as a cylindrical part but also define the outside diameter and inside diameter ranges. Igwilo (24) provides a comprehensive review of coding schemes and presents an example of applying a coding scheme.

With the help of the part code and the process routing of the components, the components are broken down into subgroups called families. Thus, the components within a subgroup have similarities in production processes.

Finally, for each family of parts based on the components in the family, the machines and their numbers are determined. Several methods have been proposed for the formation of machine cells (9, 12, 27, 31, 40). This set of machines are then relocated if need be, so they are close together. Such a set of machines, located closely to each other and dedicated to the manufacture of one family of parts is called a GT cell. This research concerns itself with the scheduling aspects of such a GT cell.

What is the major benefit of a GT cell? To quote Gallagher and Knight (16):

> "The first obvious saving from a group layout is the reduction of transportation and queueing time between operations, but the similarity of components within a family, allows resetting times to be minimized by the design of quick-change group tools and fixtures or by the sequencing of parts within the families".

Thus, within a GT cell the throughput time can be reduced by attempting to reduce the setup times. This reduction is achieved two ways. Firstly, the tools and fixtures are so designed mechanically that they are accommodative for the entire family. Secondly, by proper scheduling methods the jobs are sequenced to take advantage of similarities in setups that exist when a set of components belonging to one family are to be manufactured simultaneously.

This research evaluates the performance of a few scheduling heuristics and priority assignment procedures for a GT cell that implicitly attempt to take advantage of setup similarities.

D. Scheduling within a GT Cell

From a scheduling perspective, how similar or dissimilar is the GT cell environment in comparison to the traditional job or flow shop?

In job and flow shop scheduling research, the "goodness" of a schedule was evaluated with respect to flowtime or a related measure. Although other measures were not neglected, major emphasis was placed on flow time related measures. Some of the major advantages of GT are the simplification in the design, preparation of process sheets and production control by manufacturing within a cell. This is achieved via proper identification of part families and machine cells. In manufacturing, this advantage among others translates to similar setups. Other advantages include ease in preparing routing sheets and determination of standard times. This ease is primarily the result of having established a finite set of standard plans. The simplification and ease is achieved via proper identification of part families and machine cells. Thus, any scheduling rule or procedure devised for the GT cell should explicitly attempt to gain this advantage. In comparison then, the question "how well does this scheduling procedure perform for the GT cell" should be answered with more emphasis on reduction in set up times gained than would be the case for the question "how well does this scheduling procedure perform for the job or flow shop?"

In addition, the question of machine utilization should be viewed carefully. In the formulation of dedicated machine cells without intercell movements, there exists a strong tendency to require more machines than would in a composite job or flow shop. This tendency under given circumstances could result in under utilization not to mention higher fixed costs of manufacturing. On the opposite end of the spectrum, standardization of process plans and routes could result in increased or decreased load on machines. Hence, the scheduling procedure for a GT cell has to be evaluated giving consideration to machine utilization. It is not to say that machine utilization is of no consequence in a job or flow shop. The contention here is that it merely takes on additional importance.

In comparison to a general job or flow shop, the total number of machines and jobs is significantly smaller in a GT cell. Of what significance is this from a scheduling perspective? In a n X m job shop problem, the number of active schedules is $(n!^{m})$. Thus, the number of schedules to evaluate as n and m tend to be large, becomes computationally infeasible. Further, this complexity inhibits the development of involved scheduling algorithms to solve the n X m scheduling problem. Within the GT cell, both n and m, although still sufficiently large to prohibit exhaustive enumeration of all feasible schedules, are however within the domain of involved scheduling procedures. For example, in this research up to 65 x 11 problems were solved using procedures developed here in less than 10 CPU seconds on an NAS/6 computer.

One of the principal characteristics in the original part family classification that identified a part with a family is the part's general similarity in production process to those of other members in the family. Thus within a family of parts, there exists subsets that have distinct flow shop overtones. Also within a part family, the nature of the production process may be such that revisits and back tracking are a necessity, thus giving the GT cell the flavor of a job shop. It is in this sense that a GT cell is said to have a little of both, i.e., the job as well as the flow shop characteristics.

In a GT cell, the machines that comprise the cell are grouped together and generally are physically situated close to each other. Thus, it can be justifiably assumed that transportation costs between operations are negligible. In a general job shop, the assumption of negligible transportation cost may or may not be valid depending on the type of layout.

In a traditional job or flow shop, the emphasis is on throughput or related measure, percent jobs delayed and machine utilization. These same concerns are present in a GT cell. However, another criteria that is of critical importance in a GT cell is the setup savings realized by recognizing setup similarities. There are two important reasons why this is so.

Firstly, orders to the GT cell are most likely to come from a MRP system. Although MRP and GT can and do function harmoniously, the individual goals are different. The differences force a compromise to function together. The goal in MRP system is to manufacture only when needed. The goal in a GT cell is to realize potential savings in setup. Thus, parts that have similar setups need to be scheduled together in a GT cell to reduce setup changeovers. This could be the case even if when doing so, some parts are manufactured ahead of the time they are needed and so kept in stock. The resulting inventory is looked upon with disdain by the MRP system. On the other hand, the GT cell operates less effectively as a system, if potential savings in setup similarities are neglected in deference to a "zero" inventory goal. Thus, there exists a clash of goals. Ideally in a MRP-GT environment, the release time for make-to-stock orders have to be determined after weighing the savings in setup time by either preparing or postponing the release time against the inventory and possible delay costs of doing so. In the case of make-to-order orders the establishment of due date should again be determined weighing the savings in setup times by manufacturing at the "ideal" time against the requirements of the customer. In either case the savings in setup time becomes crucial for an economic evaluation. For example, consider two hypothetical schedules determined by two different procedures. The two different schedules generate the two sets of schedule related statistics shown in Table 1.3. For the sake of illustration let it be assumed that the statistics shown in the table establish all manufacturing related costs and savings. In this case:

Cost/job = Cell resident time/job * cost/unit cell resident time
+ tardiness time/job * % jobs late * cost/unit time tardy

Savings/job = Earliness time/job * (1-% jobs late) * rate/unit time
tardy + Savings in setup time/job * rate/unit time
saved in setup

Total Cost/job = Cost/job - Savings/job.

If the rates given in Case 1 are assumed, then schedule 2 is economically better than schedule 1 since the savings in setup time more than compensates for increased cell resident time and number of jobs late. If the rates given in Case 2 are assumed, then schedule 1 is economically better than schedule 2. In this case, the increased savings in setup time is not enough to overcome the cost of increased cell resident time and number of jobs delayed.

Secondly, the GT cell is the nucleus of the future flexible manufacturing system. The goal of the future, of course, is to operate "with the lights turned off" and no operators on the floor. In this future, the shop floor will be comprised of distinct quasi independent cells consisting of NC machines and machining centers, with robots and other material handling equipment such as conveyors transferring parts and tools, and

Table 1.3 Hypothetical Schedule Results

Criteria	Schedule 1	Schedule 2
Savings in setup time/job	15.00	45.000
Cell resident	50.00	65.000
Tardiness/job	10.00	15.000
Earliness/job	12.00	9.000
% jobs delayed	10.00	15.000
Case 1 Costs[a]	10.50	14.125
Case 1 Savings[a]	6.56	15.030
Case 1 Total Costs[a]	3.94	-0.905
Case 2 Costs[b]	15.50	20.000
Case 2 Savings[b]	4.91	8.280
Case 2 Total Costs[b]	10.59	11.720

[a] Case 1: $0.20/unit cell resident time per job
0.50/unit time tardy per job tardy
0.20/unit time early per job early
0.30/unit setup savings time per job

[b] Case 2: $0.30/unit cell resident time/per job
0.50/unit time tardy per job tardy
0.20/unit time early per job early
0.15/unit setup savings time per job

cameras coupled with sensors inspecting and monitoring the activities within the cell. The individual cells will be responsible for the manufacture of family of parts. With current technology, the limitation in achieving this goal lies in the inability of the robot to do complicated tasks that involve heavy parts and also demand precision. Jigs and fixtures fall in this category.

Thus in the cell of the future, an operating constraint and goal would be minimal changes in jigs and fixtures. This translates to requiring minimal setup changeovers for a given set of parts manufactured over a finite time. How can this be achieved? From a scheduling perspective, one way is to subgroup parts that are highly similar in setup from a large set of orders and schedule the similar parts together. This subgrouping also reduces the number of different parts that the robot has to recognize and identify. The scheduling procedure that schedules the subgroup consisting of parts highly similar on individual machines must again recognize setup similarities to minimize setup changes. This is figuratively shown in Figure 1.1. From Figure 1.1, it can be seen that scheduling needs to take place at two levels. The scheduling procedures needed at the two levels are broadly classified in this research as job-group scheduling and operation scheduling. At both the levels, there exists a demand on the scheduling procedure to recognize setup similarities and minimize setup changeovers. Consequently, the performance of a scheduling procedure devised for the cell needs to be evaluated on its ability to recognize setup similarities which can be translated to measuring the savings in setup time.

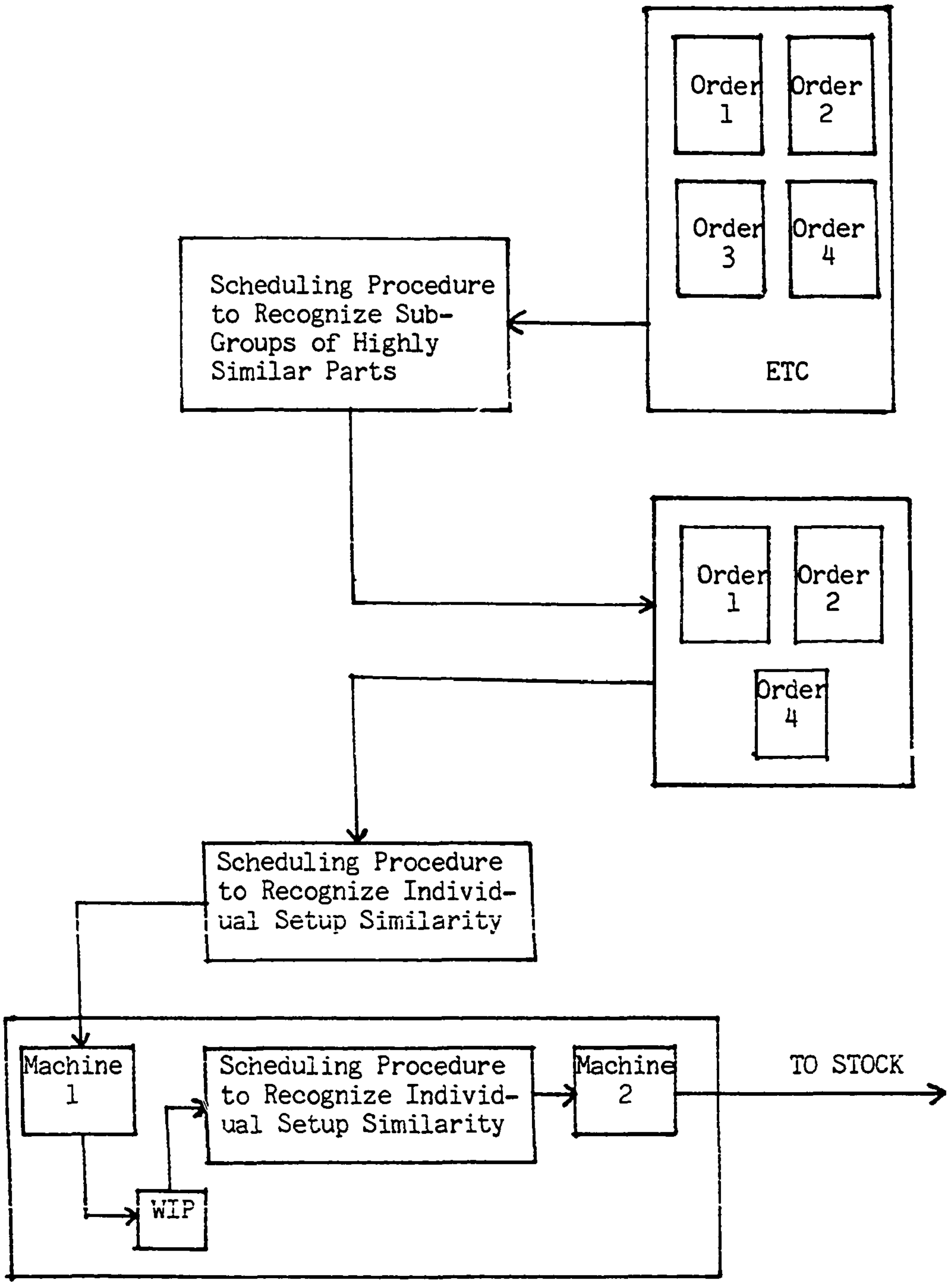


Figure 1.1 Scheduling in a "Future" Cell

The traditional emphasis on flow time related measures for a job or flow shop is supplanted by the savings in setup times for a GT cell. The need in the cell to recognize setup similarites forces compromises to be made on flow time related measures.

With MRP systems, the frequency of work loading and material issue has been at the best weekly, more often biweekly and sometimes even monthly. This was acceptable for the general job shop where considerable human intelligence and judgment scheduled the released material. The biweekly release of material resulted in higher WIP on the floor. In the future cell, with increased sophistication in computerized production systems coupled with minimal human intervention in the cell and limitations of computer controlled machinery such as robots, the frequency of work release to the floor will be more than is practiced now. This will automatically result in lower WIP and tend to de-emphasize importance of flow time related measures in a schedule.

E. Problem Statement

Consider an established GT cell that is loaded with two types of orders, i.e., make-to-stock orders (medium to large quantities) and make-to-order orders (individual to small quantities). Both the make-to-stock as well as make-to-order orders have multiple routes. There exist setup similarities between and among make-to-stock orders and make-to-order orders. The make-to-order orders need not necessarily be identical to make-to-stock orders in process routing.

Consider the following three questions:

1) What scheduling heuristic/procedure is to be adopted for the GT cell?
2) How should due dates be realistically estimated for orders to be processed in a GT cell?
3) How should release time be determined for a GT order through the MRP system?

Although at first glance, these questions may seem unrelated, a closer look will indicate that the answer to the first question will provide the basis for the examination of the other two.

Consider the second question. How should due dates be realistically established for an order? One approach is to base due dates on historical flow times. A second approach is to determine current shop load and then estimate the due date. In the above two approaches, the implicit assumption is that the processing of the order in question is independent of the process routings of jobs currently in WIP. This assumption is realistic in a job shop environment where generally the choice of the $(n+1)^{th}$ job to be processed is independent of the n^{th} job being processed.

In a GT cell, the jobs tend to have similar routings and subsets of the jobs tend to be related in setups. This was evidenced in the job sets used in this research. Also, the scheduling procedure should take advantage of this relationship between jobs in order to maximize the potential gains of a group manufacturing philosophy. Thus, the choice of the $(n+1)^{th}$ job to be processed is to some degree dependent on its setup characteristics and that of the n^{th} job. If this argument is accepted and extended, it can be stated that the due date of a new job to be manufactured in a GT cell is not only dependent on the cell load, but also on the process and setup characteristics currently at the cell and that in the immediate future. This current and immediate future setup character of the cell is implicitly a function of the scheduling procedure adopted in the cell. Thus, deciding a scheduling procedure is a prerequisite to solving the problem of establishing due dates for orders within a GT cell.

The release time for any order in a MRP system is generally limited to a function of the following:

1) due date
2) lead time
3) lot sizing techniques
4) load and capacity restrictions.

In the above, the release time established by due date and leadtime is tempered by either pulling ahead or pushing forward due to economies of lot sizing. For a make-to-order order that is to be manufactured within the GT cell, the economies of production take on a different perspective. Traditionally, the interest is to trade inventory costs/unit against cost of manufacture/unit, and the solution is to determine the quantity that will balance the two costs. For a customer order in the GT cell described, the quantity has already been establised. The problem now is to determine a release time for the order. By "pulling ahead" the release time, an inventory cost is incurred. This increased cost can be offset if the manufacturing cost of the order can be reduced. The quantity is not increased to reduce cost/unit, but by careful timing of the releases of the order the total manufacturing cost of the entire order can be reduced.

One way to reduce the manufacturing cost of the entire order is to reduce the total set up time required for the order. Reduction in set up times is achieved if orders with similar setups can be grouped together. Thus, if the release time for the make-to-order order is so timed that during its manufacture it can utilize existing setups then the manufacturing cost would be essentially reduced. This release timing is a function of the process and setup characteristics of the order, as well as the setup character of the cell at that time. This setup character of the cell as before is a function of the scheduling procedure used in the cell.

It is clear from the above discussion that for both establishing due date for orders as well as studying interactions between GT and MRP, the first-step is to establish a scheduling procedure for the GT cell. The procedure should explicitly recognize setup similarities.

For the future GT cell which will be a part of a true human "independent" CAM environment, scheduling procedures have to be devised that again recognize setup similarities. This is so, even at the expense of sacrifices in flow time related measures such as cell resident time, waiting time, etc.

Briefly stated, the problem in this research effort is to establish a scheduling procedure for the GT cell as in existence now and also that envisioned in the future. Further, this procedure should provide the basis to study MRP-GT interactions.

F. Research Objectives

1) To develop setup oriented heuristics/procedures for a GT cell that recognize setup similarities during operation scheduling.

2) To develop a procedure for sub-grouping parts that are highly similar with respect to common setups as a part of job-group scheduling.

3) To compare the performance of job-group scheduling against operation scheduling for detailed level scheduling.

4) To develop a simulation model and associated software that will permit the above analysis and provide a basis to study MRP-GT interactions.

5) To develop a plausible procedure for the future GT cell which will be part of a computer controlled manufacturing system.

G. Report Structure

The chapters in this report contain the following:

Chapter I introduces the reader to scheduling, group technology, scheduling in a GT cell, the problem statement and the scope of the research.

Chapter II provides a review of relevant literature in the area of setup oriented scheduling, decomposition in scheduling, and scheduling in a GT cell.

Chapter III details the assumptions in the model and defines the terminology. The notations used in the report are detailed in Appendix A. The reader is asked to review the notations carefully as they are used extensively in subsequent chapters.

Chapter IV details the operation and job-group scheduling procedures.

Chapter V details the structure of the software in the simulation model.

Chapter VI details the setup and results of the simulation analysis.

Chapter VII provides the conclusions of this study.

II. LITERATURE REVIEW

A. Traditional Approaches

Since the Gantt chart, there has been a considerable amount of research done in the area of scheduling. Much of this research has been directed to the solution of the n X m job and flow shop problem under restrictive conditions. For an excellent review and bibliography of this research, the interested reader is referred to Rowe (43), Conway (10), Eilon and King (13), Conway, Maxwell and Miller (11), Mellour (32), Gere (18), Mahendra and Arora (29), Moore and Wilson (33), Panwalker and Islander (39) and Spencer (46).

These approaches can be broadly classified into two categories.

1) Theoretical "optimizing" approaches

2) Heuristic "optimum tending" approaches

The optimizing approaches attempt to optimize with respect to a single criteria under restrictive conditions. They also tend to be computationally expensive for large scale problems. Optimization techniques in scheduling use linear programming techniques and more popularly branch and bound approaches. Also a third approach based on the permutation of job orders has resulted in optimum solutions. Examples of this case are the Johnson's algorithm for the 2 machine flow shop and Nayeri's (34) approach for the general flow shop. Bae (4) used extreme value theory to limit the searches in a branch and bound algorithm for the general job shop.

The advent of the computer provided considerable impetus to study the scheduling problem via simulation. Also, the difficulty of studying the n X m problem mathematically stimulated scheduling research using computer simulation as a tool. In the general generic of simulation, two major categories are included.

1) Random schedule generation or Monte Carlo technique.

2) Priority assigning/scheduling procedures.

In the Monte Carlo technique, a number of feasible schedules are randomly generated and the best is chosen. The reasoning here is that if enough feasible schedules are generated, one that is close to the optimum will be in that set, even if the optimum schedule is not in the set. Haque (21) coupled this approach with left-shifting to improve the random schedules generated for the general n X m job shop problem.

Priority assignment procedures and critical ratio techniques are evaluated through simulation of a typical shop. Considerable work in this approach was done by Conway (10). In this type of research, the evaluation of a scheduling procedure is based on mean flow time as a measure of performance. Conway in his study determined that the Shortest Processing Time (SPT) rule gave the best overall performance. Since this study, any new priority assignment procedure or critical ratio technique has been evaluated against the shortest processing time for comparison purposes.

Beyond the basic shop models, complex models were developed by Nelson (35) that were limited both by labor and machine. Holloway and Nelson (23) developed a model with multiple identical machines. Pai and McRoberts (38) developed a simulation model that permitted assembly operations. Bennet and Sawyer (7) used a simulation model to study batch/flow environments.

There have been few published papers in simulation modeling of an actual machine shop. Le Grande (28) was probably one of the pioneers of factory simulation using actuarial data. One of the earliest successful implementations of an application oriented scheduling-simulation model was that at the El Segundo Division of Hughes Aircraft Company (8). In retrospect, this could be called the forerunner for current on-line, shop floor reporting systems. Subsequently there have been other reported simulation studies of shop floor systems (36).

B. Setup Oriented Approaches

Research in the area of sequence dependent setup times has particular importance in a GT cell. Baker (5) presented the problem of sequencing setup dependent jobs for the n X 1 dynamic problem. He studied four setup oriented rules:

(i) Fixed Sequence:	Jobs are classified into classes based on sequence, and a fixed class sequence is established for sequencing among classes.
(ii) Minimum Setup time:	The class with the minimum setup time is chosen from the queue.
(iii) Fixed Sequence (SPT):	Within a fixed sequence of class, the job with the shortest processing time is chosen.

(iv) Minimum Setup time (SPT): The class is chosen based on minimum setup time and within the class by shortest processing time.

The above rules when compared to simple SPT were found to be inferior with respect to mean flow time. The above was true when setup time to processing time ratio was less then 0.25. For ratios greater than 0.25, the fixed sequence rule was found to be better than SPT.

There is a good news and a bad news side to Baker's research. The good news is that his research proved under certain conditions for set up dependent sequencing that there are heuristics better than SPT. The bad news is that this was proved only for a special case of the n X m problem, the n X 1 problem.

Gavett (17) studied the sequencing problem in a single production facility as a traveling salesman problem. The objective was to minimize setup times. The traveling salesman matrix then contained setup times required between changes in all pairs of jobs. Gavett tested three heuristics based on choosing a job with minimum transition time from the current one. The limitation of the study is that it is restricted to a single production facility. However, this study indicated that choosing jobs based on setup similarity to the last processed job gave near optimal solutions, i.e., optimal with respect to total setup time required for the entire batch.

Wilbrecht and Prescott (51) conducted a simulation study to decide the effect of setup oriented rules. This paper presents the results for only one rule, the SIMSET. In this rule, the intent is to choose a job from a queue that has a setup class similar to the previous one. However, the proposed method of achieving this was inaccurate. The method always chooses the job with the shortest setup time rather than the job that has a similar setup class to the previous job. Thus the research results can hardly be used as a basis for general interpretation. Further deficiencies in the reference paper were pointed out by Aggarwal (1).

White and Wilson (50) concentrated their research on estimating the sequence dependent setup times. On the foundation that in sequence dependent setup times the actual setup times are difficult and expensive to estimate, the authors developed a classification scheme for jobs based on mechanical processing characteristics. Based on the classification then, the setup time was estimated using regression models.

In summary, with respect to research in sequence dependent set up times, the following can be observed:

1) Limited effort has been directed so far in setup oriented scheduling procedures.
2) Almost all emphasis has been on single production facilities.
3) The identification of setup similarities has been restricted to single operation on a machine.

C. Decomposition Approaches

Parallel to the development of simple algorithms to solving large scale n X m problems, another approach was being pursued by researchers in scheduling. This approach emphasized decomposing the original n X m problem into smaller subgroups that could be handled mathematically. The effort was directed towards reducing the problem without concern for how the reduction is to take place.

Ashour (2) originally proposed a decomposition approach for the n X m job or flow shop problem based on exhaustive arrangement of all permutations. As an example, six jobs divided into two subgroups of three jobs each would result in $(6!/(3!)^2)$ arrangements. Those twenty arrangements are then evaluated as opposed to the original (6!) or equivalently 720 arrangements. Ashour (3) later modified this approach to avoid duplicate arrangements among subgroups. There was no rationale behind the choice of number of subgroups or number of jobs within each of the subgroups. However, the results were close to the real optimum with respect to the flow times.

Gupta and Maykut (20) proposed a more computationally efficient decomposition approach for the flow shop. This approach involved the decomposition of the original n job problem into two subgroups of n_1 and $(n - n_1)$ problems. The subgroup containing n_1 jobs was scheduled using the job pairing algorithm proposed by Page (37). The other $(n - n_1)$ jobs were then scheduled using exact techniques. The choice of n_1 was based on the computational feasibility of solving the $(n - n_1)$ subgroup using exact techniques.

Yamamato (52) presented an approximate solution to a static job shop scheduling problem. As opposed to a random permutation used by Gupta and Ashour, Yamamato used the following approaches:

1) Equalize machine load: Jobs are assigned to subgroups in a manner that will result in uniform assignment.
2) Similar pattern of machine sequence: Jobs that have similar patterns are grouped together.

3) Dissimilar pattern of machine sequence: Jobs that have dissimilar machine patterns are not grouped together.
4) Equalize machine load and dissimilar patterns. In the above, equal machine loads are set as additional criteria.

Apparently from Yamamato's work, subgrouping by similar pattern of machine sequence yields better results with respect to the throughput time than do others.

From a review of the literature it appears that:

1) Surprisingly, there has not been much interest in the decomposition approaches.
2) Decomposition techniques in existing research do not appear to be related to scheduling procedures.
3) No work has been done to decomposing an original problem into subgroups that contain jobs with a high degree of setup similarity among them.

D. Scheduling in a GT cell

Very little published work exists on scheduling approaches related to a group technology cell. Hitomi and Ham (22) considered the scheduling of jobs in a GT cell. The emphasis of this research was on obtaining optimum machine speeds to determine optimum group and job scheduling with respect to the total cell resident time. Adopting a branch and bound approach where the nodes in effect represent both the groups and the jobs within the groups, the procedure is to choose a group node from a set of group nodes. Once having chosen a group node, the job nodes belonging to that group node are chosen successively till all of them are sequenced. The process is then repeated till all other group nodes are chosen. The lower bound is the total flow time at each node and is estimated on the maximum sum of flow times of groups and jobs scheduled thus far and those still to be scheduled.

Khator and Moodie (26) expanded the piece part coding system and used it for scheduling. They extended the OPITZ coding system to incorporate lot size ranges, number of operations, accuracy, etc. Based on the extended part coding system, a machine capability matrix was developed to synchronize with this extended code. Thus, for every operation on all jobs, based on its extended OPITZ code and the machine capability matrix, it is assigned a desirability index value. The machine matrix that produces the largest desirability index value is the machine to which the job is sequenced. Within a machine queue, SPT is used to schedule jobs. This research emphasizes choosing a machine for a job to be scheduled when alternate machines are available. The problem of choosing a job from a set of jobs to be scheduled on a particular machine is not treated.

Spencer's (47) research attempts to minimize the total number of setups in top level scheduling. This attempt involves the incorporation of cost of lost production due to setup in the Wilson's classic formula for economic lot quantity determinations.

E. Summary

Based on the literature reviewed:

1) The theoretical approaches developed thus far have given minimal consideration to scheduling in a real manufacturing environment.

Consequently an ever expanding rift has been forged between theoreticians and practitioners (19).

2) The problem of obtaining an optimum solution for a large n X m problem is still not completely solved. However, approaches such as extreme value analysis indicate that biasing techniques are very effective in determining optimal tending solutions even if not optimal (4).
3) Little research in scheduling of n X m job or flow shop has concentrated on setup effects.
4) Decomposition approaches have had little or no basis in job/process characteristics.
5) Little research exists on scheduling in a GT cell.
6) Little quantitative research has been conducted to study GT-MRP interactions.

III. DEFINITIONS AND ASSUMPTIONS

A. Definitions

In essence, the manufacturing environment can be envisioned as a network of queues, the facilities being the machines and the customers being the jobs. The problem of scheduling is then to determine a way to service the next job from the job queue in front of each machine. In choosing a particular job over the others in line, the jobs in queue have essentially been prioritized. The scheme to prioritize is denoted as the priority assignment procedure. This priority assignment is generally done with a view to optimize some criteria that has an impact on the overall performance of the shop floor. The scheduling procedure is then merely a policy that dictates the particular order of choosing a particular job among the set of prioritized jobs in queue. Thus, the priority assignment procedure is merely a subset of the overall scheduling procedure. The measurement criteria refer to the method of evaluating the performance of the scheduling procedure.

Consider the example where in a queue, the jobs with earliest due dates are given a high scalar value which is the priority index. This then would be a priority assignment procedure. Now further, the scheduling procedure might dictate that jobs with the highest priority index be scheduled first. In case of a conflict, the job entering the queue first will be scheduled ahead of a job with a similar priority index. The effectiveness of this scheduling procedure could be measured by determining the number of late jobs.

The definitions are not standarized in the area of scheduling. Hence, the following is presented to define terms as used in the research.

Cell	A set of workcenters.
Completion Time	The time when the last operation on a job is completed.
Due Date	The calendar date on which the job is due out of the shop.
Earliness	The time by which a job is completed ahead of its due date.
Flow Time	The time spent by a job on the shop until the process of the last operation.
Heuristics	General method used to solve problems which defy solution by standard techniques. Examples are: "sequencing" and "line balancing" problems. Literally, heuristics means "serving to find out and encouraging further investigation." Techniques that lead to solutions by trying "common sense" rules and procedures rather than rigorous optimality criteria.
Job	An individual shop order number for a part to be made. May define one or several quantities of the part.
Job Order Number	A number assigned to each job.
Lateness	The difference between the completion time and due date. If this quantity is negative it implies that the job was completed ahead of schedule.
Lot Quantity	The number of identical units to be produced of an individual part number under a job order.
Machine number	A unique number given to a machine within a workcenter. All machines in the cell have to belong to a workcenter.
Operation Time	The amount of actual time spent on the machine for each operation.
Part	An individual component that has its standardized routings and associated times.

Part Number	A number given to a part for identification purposes.
Priority Index	A scalar value assigned to every job to reflect the importance of the job in relation to other jobs.
Process Routing	The precedence constrained sequence of machine numbers that the part has to follow from raw material to finished stage.
Process Time	The sum of the setup and operation times.
Remaining Time	It is the time remaining between the due date and the current current time after allowing time for remaining operations.
Setup Time	The amount of time spent in machine or labor preparation before individual operations can commence on respective machines.
Start Date	The date before which the job cannot be routed due to several reasons. One of which may be a lack of raw material.
Tardiness	The time taken, if any, beyond the due date for a job to be completed.
Workcenter	A unique center with a given identification number. The workcenter may consist of one or more machines.

B. Assumptions

The following assumptions are made in the development of the scheduling procedure and the simulation model.

1. Assumptions regarding the cell

1) An established GT cell with the machine and boundaries defined is existent.
2) Work-in-Process storage is available in the cell.
3) The set of machines included in the cell are located in close proximity, thereby validating the assumption of negligible transportation time and cost.

2. Assumptions regarding jobs

1) The job set is a combination of make-to-stock and make-to-order orders.
2) No intercell movement of jobs.
3) Only intracell movements of jobs are permitted.
4) The due dates for all job orders are known.
5) An earliest start date is provided for each job order.
6) Priorities of jobs at time of release are assumed to be equal unless otherwise specified. Note that this priority does not refer to priority index values that may be calculated as part of the scheduling procedure.

3. Assumptions regarding machines

1) Each machine can work on only one operation of one job at a time.
2) A rated operating capacity for all machines within the cell is available.
3) A maximum capacity with lead time required to obtain this capacity is available.

4. Assumptions regarding processes

1) At time of release to the GT cell, routings are available.
2) At time of release to the GT cell, the standard setup times and operation times are established and known.
3) The established times are assumed to be deterministic.
4) Multiple routes for an order are permitted with a maximum of three. However, if one route is chosen, all units in the lot follow that route.

5. Assumptions regarding operating environment

1) An established part family classification and associated code is existent.
2) The cooperation or synchronization of two or more machines is never required to perform an operation.
3) There is a single limiting source called "machine" for which the job must compete and wait.
4) The scheduling procedure assumes that lots will not be split.
5) An operation once started will not be interrupted due to shift changes.

As detailed in Chapter I, the environment and requirements of a scheduling procedure, and the nature of this research requires the evaluation of a scheduling procedure to place more emphasis on savings in setup times as opposed to waiting time, cell resident time etc. For purposes of evaluation of a scheduling procedure for a GT cell, the following criteria in descending order of importance is assumed:

1) Savings in setup time.
2) Percent jobs delayed.
3) Throughput time.
4) Machine utilization.
5) Flow time related measures.

IV. THE APPROACHES

A. Operation and Job-Group Scheduling

A review of the following in Appendix A would help the reader to better understand this chapter.

COMPOP(I)
DUE (I)
LOT (I)
MCHNUM(I,J)
MCHOP(I,J,K)
MCHSTS(I,J,K)
NJOBS
NMCH
NOPN(I)
NUMMCH
OPNALL(I,J)
PRSTIM(I,J,K)
QUE(I,J,K)
QUELNG(I,J)
STPCLS(I,J,K)
STPTIM(I,J,K)

How should scheduling take place within a GT cell? Any scheduling procedure for a GT cell must attempt to take advantage of similarities in setup that might exist among a set of job orders. The identification of this setup similarity can be defined in two ways.

1. Definition 1

The set of job orders in which similar setups has to be recognized for a particular machine Q in workcenter P (MCHNUM (P, Q)) can be defined as the following:

$$\begin{aligned} S\{\text{candidates}\} = \{i \mid i \cap j \neq \emptyset \text{ for all } j,\ 1 \leq j \leq \text{NJOBS}\} \text{ and} \\ \{i \mid i \cap k \neq \emptyset \text{ for all } \{k \subset \{\text{QUE }(P,Q,L)\}, \text{ subject to} \\ 1 \leq L \leq \text{QUELNG }(P,Q)\}\} \end{aligned} \qquad 4.1$$

where QUE(P,Q,L) refers to the job identification of the 1^{th} job in queue in front of the Q^{th} machine in the P^{th} workcenter,

and QUELNG(P,Q) refers to the length of the queue in front of the Q^{th} machine in the P^{th} workcenter.

This restricted set is comprised of jobs waiting in line to be processed in front of a given machine Q in workcenter P. Implicit in the above is that all previous operations are completed for this set of job orders. In terms of the simulation model developed, this implies that in addition to above, the following should also be true.

$$\mathrm{OPNALL}\ (x, 1) = i \tag{4.2}$$

$$\mathrm{OPNALL}\ (x, 4) = P \tag{4.3}$$

$$\mathrm{OPNALL}\ (x, 5) = Q \tag{4.4}$$

$$\mathrm{OPNALL}\ (x, 7) = 0 \tag{4.5}$$

$$\text{and } \mathrm{OPNALL}\ (x, 2) = \mathrm{COMPOP}\ (i) + 1 \tag{4.6}$$

where $\mathrm{MCHOP}\ (P, Q, 1) \leq x \leq \mathrm{MCHOP}\ (P, Q, 2)$

From this restricted set of likely candidates, the jobs with similar setups are those that have the following condition satisfied.

$$\mathrm{OPNALL}\ (x, 6) = \mathrm{MCHSTS}\ (P, Q, 1) \tag{4.7}$$

From the set of jobs that satisfy Equations 4.1 through 4.7, a choice can be made. Equation 4.1 restricts consideration to only those jobs in queue in front of a machine in a workcenter and Equation 4.6 further restricts to consideration of only a single operation of each of the jobs.

The set of heuristic procedures that is limited to the set defined by Equation 4.1 is defined as operation scheduling.

2. Definition 2

A second way to define the set of job orders in which similar setups have to be recognized is to consider the entire job set. Thus, no restriction is placed here on a machine or a queue.

$$S\{\text{candidates}\} = \{\ i \mid 1 \leq i \leq \mathrm{NJOBS}\} \tag{4.8}$$

The setup similarity between pairs of jobs i and j is defined as

$$SIM = F(STPCLS(i, k, 1), STPCLS(j, L, 1)) \qquad 4.9$$

subject to $1 \leq k \leq NOPN(i)$ and

$1 \leq L \leq NOPN(j)$

and $1 \leq i$ and $j \leq NJOBS$

Based on the SIM(I,J)s subgroups of jobs that are highly similar in setups can be clustered together. This approach to identifying similarity and scheduling is defined as job-group scheduling.

Table 4.1 details the difference between operation scheduling and job-group scheduling.

Table 4.1 Operation Scheduling Vs Job-group Scheduling

Operation Scheduling	Job-Group Scheduling
Setup similarity identified at time of scheduling	Setup similarity identified prior to scheduling
Setup similarity is identified considering individual operations	Setup similarity is identified considering all operations

The next section is devoted to detailing the various heuristics in operation scheduling that are investigated. A detailed explanation of job-group scheduling and subgroup determination follows the section on operation scheduling.

B. Operation Scheduling

In operation scheduling, the set of candidates to be considered for scheduling for a particular machine MCHNUM (P, Q) is defined using definition 1 as follows:

$$S\{\text{Job candidates}\} = [i \mid i \cup j \neq \emptyset \text{ for all } j,\ i \leq j \leq \text{NJOBS}] \text{ and } \{i \mid i \cup k \neq \emptyset \text{ for all } \{k \subset \{\text{QUE }(P, Q, L), \text{ where } 1 \leq L \leq \text{QUELNG }(P, Q)\}\} \quad 4.10$$

In case

$$\{i \mid i \cup k = \emptyset, \text{ for all } \{k \subset \{\text{QUE }(P, Q, L),\ 1 \leq L \leq \text{QUELNG }(P, Q)\}\}$$

which can happen when QUELNG (P, Q) = 0 then MCHNUM (P, Q) is no more in consideration. In terms of the simulation, any element X belongs to the set when the following is true:

OPNALL (x, 1) = i,

OPNALL (x, 4) = P,

OPNALL (x, 5) = Q,

OPNALL (x, 7) = 0,

where MCHOP (P, Q, 1) $\leq$ X $\leq$ MCHOP (P, Q, 2). 4.11

The precedence constraints of scheduling are satisfied by ensuring that

OPNALL (x, 2) = COMPOP (i) + 1 4.12

There are 3 cases of S{job candidates}.

1. Case 1

In the first case, the set could be a null set, when S{Job candidates} = $\emptyset$. No problem exists as the decision process is now transformed to choosing another machine that is to be scheduled. This is achieved in the model by building a set of workcenter indices and machine indices such that:

$$S\{\text{machine candidates}\} = \{i, j \mid \text{MCHSTS }(i, j, 2) > 0\} \text{ and } 1 \leq i \leq \text{NMCH},\ 1 \leq j \leq \text{NUMMCH}(i) \quad 4.13$$

From this set of machine candidates, the particular machine Q in particular workcenter P is chosen by

$$P, Q = \min \{MCHSTS (i, j, 2) \mid i, j \subset S\{\text{machine candidates}\}\} \qquad 4.14$$

2. Case 2

The second case is when the following is true:

$S\{\text{Job candidates}\} \neq \emptyset$ and

$N(S\{\text{Job candidates}\}) = 1$,

where $N(S\{\text{Job candidates}\})$ is the number of elements in the set.

This case again presents no problem, since only one job candidate exists to be chosen.

3. Case 3

The third case is when the following is true:

$S\{\text{Job candidates}\} \neq \emptyset$ and

$N(S\{\text{Job candidates}\}) > 1$.

In this case, multiple candidates are available, and a choice has to be made. Such a choice could be a function of any one of the following:

1) Setup time.
2) Total processing time.
3) Remaining slack.

Heuristics/procedures can be derived as a function of the above singly or as a combination.

In defining the set of likely job candidates as in Equation 4.10, no consideration was given to identifying setup similarities. From the set defined in Equation 4.10 the following subset is defined.

$$\begin{aligned} S\{\text{similar setup jobs}\} = \{i \; &\mid i \subset S\{\text{job candidates}\} \text{ and} \\ &\mid \text{OPNALL } (x,\ 6) = \text{MCHSTS } (P,Q,1) \\ &\mid \text{OPNALL } (x,\ 1) = i\} \end{aligned}$$

$$\text{where MCHOP } (P,Q,1) \leq x \leq \text{MCHOP } (P,Q,2) \qquad 4.15$$

It is evident from Equation 4.15, that in operation scheduling, the setup similarity is recognized with respect to the last or current setup on the machine. There are again 3 cases based on the number of elements in the set defined by Equation 4.15.

a. Case 3A   The first case is when N(S{similar setup jobs}) = 1. This presents no problem as the single element is chosen to be the next job to be processed.

b. Case 3B   The second case is when N(S{Similar setup jobs}) > 1. In this case a choice has to be made among the various jobs. Three criteria were defined for this purpose in this research.

1) Criterion 1   This criterion is a ratio of the setup time to the processing time and is defined generally as follows:

$$\text{Setup/Process Ratio} = \frac{\text{STPTIM } (I,\ J,\ K)}{\text{PRSTIM } (I,\ J,\ K) * \text{LOT } (I)} \qquad 4.16$$

From the several ratios defined as in 4.16 the priority is given to the m^{th} job that satisfies the following relationship.

$$\frac{\text{STPTIM } (M,J,K)}{\text{PRSTIM } (M,J,K)*\text{LOT}(M)} = \text{MAX} \left[\frac{\text{STPTIM } (I,J,K)}{\text{PRSTIM } (I,J,K)*\text{LOT}(I)} \right] ;\ M \subset I \qquad 4.17$$

where $M \subset I$ and for all I's the K^{th} alternate of the J^{th} operation is processed in the Q^{th} machine of the P^{th} workcenter, which is the machine under consideration.

2) Criterion 2 This priority assignment procedure is a function of the remaining slack and the number of operations yet to be completed for the job. The remaining slack could be defined as follows:

$$\text{Remaining slack} = \text{DUE(I)} - \left[\sum_{\text{J=COMPOP(I)+1}}^{\text{NOPN(I)}} \{\text{STPTIM (I,J, 1)} + \text{PRSTIM (I, J, L)} * \text{LOT(I)}\} \right] \qquad 4.18$$

The above value merely determines the amount of work yet to be done on a job I. It makes no allowance for work in queue in front of machines that the I^{th} job has to visit for its yet to be completed operations. The following equation redefines Equation 4.18 and accommodates for the work in queue.

$$\text{Remaining slack}_i = \text{DUE (I)} - \left[\sum_{\text{J=COMPOP(I)+1}}^{\text{NOPN(I)}} [(\text{STPTIM (I,J,1)} + \text{PRSTIM (I,J,1)*LOT(I)} + \text{MCHSTS (OPNALL (}x_j\text{,4), OPNALL (}x_j\text{,5), 8))]} \right] \qquad 4.19$$

subject to OPNALL $(x_j, 1) = I$

OPNALL $(x_j, 2) = J$

and OPNALL $(x_j, 3) = 1$

From the remaining slack as defined in Equation 4.19, the following ratio is obtained:

$$\text{Slack/Operation ratio}_i = \frac{\text{Remaining slack}_i}{\text{NOPN(I) - COMPOP(I) + 1}} \qquad 4.20$$

Again based on several ratios defined as in Equation 4.20, priority is given to the m^{th} job that satisfies the following relationship

$$M = \min_{\text{all } i} \{\text{slack/operation ratio}_i\}; \; M \subset I \qquad 4.21$$

where for all I^s the K^{th} alternate of the imminent operation is processed in the Q^{th} machine of the P^{th} workcenter. Whereas the first criterion attempted to maximize the setup savings, this attempts to minimize overall tardiness.

3) Criterion 3 Although operations in a GT cell are biased towards maximizing potential advantages in similar setups, there are obviously other concerns. For example, much as desire exists to minimize setup time, this cannot be achieved without considering the due dates of the various jobs and overall tardiness. The above detailed heuristic criteria emphasize either minimization of setup times or lateness. To effect a balance the following ratio is investigated.

$$\text{Combination ratio}_i = \frac{A}{(\text{setup/process ratio}_i} + \frac{1}{B}(\text{slack/operation ratio}_i) \qquad 4.22$$

where A & B are weights assigned to the individual ratios. In this research, a few combinations of A and B were investigated for a small number of data sets from which, A = 10.0 and B = 2.0 was determined to be superior with respect to criteria established in the previous chapter.

From the several combination ratio_i's possible for the various jobs the m^{th} job is chosen such that the following is true.

$$\text{Combination ratio}_m = \underset{\text{all } i}{\text{MAX}} \ (\text{combination ratio}_i) \ ; \ M \subset I \qquad 4.23$$

Here, again, the i's satisfy the earlier condition given for Equation 4.17.

The above three heuristic criteria are for the case when N(S {similiar setup jobs}) > 1 where S{similar setup jobs} is as defined in Equation 4.15. To recap, both the cases, when N(S{similar setup jobs}) = 1 and > 1 have been discussed. The last possible case yet to be discussed is when N(S{similar setup jobs}) = 0.

c. Case 3C Quite simply, N(S{similar setups}) = 0 implies that for the machine in consideration, none of the jobs waiting in queue to be processed have a setup similar to the one already in the machine. The corollary then is to choose a job with a new setup for the machine.

A rational decision needs to be made in determining which of the several possible setup classes to choose. One such decision would be to choose the setup class that results in minimum setup change over time, thereby minimizing lost production time due to setup changeovers. This of course requires a matrix of setup change times, from which the choice can be made.

Rather than basing the decision on the characteristic of the setup class itself, another approach is to base it on the characteristics of the jobs that are in queue in front of the machine and which belong to the various setup classes. This is intuitively more appealing since the time reduced in setup due to choosing a class that consists of a large number of jobs with similar setups will be greater than the time spent in the actual setup changeover from one class to the other. For example, given that the changeover time from class A to class B is three time units and that from A to C is 5 time units, there exists a temptation to choose B following A. However, this decision is unwise if the number of jobs in classes B and C are such that a total of reduction of two time units and 15 time units respectively is possible. In this case, C is a preferred choice over B. Further, this approach is more in line with the philosophy of operation scheduling which is job based on job and operation characteristics.

When N(S{similar setup jobs}) = 0 the parent set S{job candidates} as defined in Equation 4.10 becomes the base set for consideration. Consider the following definitions:

NCLS - the number of different setup classes among jobs in queue in front of Q^{th} machine in the P^{th} workcenter.

NUMCLS(I) - the number of jobs in the i^{th} setup class. $1 \leq I \leq NCLS$

CLSVAL(I) - the setup class value of the i^{th} setup class. $1 \leq I \leq NCLS$

By the above definition of NUMCLS the following is true.

$$\sum_{I=1}^{NCLS} NUMCLS(I) = N\ (S\{job\ candidates\})$$

If NCLS = 1, obviously there is no choice to be made. However, there is a choice to be made when NCLS > 1. Consistent with the 3 heuristic criteria detailed for the case when N(S{similar setup jobs}) > 1, the following two are defined for the choice of a setup class to changeover to when N(S{similar setup jobs}) = 0 and NCLS > 1.

<u>1)</u> <u>Criterion 4</u> Here the class chosen is the one that has the largest sum total of setup times for all jobs within that class. The

assumption is that by this choice the greatest reduction in setup time can be achieved. Thus the setup class to change to is given by

$$\text{CLSVAL} \langle i \rangle = \underset{\text{all } i}{\text{Max}} \sum_{k=1}^{\text{NUMCLS(I)}} \left[[\text{STPTIM (OPNALL } (x, 1), \text{ OPNALL } (x, 2), \text{ OPNALL } (x, 3)] \right] \qquad 4.24$$

Subject to

a) $\text{OPNALL } (x, 6) = \text{CLSVAL } (K)$ respectively

b) $1 \leq I \leq \text{NCLS}$

c) $\text{MCHOP } (P, Q, 1) \leq X \geq \text{MCHOP } (P, Q, 2)$

2) Criterion 5 Here the class chosen is the one that as a subgroup has smallest remaining slack or in shop language, the "hottest". Here the setup class to change to is given by

$$\text{CLSVAL} \langle I \rangle = \underset{\text{all } i}{\text{Min}} \sum_{K=1}^{\text{NUMCLS(I)}} \left[(\text{DUE(OPNALL } (x_1, 1)) - \text{ITIM}) \right] \qquad 4.25$$

subject to

a) $\text{OPNALL } (X, 6) = \text{CLSVAL(K)}$

b) $\text{MCHOP } (P, Q, 1) \leq x \leq \text{MCHOP } (P, Q, 2)$

and where ITIM is the "current" clocktime when the decision is made.

In both cases, when NCLS = 1 and NCLS > 1, as long as NUMCLS (J) > 1, a particular job from a chosen class has yet to be chosen. Under these conditions, a first-come-first-serve rule was adopted. First-come is defined as the earliest entry time into the queue in front of the machine.

Based on the 5 heuristic criteria above the following six procedures were tested. For lack of better names, the procedures are named by combining the numbers assigned to the criterion used by the procedure. The six procedures are:

ONEFOR - A combination of heuristics one and four.

ONEFIV - A combination of heuristics one and five.

TWOFOR - A combination of heuristics two and four.

TWOFIV - A combination of heuristics two and five.

THRFOR - A combination of heuristics three and four.

THRFIV - A combination of heuristics three and five.

The implication of the word combination as used here needs clarification. In all cases, the first of the two part criteria is used when $N(S\{\text{similar setup jobs}\}) \geq 1$ and the second part is used when $N(S\{\text{similar setup jobs}\}) = 0$ and $NCLS > 1$.

C. Job-Group Scheduling

In operation scheduling, the setup similarity is recognized for the particular operation to be processed in front of a machine when the machine is due for scheduling. In job-group scheduling, the similarity is identified even before scheduling takes place on the first operation of the first job. Further, this similarity between pairs of jobs is not restricted to one operation. Rather, the entire set consisting of all operations for both the jobs is spanned. From this, the degree of similarity between the pairs of jobs is established. By this measure, two jobs highly dissimilar for one operation but highly similar for a majority of the others would be deemed as a highly similar pair of jobs. The method of computing similarity ensures that this value for any pair of job lies between 0 and 1. In this manner, similarity between all possible combinations of pairs of jobs can be established. From this set of similarity values, subgroups that consist of jobs highly similar to each other can be formed. Based on different characteristics of the elements in the subgroup as a whole, the sequencing of subgroups is established. After the subgroup sequence is established, sequencing of individual jobs within a subgroup can be determined.

Based on the above discussion, job-group scheduling is essentially a four step process.

1) Establish similarity and develop the similarity matrix.
2) Based on the similarity matrix identify subgroups.
3) Determine subgroup sequencing.
4) Determine job sequence within subgroups.

1. Establishing similarity

This is the first step in the four step process of job-group scheduling. The basic similarity measure used was first proposed by Jaccard in 1908 (45). It is defined as follows:

The similarity between two sets R_i and R_j denoted by

$$\mathrm{SIM}(i,j) = \frac{R_i \cap R_j}{R_i \cup R_j} \qquad 4.26$$

The value of SIM(i,j) will range from 0 to 1. This simple measure has been used extensively in numerical taxonomy, information theory and group technology (9, 44). Measure 4.26 can also be used to quantitatively determine the similarity of two jobs with respect to setups.

Setup class values are first established for each machine within the cell based on the different setups in the machine. Here the machines are dedicated to performing a single operation. Examples of such machines could be vertical drilling machines and punch presses. Thus, if there are 4 different setups on a particular dedicated machine within the GT cell, then there are 4 setup classes.

If the machines within the cell are utilized for multiple operations such as a vertical turret milling machine for both milling and drilling, then setup class values have to be established to differentiate between individual operation. Thus, a face milling operation on a vertical turret milling machine might have 3 different setup classes, whereas a trial bore operation on the same milling machine might have 5 different setup classes for all parts within the part family.

How can the various setup class values be established? One method is to look at individual parts within the family and, based on actual knowledge of the production process, manually establish the class values.

An alternative approach would be to use the part code itself. Prior to the determination of part families, each part would have been assigned a GT part number or GT code based on the shape and dimensions, nature of production process, types of material, etc. The GT code is a block type code with each block representing a particular characteristic of the part and the values representing the ranges in the particular characteristic. For example, a block of 3 digits might represent the shape of the part with the first digit to represent a particular range of outside diameter and the second to represent the range on the inside diameter and the third digit to represent the overall length.

Using the block GT code, a matrix can be setup that defines the class value for each machine and operation based on the range of acceptable values in the various digits. An example of a building block for such a matrix is given in Table 4.2.

Table 4.2 Setup Class Identification Table

Operation Number	Machine Number	Setup Class	Digit Positions in the code 1	5	20	24
10	555	1	2	10-15	5	1-7
15	555	1	-	14	4-7	1-7

The first row implies that the set up class value is 1 for operation 10 of any part X on machine 555 if in the GT code for that part X the following is true.

1) In digit position 1 the value must be equal to 2.
2) In digit position 5 the value must lie between 10 and 15.
3) In digit position 20, the value must be equal to 5.
4) In digit position 24, the value must be between 1 and 7, both inclusive.

Thus, given a set of NJOBS with their GT part number, the setup class values for the operation can be established vis-a-vis Table 4.2. These class values are then used to compute the similarity between any pair of jobs. The quantitative estimate of the degree of similarity between jobs is based on a variation of Equation 4.26.

Consider two jobs i and j which belong to a set of NJOBS.

Assuming $NOP(I) \leq NOP(J)$,

Then SIM(I, J), the similarity coefficient between the I^{th} and J^{th} job, is given by:

$$\mathrm{SIM(I, J)} = \frac{\sum_{m=1}^{\mathrm{NOP(I)}} E_m}{\mathrm{NOP(I)} + \mathrm{NOP\ (J)} - \sum_{m=1}^{\mathrm{NOP(I)}} E_m} \qquad 4.27$$

where (1) $E_m = 0$ if STPCLS (I, M, 1) $\neq$ STPCLS (J, N, 1) for any N,

$1 \leq N \leq$ NOP (J)

(2) $E_m = 1$ if STPCLS (I, M, 1) = STPCLS (J, N, 1) for any one N,

$1 \leq N \leq$ NOP (J)

and NWKCNT (I, M, 1) = NWKCNT (J, < N >, 1) 4.28

where < N > is that n^{th} operation in Job J for which STPCLS (I, M, 1) = STPCLS (J, <N>, 1)

Also, depending upon the requirement, the following condition can also be included in addition to Equation 4.28.

OPNUM (I, M, 1) = OPNUM (J, <N>, 1) 4.29

Whereas, Equation 4.28 restricts for similarity within a particular machine, Equation 4.29 restricts to identical operations on identical machines.

Proceeding as dictated by Equation 4.27, SIM(I,J) can be calculated for all the jobs and a similarity matrix determined. The similarity matrix will be a square matrix of size NJOBS. Further, it is symmetrical about the principal diagonal because SIM(I,J) = SIM(J,I).

2. Determining subgroups

Once having established the similarity matrix, the next step is to form subgroups that contain jobs having strong similarity between each other. This can be achieved by any one of several statistical clustering techniques such as single linkage analysis, average linkage analysis, etc.

The approach adopted in this research effort is a variation of one described by Rao (41) as detailed in Kennedy (25). Ross (42) presents further variations of the algorithm. The algorithm is based on average similarity and threshold values. Before the algorithm is presented, let the following be defined.

NGRP	Number of subgroups
MXGRP	Maximum number of subgroups specified
THRESH	Threshold value specified
GRPCNT(I)	Number of jobs in the i^{th} subgroup $1 \leq I \leq NGRP$
GROUP(I,J)	The job identification of the j^{th} entry in the i^{th} group $1 \leq I \leq NGRP$ $1 \leq J \leq GRPCNT(I)$

The algorithm for determining subgroups is as follows:

1) Specify both the maximum number of subgroups and the threshold value.
2) Set I the group index to be 1.
3) Check to see if I = MXGRP - 1. If so go to Step 17 otherwise go to Step 4.

4) Choose among unallocated jobs the maximum similarity. Thus, if M = Row number, and N = column number in the matrix when the following is true:

$$\mathrm{SIM}(M,N) = \max_{\text{all } P,Q} (\mathrm{SIM}(P,Q))$$

Subject to P and/or $Q \subset$ GROUP (K, L) for all K and L where

$$1 \leq K \leq I - 1$$
$$1 \leq L \leq \mathrm{GRPCNT}(K)$$

5) Check to see if $\mathrm{SIM}(M, N) \geq \mathrm{THRESH}$. If so, go to Step 6, otherwise go to Step 17.

6) Set TOTAL = SIM(M, N) and GRPCNT(I) = 2.

7) Compute the following totals for all jobs S where S is subject to the following condition.

$S \subset$ GROUP (K, L)

for all K and L where

$1 \leq K \leq I$

$1 \leq L \leq \mathrm{GRPCNT}(K)$

$\mathrm{TOTALS}(S) = \mathrm{TOTAL} + \mathrm{SIM}(M, S) + \mathrm{SIM}(N, S)$

8) Choose R such that the following is true.

$$\mathrm{TOTALS}\ (R) = \max_{\substack{\text{over all} \\ S}} (\mathrm{Totals}\ (S))$$

9) If $\mathrm{TOTALS}\ (R)/((\mathrm{GRPCNT}(I) * (\mathrm{GRPCNT}(I) + 1))/\ 2) < \mathrm{THRESH}$ go to Step 15 otherwise go to Step 10.

10) Increment the number of jobs in Subgroup I by 1.

$\mathrm{GRPCNT}(I) = \mathrm{GRPCNT}(I) + 1$

11) Record the R^{th} job to be in Subgroup I.

$$GROUP\ (I,\ (GRPCNT(I) + 1) = R$$

12) Set TOTAL = TOTALS(R).

13) Compute the following totals for all jobs S where S is subject to the following condition:

$$S \not\subset GROUP\ (K,\ L)$$

for all K and L where

$$1 \leq K \leq I$$

$$1 \leq L \leq GRPCNT(K)$$

$$TOTALS(S) = TOTAL + \sum_{M=1}^{GRPCNT(I)} S(S,GROUP(I,\ M))$$

14) Go to Step 8.

15) Done with subgroup I and ready to start the next subgroup. Therefore I = I + 1

16) Go to Step 3.

17) Allocate all unallocated jobs T to subgroup I. T is also subjected to the following:

$$T \not\subset GROUP\ (K,\ L)$$

for all K and L where

$$1 \leq K \leq I$$

$$1 \leq L \leq GRPCNT(K)$$

18) Fini. Stop the algorithm.

The threshold value THRESH and the maximum number of groups MXGRP have a direct impact on the composition of the jobs in the subgroup. The higher the threshold value, the more the number of subgroups formed. Consequently, the number of jobs in each subgroup are few. The few jobs in the large number of subgroups are highly similar to each other. The opposite is true as the threshold value is lowered.

The maximum number of sub-groups, MXGRP, and the threshold value, THRESH, have a direct relationship. Thus, their impact on the composition of jobs in subgroups is the same. Increasing the maximum number of subgroups has the same effect as increasing the threshold value.

The threshold value could be established several ways. One way is as follows:

$$\text{THRESH} = \left[\sum_{\text{all } j} \sum_{\text{all } i} \{\text{SIM}(i, j) / \text{SIM}(i, j) > C\}/E \right] \qquad 4.30$$

where E = E+1 for every case when S(i, j) > C and C is an arbitary constant value.

When C = 0 in Equation 4.30, the threshold value THRESH results in the mean value of all SIM(i,j)s. The value of C is determined subjectively based on the shop floor answer to the question "What minimal percent of similarity in total operations should two jobs have before being considered as likely candidates for grouping?". This value of C like the threshold value lies between 0 and 1. The higher the value of C, the higher the threshold value, and consequently there exists a strong similarity between jobs in a subgroup. When C is set equal to zero, the threshold value lowers because the average is now computed over both similar and dissimilar jobs. The similar overtones between subjectively establishing the threshold value and establishing the value of C cannot be neglected.

From a practical standpoint however, it is easier to estabish C than the threshold value itself because establishing the threshold value requires knowledge of the composition of all the jobs that need to be scheduled. This composition could vary from one scheduling period to another. Establishing the value of C requires knowledge of the composition of jobs within the part family. This composition is less volatile than the composition of jobs from period to period. Besides Equation 4.30, the standard statistical measures of mode and median are likely ways for establishing the threshold value.

a. Example To illustrate the algorithm consider a sample similarity matrix, as shown in Table 4.3. Assume a threshold value of 0.8. Looking at the matrix, the first pair of jobs in the first cluster is 1 and 2 with SIM(1,2) = 0.95. Set TOTAL = 0.95.

Using steps 7, 8 and 9 of the algorithm, the third job to join the first cluster is 3 because of the following:

$$1)\quad \text{TOTAL} + \text{SIM}(1,3) + \text{SIM}(2,3) = \max_{\text{all } j} \Big[(\text{TOTAL} + \text{SIM}(1,j) + \text{SIM}(2,j))\Big] \quad 1 \leq j \leq 8 \text{ but } \neq 1 \text{ or } 2$$

$$2)\quad (\text{TOTAL} + \text{SIM}(1, 3) + \text{SIM}(2, 3))/3 > 0.8$$

Set TOTAL = TOTAL + SIM(1, 3) + SIM(2, 3). Again, using steps 11, 12, 8 and 9 the next job in consideration to join cluster 1 consisting of jobs 1, 2 and 3 is job 4 because of the following:

$$1)\quad \text{TOTAL} + \text{SIM}(1,4) + \text{SIM}(2,4) + \text{SIM}(3,4) = \max_{\text{all } j} \Big[\text{TOTAL} + \text{SIM}(1,j) + \text{SIM}(2,j) + \text{SIM}(3,j)\Big] \quad 1 \leq j \leq 8 \text{ but } \neq 1, 2 \text{ or } 3.$$

2) (TOTAL + SIM(1,4) + SIM(2,4) + SIM(3,4))/6 = 0.81 which is > 0.80 the threshold value.

Set TOTAL = TOTAL + SIM(1,4) + SIM(2,4) + SIM(3,4). The next job in consideration to join cluster 1 consisting of jobs 1, 2, 3, and 4 is job 7 according to steps 13 and 8, since:

$$\text{TOTAL} + \text{SIM}(7,1) + \text{SIM}(7,2) + \text{SIM}(7,3) + \text{SIM}(7,4)$$
$$= \underset{\text{all } j}{\text{Max}} \left[\text{SIM}(1,j) + \text{SIM}(2,j) + \text{SIM}(3,j) + \text{SIM}(4,j) \right]$$
$$5 \leq j \leq 8$$

Table 4.3 Sample Similarity Matrix

	1	2	3	4	5	6	7	8
1	-	0.95	0.8	0.8	0.1	0.3	0.1	0.3
2	0.95	-	0.85	0.8	0.2	0.2	0.2	0.2
3	0.8	0.85	-	0.7	0.1	0.1	0.1	0.1
4	0.8	0.8	0.7	-	0.2	0.1	0.4	0.1
5	0.1	0.2	0.1	0.2	-	0.93	0.8	0.8
6	0.3	0.2	0.1	0.1	0.93	-	0.85	0.8
7	0.1	0.2	0.1	0.4	0.8	0.85	-	0.9
8	0.3	0.2	0.1	0.1	0.8	0.8	0.9	-

However, job 7 cannot join the first cluster since from step 9,

$$\left[\text{TOTAL} + \text{SIM}(1,7) + \text{SIM}(2,7) + \text{SIM}(3,7) + \text{SIM}(4,7) \right] / 10 < 0.8.$$

Therefore, no more jobs can join cluster 1 which results in cluster 1 containing jobs 1, 2, 3 and 4. Following steps 15, 16 and 3, in the second cluster, the first pair of jobs to form the nucleus is 5 and 6 since:

$$\text{SIM}(5, 6) = \underset{\substack{\text{all } j \text{ and } i \\ 5 \le i \le 8 \\ 5 \le j \le 8}}{\text{Max}} \{\text{SIM}(i, j)\} = 0.93$$

Set TOTAL = SIM(5,6). Job 7 joins this second cluster consisting of jobs 5 and 6 by virtue of the following from steps 7, 8 and 9.

1) $\text{TOTAL} + \text{SIM}(5,7) + \text{SIM}(6,7) = \underset{\text{all } j}{\text{Max}} \ (\text{TOTAL} + \text{SIM}(5,j) + \text{SIM}(6,j))$

2) $(\text{TOTAL} + \text{SIM}(5,7) + \text{SIM}(6,7))/3 = 0.86$ which is > 0.80

Set TOTAL = TOTAL + SIM(5,7) + SIM(6,7). Job 8 also joins the second cluster as per steps 11, 12, 8 and 9 since:

$(\text{TOTAL} + \text{SIM}(5,8) + \text{SIM}(6,8) + \text{SIM}(7,8))/6 = 0.85$ which is > 0.8

Thus given the similarity matrix in Table 4.3 and a threshold value of 0.80, the algorithm results in 2 clusters containing the following jobs.

Cluster 1 containing jobs 1, 2, 3 and 4.

Cluster 2 containing jobs 5, 6, 7 and 8.

It is interesting to note that had the threshold value been 0.82, the algorithm would have resulted in the following set of clusters, underscoring the impact that threshold values have on cluster formation.

Cluster 1 containing jobs 1, 2, and 3.

Cluster 2 containing jobs 5, 6, 7 and 8.

Cluster 3 containing job 4.

3. Subgroup sequencing

Having determined the subgroups, the order in which the subgroups are to be sequenced needs to be established. The order in which the subgroups are sequenced has an impact on the overall performance. Five rules were developed and tested based on the following.

1) Remaining slack in the subgroup.
2) Total processing time in the subgroup.
3) Total setup time in the subgroup.
4) Similarity between jobs in a subgroup.

a. GROUP 1 This rule is a function of the remaining slack in the subgroup and the total processing time for the subgroup. The first subgroup to be scheduled is G when the following is true.

$$\underset{\text{all } I}{\text{Min}} \sum_{J=1}^{GRPCNT(I)} [[DUE(GROUP(I, J)) - X_{ij}]/X_{ij}]$$

$$=$$

$$\sum_{J=1}^{GRPCNT(I)} [[DUE\ (GROUP\ (G,J)) - X_{gj}]/X_{gj}] \qquad 4.31$$

where X_{ij} and X_{gj} in the general case X_{rj} is given by

$$X_{rj} = \sum_{k=1}^{NOPN(GROUP(r,j))} [[PRSTIM\ (GROUP(r,j),K,1) * LOT(GROUP\ (r,j))] + STPTIM\ (GROUP\ (r,j),K,1)]$$

where $1 \leq I \leq NGRP$ and $1 \leq G \leq NGRP$

Equation 4.31 essentially assigns the highest priority to the subgroup that has the smallest total remaining slack to processing time ratio of all the jobs in the subgroup. There are two important differences between the procedure as defined in Equation 4.31 and that defined for operation scheduling as in Equation 4.19. Firstly, whereas in operation scheduling, the priority was assigned dynamically to the individual jobs during scheduling, in this procedure the assignment is made to all the jobs for all the operation in the subgroup even before scheduling takes place. Secondly, Equation 4.31 does not incorporate work in queue in remaining slack as in Equation 4.19.

b. GROUP 2 This rule assigns the highest priority to the subgroup that has the smallest total processing time. Thus subgroup G is assigned the highest priority if the following is true:

$$\underset{\text{all } I}{\text{Min}} \sum_{J=1}^{GRPCNT(I)} X_{ij} = \sum_{G=1}^{GRPCNT(G)} X_{gj} \qquad 4.32$$

where X_{ij} and X_{gj} in the general case X_{rj} is defined as for in Equation 4.31, and

where $1 \leq I \leq NGRP$
$1 \leq G \leq NGRP$

Equation 4.32 is akin to the shortest processing time rule applied at the group level.

c. GROUP 3 Rather than basing the decision on the remaining slack or the processing time, this rule bases it on the actual total setup time. This rule assigns the highest priority to the subgroup that has the largest total setup time for all the jobs in the subgroup. The rational for assigning priority in this manner, is that given jobs within subgroups are similar, the subgroup with the greatest total setup time offers the most potential for savings in setup time. This rule attempts to realize this potential the earliest. Under this rule, the subgroup G is given the highest priority such that the following is true.

$$\underset{\text{all } I}{\text{Max}} \sum_{J=1}^{\text{GRPCNT(I)}} \left\{ \sum_{k=1}^{\text{NOPN(GROUP(I,J))}} \text{STPTIM(GROUP(I,J)},k,1) \right\}$$

$$= \sum_{J=1}^{\text{GRPCNT(G)}} \left\{ \sum_{k=1}^{\text{NOPN(GROUP(G,J))}} \text{STPTIM(GROUP(G,J)},k,1) \right\} \qquad 4.33$$

where $1 \leq I \leq \text{NGRP}$
$1 \leq G \leq \text{NGRP}$

d. GROUP 4 This rule is similar to the above, except that priority is assigned to the subgroup that has the smallest total setup time for all jobs within the subgroup. Thus, subgroup G is assigned the highest priority when the following is true.

$$\min_{\text{all } I} \sum_{J=1}^{\mathrm{GRPCNT(I)}} \left\{ \sum_{k=1}^{\mathrm{NOPN(GROUP(I,J))}} \mathrm{STPTIM(GROUP(I,J),k,1)} \right\}$$

$$=$$

$$\sum_{J=1}^{\mathrm{GRPCNT(G)}} \left\{ \sum_{k=1}^{\mathrm{NOPN(GROUP(G,J)}} \mathrm{STPTIM(GROUP(G,J),k,1)} \right\} \qquad 4.34$$

where $1 \leq I \leq \mathrm{NGRP}$
$1 \leq G \leq \mathrm{NGRP}$

e. GROUP 5 This rule rather than be based on characteristics such as slack, processing time, setup time of jobs within a subgroup, bases the decision as a function of the similarity measure that determined the subgroups in the first place. The subgroup that has the highest mean similarity between all pairs of jobs within the subgroup is assigned the highest priority.

$$\max_{\text{all } I} \left[\left\{ \sum_{J=1}^{\mathrm{GRPCNT(I)-1}} \sum_{k=J+1}^{\mathrm{GRPCNT(I)}} \mathrm{SIM(GROUP(I,K),GROUP(I,J))} \right\} * 2/ [\mathrm{GRPCNT(I)} * \mathrm{GRPCNT(I-1)}] \right]$$

$$=$$

$$\left\{ \left[\sum_{J=1}^{\mathrm{GRPCNT(G)-1}} \sum_{k=J+1}^{\mathrm{GRPCNT(G)}} \mathrm{SIM(GROUP(G,K),GROUP(G,J))} \right] * 2/ [\mathrm{GRPCNT(G)} * \mathrm{GRPCNT(G-1)}] \right\} \qquad 4.35$$

where $1 \leq I \leq \mathrm{NGRP}$
$1 \leq G \leq \mathrm{NGRP}$

4. Job sequencing

The rules GROUP 1, GROUP 2, GROUP 3, GROUP 4, GROUP 5 establish the sequence of the subgroups. The procedure by which the individual jobs within the sequenced subgroups are to be sequenced is yet to be determined. The procedure is to establish the set of likely job candidates as defined in Equation 4.15 and when N(S{similar setup jobs}) > 1, the shortest processing time rule is used. When N(S{similar setup jobs}) = 0 the rule defined as in Equation 4.24 is used.

The next chapter briefly describes the logical structure of the simulation software used to investigate the procedures detailed in this chapter.

V. THE SIMULATION SOFTWARE

A. Birds-eye View

Rather than use standard simulation software, software based on standard FORTRAN was developed for the following reasons.

1) The difficulty of using pregenerated job data sets in standard simulation software.
2) The difficulty of simulating job-grouping techniques.
3) The possible need for using it on a stand alone minicomputer.
4) The need for providing a means for studying MRP-GT interactions.

The complete software consists of 3 modules functioning as independent programs. These 3 modules are the input module, the logic module and the output module. The relationship between the 3 modules and the flow of input and output information is shown schematically in Figure 5.1. The modularity provides the flexibility needed to simulate both operation scheduling approaches as well as job-group scheduling techniques.

The logic used is event oriented. Most event oriented software is triggered by two kinds of events, i.e., the arrival and the departure events. However, in the model developed, only one type of event is considered which is the departure type event. Further, the logic uses a modified approach to treating departure. Generally, the departure of a job from a machine triggers the next action to be taken which could be releasing another job from another machine, moving this job to the next operation etc. In this model, the same is accomplished indirectly. A departure of a job from a machine is translated to mean the availability of a machine. At this juncture, all machines are scanned to determine the next machine to be considered for scheduling. Thus, the event (the de-

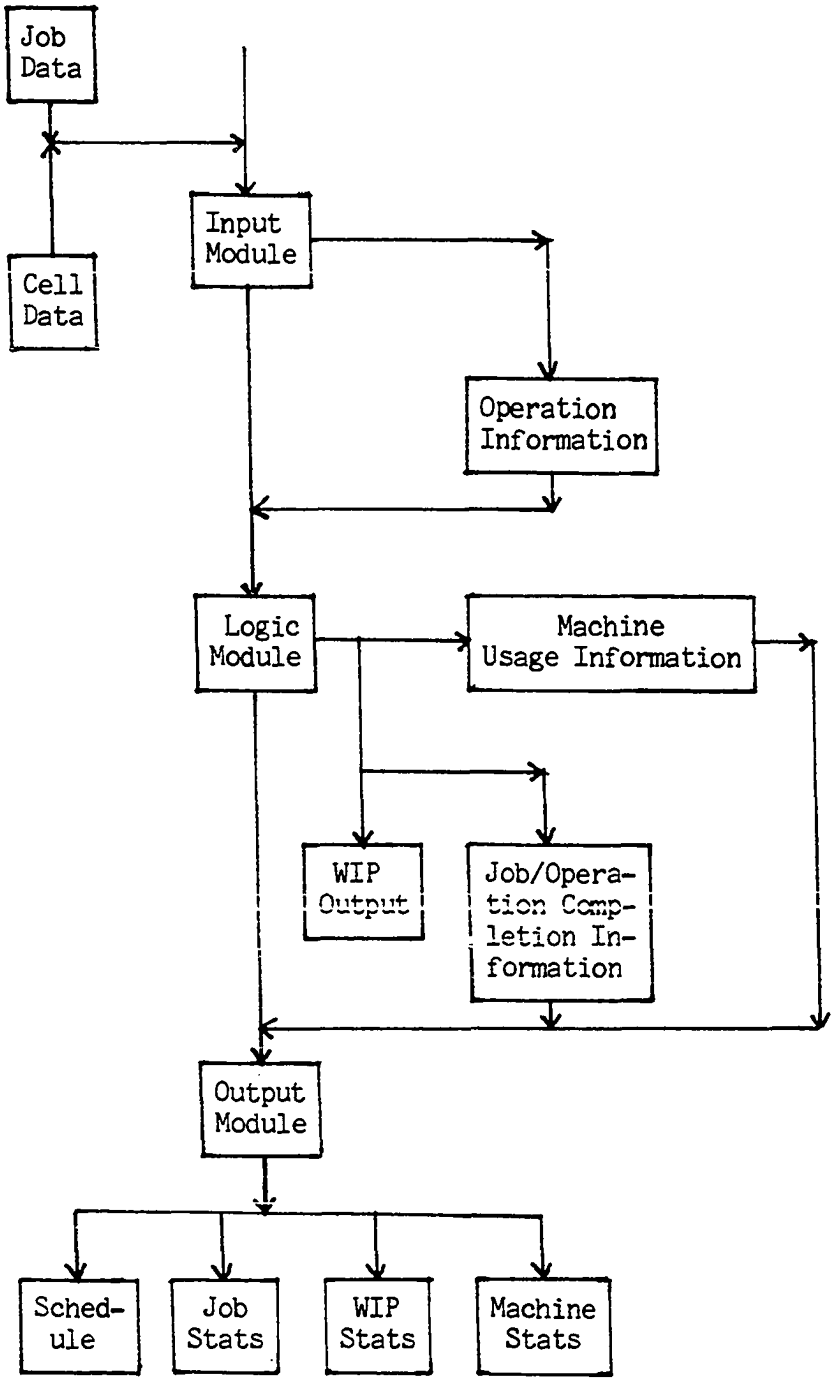


Figure 5.1 Overall Software Structure

parture) triggers the logic that determines which machine needs to be scheduled next.

The necessity of monitoring arrival type events is avoided by the generation of the operation information from the input module as shown in Figure 5.1. This is an important file of information kept for each alternate of each operation of each job. Every record contains the following information.

1) Job identification.
2) Operation identification.
3) Alternate identification.
4) Workcenter identification.
5) Machine number identification.
6) Setup class.
7) Status of operation.
8) Total processing time required.
9) Queue entry time.

The above essentially generates queues for all machines containing all possible jobs that can be processed at some time on the machine. Thus, jobs arrive at queues even before they are ready to be processed. The status indicator in the record helps classify whether the job is active in the queue and can be processed at the current time, or a job that has been processed, or a job that is likely to be processed.

By generating the operation file information two major benefits are realized.

1) The arrival process need not be monitored during simulation.

2) A central control pool of subsequent actions to be taken need not be maintained.

The disadvantage of course, is the generation of likely candidates in queues that might prove to be unnecessary during simulation. The memory utilized by these candidates is wasted. A more detailed discussion of the use of this operation information is left to section C in this chapter.

The primary purpose of the input module is to generate the operation file information. In the case of job-group scheduling, the subgroup determination is also performed in the input module.

The logic module performs the actual simulation and outputs raw statistical data on work-in-process, completed jobs, machine utilization etc. The output information from the input module facilitates simulation on principles akin to the Gantt chart thereby avoiding complicated logic to monitor several events and event chains as in GPSS.

The raw statistical data generated and stored on disk files in the logic module are analysed, and summarized in the output module. The following output is generated automatically.

1) Machine statistics.
2) Job statistics.

Optionally, the following can also be obtained.

1) Original job data set.
2) Input machine cell information.
3) WIP statistics.
4) Actual schedule generated.

B. The Input Module

The input module has two main functions.

1) Generate the operation information array.

2) In case of job-group scheduling determine the subgroups.

Figure 5.2 is an overall logic flow chart of the input module. For both operation scheduling and job-group scheduling, the information on the job set and the machines in the cell is the same. The following information is input as part of the cell information.

1) Number of workcenters and their codes.
2) Number of machines and their codes.
3) The capacity and efficiency of each machine within each workcenter.
4) The shifts that each machine within each workcenter is available.

As part of the job information the following is required.

1) The number of operations, the lot size, the due date, and the material cost.
2) Whether it is a make-to-stock or make-to-order job.
3) For each operation, the operation number and the number of alternates.
4) For each alternate of each operation of each job, the workcenter and the machine number on which it is to be processed, the setup and the processing time required, and the setup class.

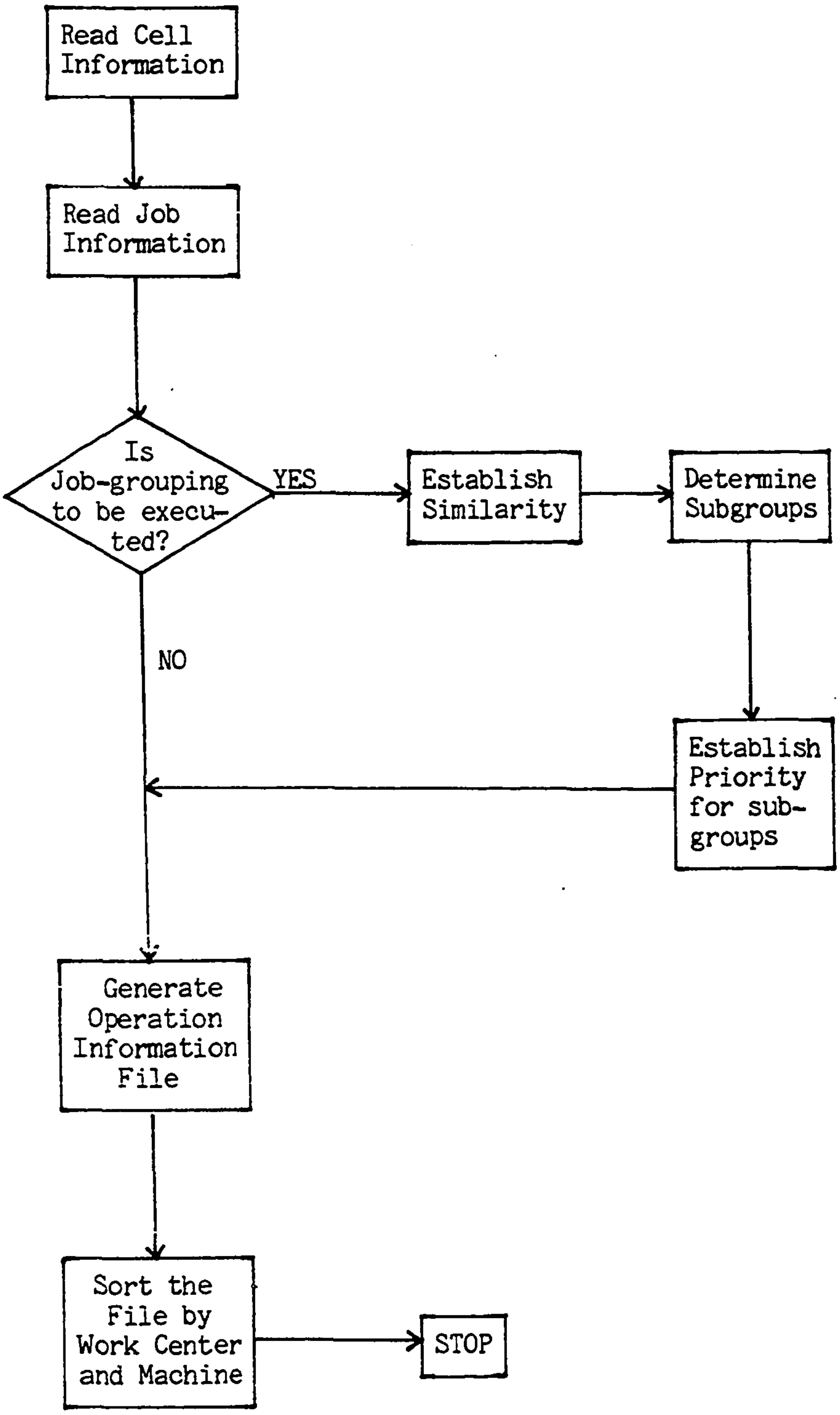


Figure 5.2 The Input Module

Based on the cell and job information the operation information file is generated. Each record in this operation file contains the following information for each alternate of each operation of each job.

1) The job identification number.
2) The operation identification number.
3) The alternate number.
4) The workcenter number.
5) The machine number within the workcenter.
6) Setup class.
7) The status of the operation/record.
8) The total time required to process the operation.
9) The entry time into the queue.
10) The priority, if any, of the operation.

To illustrate the generation of this important file, consider a hypothetical machine cell as shown in Table 5.1 and a job set detailed in Table 5.2. The setup class values for all alternates of all operations are estabished prior to processing of the input module. The logic for establishing the setup classes based on the GT code could be easily incorporated in the input module itself. For data processing convenience, it was established prior to the processing of the input module in the experimental analysis conducted.

Consider the first job shown in Table 5.2. The operation record for the first operation of this job is detailed in the first row of Table 5.3. The first field in the record is the job number which is 1 in this case. The second field is the operation number which is the first operation. Rather than use the actual numbers a transformation to sequential numbers

Table 5.1 Partial Machine Cell Information

Workcenter	Workcenter	Machine	Machine	Shift	Availability	
Number	Number	Number	Number	First	Second	Third
1	189	1	672	Y	N	Y
2	422	1	717	Y	Y	Y
2	422	2	718	Y	Y	Y
3	293	1	614	Y	N	Y
3	293	2	615	Y	N	Y
3	293	3	749	Y	Y	Y
4	180	1	754	Y	N	Y
5	469	1	1017	Y	Y	Y
6	431	1	1110	Y	Y	Y
7	188	1	1348	Y	Y	Y
8	295	1	1439	Y	Y	Y

Table 5.2 Sample Job Set

Job #	Lot Size	Due Date	# of Opns.	Opn. #	Alter-nate	Work Center	Mchn #	Setup Time	Process Time	Setup Class
1	22	3000	5	20	1	180	754	84.0	17.80	1
				40	1	189	672	94.0	15.10	1
				45	1	431	1110	7.0	5.41	1
				50	1	431	1110	11.0	4.60	1
				60	1	431	1110	18.0	2.98	1
2	1	2000	5	15	1	180	754	85.0	10.30	2
				40	1	189	672	97.0	19.00	1
				45	1	431	1110	19.0	4.71	2
				50	1	431	1110	14.0	4.08	1
				60	1	431	1110	18.0	2.96	1
3	25	2500	5	15	1	180	754	85.0	13.30	2
				40	1	189	672	95.0	11.90	3
					2	293	614	238.0	2.83	1
					2	293	615	238.0	2.83	1
					2	293	749	238.0	2.83	1
				45	1	469	1017	81.0	1.04	4
				50	1	422	717	34.0	1.09	1
					1	422	718	34.0	1.09	1
				60	1	422	717	32.0	0.89	2
					1	422	718	32.0	0.89	2
4	12	3000	5	20	1	180	754	68.0	11.00	1
				40	1	189	672	93.0	18.00	1
				45	1	469	1017	81.0	1.01	4
				50	1	422	717	25.0	1.22	1
					1	422	718	25.0	1.22	1
				60	1	422	717	32.0	0.91	1
5	1	2700	5	15	1	180	754	85.0	14.20	2
				40	1	184	672	94.0	14.00	3
					2	295	1439	328.0	4.65	1
				45	1	431	1110	16.0	8.49	1
				50	1	431	1110	11.0	4.96	1
				60	1	431	1110	18.0	3.26	1

Table 5.3 Operation File Information

Job #	Opn. #	Alt. #	Work Center	Mchn #	Set-up Class	Status	Total Time	Queue Entry Time	Priority
1	1	1	4	1	1	0	475.60	0	0
1	2	1	1	1	1	1	426.20	-1	0
1	3	1	6	1	1	1	126.02	-1	0
1	4	1	6	1	1	1	112.20	-1	0
1	5	1	6	1	1	1	83.56	-1	0
2	1	1	4	1	2	0	95.30	0	0
2	2	1	1	1	1	1	116.00	-1	0
2	3	1	6	1	2	1	23.71	-1	0
2	4	1	6	1	1	1	18.08	-1	0
2	5	1	6	1	1	1	20.96	-1	0
3	1	1	4	1	2	0	417.50	0	0
3	2	1	1	1	3	1	392.50	-1	0
3	2	2	3	1	1	1	308.75	-1	0
3	2	2	3	2	1	1	308.75	-1	0
3	2	2	3	3	1	1	308.75	-1	0
3	3	1	5	1	4	1	107.00	-1	0
3	4	1	2	1	1	1	61.25	-1	0
3	4	1	2	2	1	1	61.25	-1	0
3	5	1	2	1	2	1	54.33	-1	0
3	5	1	2	2	2	1	54.33	-1	0
4	1	1	4	1	1	0	200.00	0	0
4	2	1	1	1	1	1	309.00	-1	0
4	3	1	5	1	4	1	93.12	-1	0
4	4	1	2	1	1	1	39.64	-1	0
4	4	1	2	2	1	1	39.64	-1	0
4	5	1	2	1	1	1	43.02	-1	0
4	5	1	2	2	1	1	43.02	-1	0
5	1	1	4	1	2	0	99.20	0	0
5	2	1	1	1	3	1	108.00	-1	0
5	2	2	8	1	1	1	332.65	-1	0
5	3	1	6	1	1	1	24.49	-1	0
5	4	1	6	1	1	1	15.96	-1	0
5	5	1	6	1	1	1	21.26	-1	0

is adopted for ease in array manipulation in the logic module. The same is also adopted for workcenter number and machine numbers. The third field specifies the alternate number. The fourth field is the workcenter number which is 180. This is transformed to 4 using Table 5.1. The fifth field is the machine number which in likewise manner is transformed to 1. The sixth field is the setup class value which from Table 5.1 is 1 in this case. The seventh field is the status field. This field takes on 3 possible values as follows.

0 - If the operation can be currently scheduled.

1 - If the operation is a future candidate for scheduling.

-1 - If the operation has been completed either on this machine or on alternates.

Since this is the first operation the status is 0. The eighth field is the total operation time given by the sum of the setup time and the product of the process time and lot size. Hence

Operation time = 84.0 + 17.8 * 22 = 475.60.

The ninth field is the time the job entered the machine queue. In this case, since it is the first operation it is 0. For subsequent operations, a negative value is assigned to indicate that in reality the operations are yet to enter the queue. The tenth field is the priority field. In case of job-group scheduling, this would reflect the sequence number of the subgroup to which job 1 belongs.

In a likewise manner, Table 5.3 can be constructed for all the operations in all the jobs in the job set. This file is then sorted by the following major to minor fields all in ascending order.

Field 4 - Workcenter number.

Field 5 - Machine numbers.

Field 6 - Total operation time.

The sorted file as shown in Table 5.4 is used as the basic workfile in the logic module detailed in the next section.

C. The Logic Module

This module simulates the flow of orders through the cell. A traditional approach to shop simulation is to associate each order with a transaction and then "moving" this transaction through the various machines in the cell. This approach requires a complex sophisticated transaction maintenance, monitoring and event scheduling logic. This approach is used by popular simulation software languages such as SIMSCRIPT and GPSS. The advantage of this type of logical structure becomes evident if new job orders have to be generated while existing ones are being scheduled.

Rather than use conventional logic, a simpler radically different approach based on the Gantt chart is adopted. The "abjuration" of the conventional approach and the "genesis" of the approach used resulted from the following reasons.

1) A simpler (from a development viewpoint), more adaptable software was required.
2) In a production environment, the objective is to determine a schedule, given a set of orders to be scheduled. Additional orders arrive at discrete time intervals rather than on a continuous basis. This permits periodic scheduling, thereby

Table 5.4 Sorted Operation File

Rec- ord	Job #	Opn. #	Alt. #	Work Center	Mchn #	Set- up Class	Status	Total Time	Queue Entry Time	Pri- ority
1	5	2	1	1	1	3	1	108.00	-1	0
2	2	2	1	1	1	1	1	116.00	-1	0
3	4	2	1	1	1	1	1	309.00	-1	0
4	3	2	1	1	1	3	1	392.50	-1	0
5	1	2	1	1	1	1	1	426.20	-1	0
6	4	4	1	2	1	1	1	39.64	-1	0
7	4	5	1	2	1	1	1	43.02	-1	0
8	3	5	1	2	1	2	1	54.33	-1	0
9	3	4	1	2	1	1	1	61.25	-1	0
10	4	4	1	2	2	1	1	39.64	-1	0
11	4	5	1	2	2	1	1	43.02	-1	0
12	3	5	1	2	2	2	1	54.33	-1	0
13	3	4	1	2	2	1	1	61.25	-1	0
14	3	2	2	3	1	1	1	308.75	-1	0
15	3	2	2	3	2	1	1	308.75	-1	0
16	3	2	2	3	3	1	1	308.75	-1	0
17	2	1	1	4	1	2	0	95.30	0	0
18	5	1	1	4	1	2	0	99.20	0	0
19	4	1	1	4	1	1	0	200.00	0	0
20	3	1	1	4	1	2	0	417.50	0	0
21	1	1	1	4	1	1	0	475.60	0	0
22	4	3	1	5	1	4	1	93.12	-1	0
23	3	3	1	5	1	4	1	107.00	-1	0
24	5	4	1	6	1	1	1	15.96	-1	0
25	2	4	1	6	1	1	1	18.08	-1	0
26	2	5	1	6	1	1	1	20.96	-1	0
27	5	5	1	6	1	1	1	21.26	-1	0
28	2	3	1	6	1	2	1	23.71	-1	0
29	5	3	1	6	1	1	1	24.49	-1	0
30	1	5	1	6	1	1	1	83.50	-1	0
31	1	4	1	6	1	1	1	112.20	-1	0
32	1	3	1	6	1	1	1	126.02	-1	0
33	5	2	2	8	1	1	1	332.65	-1	0

eliminating the need in the simulation to dynamically generate orders, while other orders are being scheduled.

3) Relevant machine and job statistics can be collected after simulation from the schedule as opposed to dynamic collection while simulation proceeds. The dynamic collection necessitates more sophisticated logic.

The difference between the conventional approach and that used in this research is quadruple. Firstly, the input, logic, and output functions are independent segments. Secondly, rather than the completion of an operation on a job triggering processing logic, the availability of a machine triggers the processing logic. Thirdly, rather than maintain a common events chain or transaction pool where each transaction is associated with one job, each operation of each job is given a separate identity. Finally, many of the statistics are not generated during simulation but calculated independently after simulation.

The operation file information output in the input module is crucial to the processing logic. A sorted operation file of Table 5.3 is detailed in Table 5.4. Very simply, the processing logic can be viewed as building a Gantt chart. Each machine within each workcenter has a queue and available time slots when jobs can be processed. Such a scheme is shown in Figure 5.3. The queue consists of both active and inactive jobs. A job is active in a queue if it is ready to be scheduled. It is inactive if it has already been scheduled or is not ready to be scheduled yet. The operation file information contains the elements of the queue for all the machines.

With the queue established, the task abates to determining the earliest available machine. Once having chosen the machine, the queue behind the machine is scanned so as to choose a particular job based on the scheduling procedure. The job, so chosen, needs to be active at time of choice. The time required to process the particular operation of the chosen job is determined and alloted to the machine. Also the next operation for the job is determined and made active in the queue in which it resides while the current operation is made inactive in the current queue. The next available time for the chosen machine is changed to reflect the allotment of the current job. Following these updates, the next task is to choose the next earliest available machine and repeat the process. This process is repeated until all operations of all jobs have been scheduled.

The overall logical flow chart is shown in Figure 5.4. In the program, most of the functional blocks in Figure 5.4 are coded as quasi independent modules. Thus, future changes in any of the functional blocks can be easily accommodated. Functional block D in Figure 5.4 is a case in point. This segment chooses the job from the queue based on a rule and is contained in a subprogram. Thus, if programmed rules need to be changed or new ones accommodated this subprogram alone needs to be modified. Such modular interchangeability is true for most of the software in the logical module.

To illustrate the workings of the model, the data provided in Tables 5.1, 5.2 and 5.4 are used. For a scheduling rule, the simplest case of the shortest processing time is used. The reader would benefit more from the following if the explanation of the arrays OPNALL, MCHOP and MCHSTS

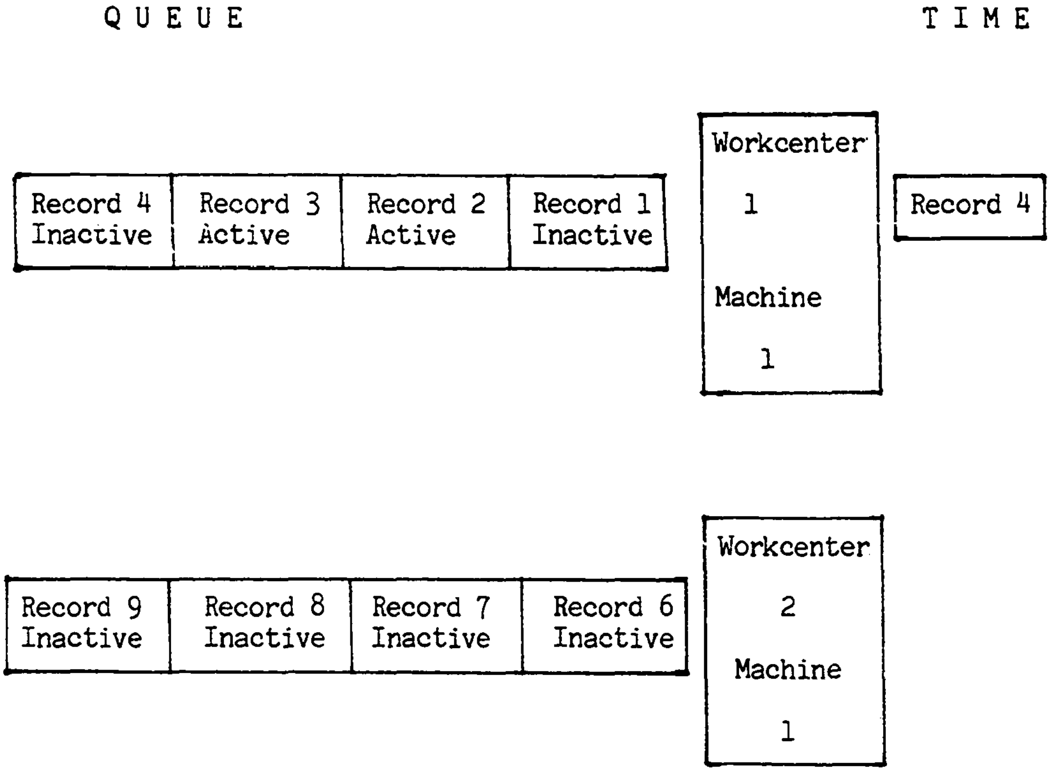


Figure 5.3 Logical Scheme of Machine/queues

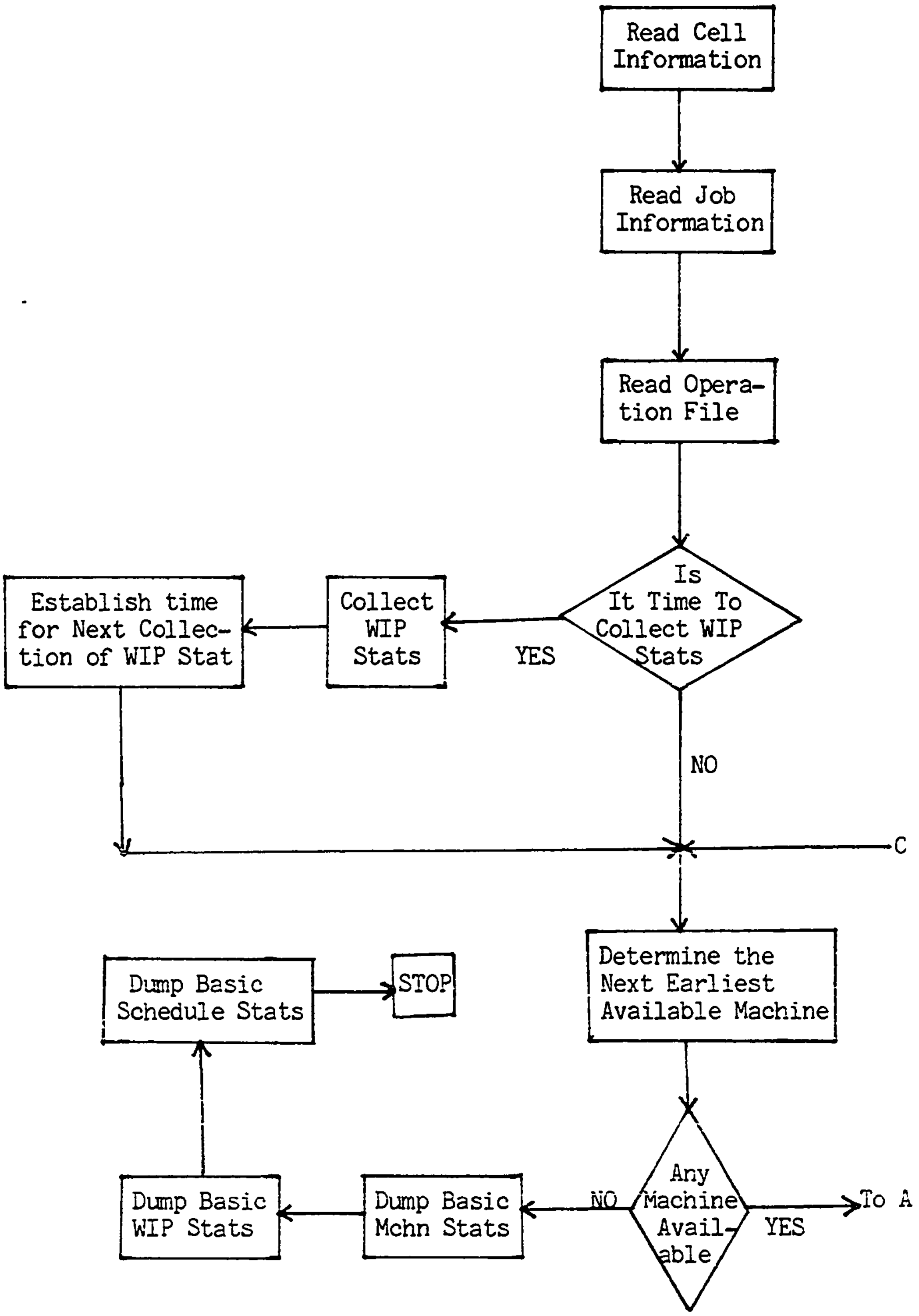


Figure 5.4 Overall Flowchart of Logic Module

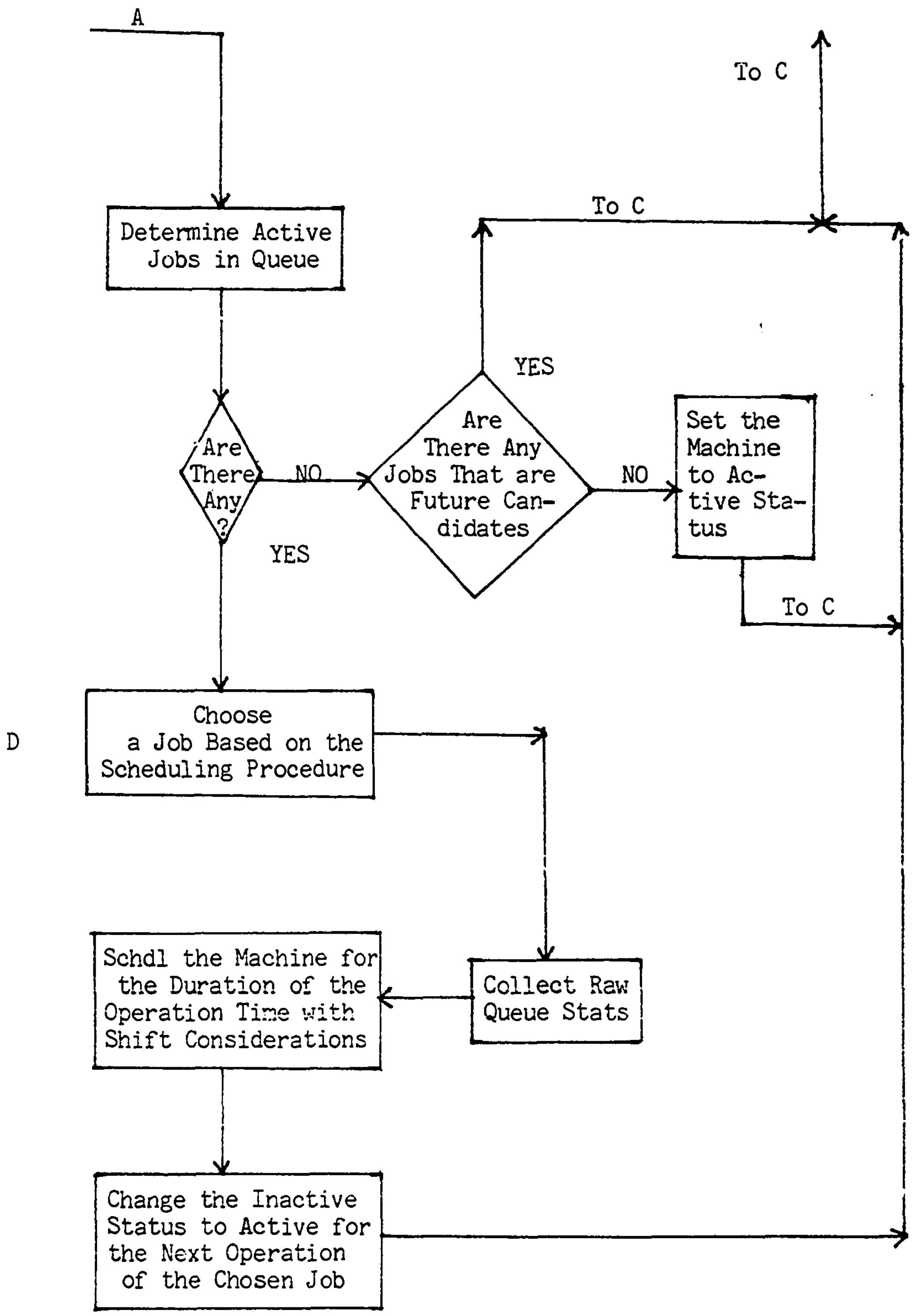


Figure 5.4 (continued)

given in Appendix A are reviewed.

1. Preliminary steps

The job information and all cell information are read. The operation file information is read into an array OPNALL. The starting record number and ending record number are recorded in array MCHOP (I, J, 1) and MCHOP (I, J, 2) respectively. For example, from Table 5.4 MCHOP (1, 1, 1) = 1, MCHOP (1, 1, 2) = 5, MCHOP (1, 2, 1) = 6, and MCHOP (1, 2, 2) = 9. All elements of the array MCHSTS are set to zero. The work in queue in front of each machine is computed and stored under the appropriate row vector. In the example, only the machine workcenter 4 has any work in queue. This work in queue is the sum of total operation times of all jobs in queue. This value is calculated from Table 5.2 and stored as shown in Table 5.5.

2. Step 1

The first machine available now is machine 1 in workcenter 1. However, since there are no jobs that can be scheduled in this machine, the next machine that is available, i.e., machine 1 of workcenter 2, is investigated. Again, there are no jobs ready to be scheduled and the next machine is chosen. Proceeding in this manner, machine 1 in workcenter 4 is chosen. Here, there are 5 jobs that can be scheduled and, based on SPT, job 2 is chosen. As a result of this, the MCHSTS array for the first machine in the fourth workcenter is updated as shown in Table 5.5. Since the first operation of job 2 is completed, the second operation is changed to an active status at time 95.30 when the first operation is completed. This change is reflected in the OPNALL array as shown in Table 5.6.

Table 5.5 Changes to MCHSTS Array

Step #	Work Cntr	Mch #	Last Set up Class	Next Avail-able	Operation Startup time	Total Setup Time	Total Time Used	Current Job #	Setup Sav-ings	Work in Queue
0	4	1	0	0	0	0	0	0	0	1287.50
1	4	1	2	95.3	0	85.0	95.3	2	0	1192.20
2	1	1	1	212.3	95.3	97.0	116.0	2	0	0
3	4	1	2	118.0	95.3	93.5	118.0	5	76.5	1093.0
4	4	1	1	318.0	118.0	161.5	318.0	4	76.5	893.0
5	8	1	1	350.65	118.0	328.0	350.65	5	0	0
6	6	1	2	226.01	212.3	19.0	23.71	2	0	18.08
7	6	1	1	244.09	226.01	33.0	41.79	2	0	20.96
8	6	1	1	248.85	266.09	34.8	46.55	2	16.2	0
9	1	1	1	543.40	318.00	106.3	341.3	4	83.7	0
10	4	1	2	735.50	318.0	246.5	735.50	3	76.5	475.60
11	6	1	1	360.74	350.65	36.4	56.64	5	30.6	15.96
12	6	1	1	365.80	360.74	37.5	61.70	5	40.5	21.26
13	6	1	1	369.86	365.80	39.3	65.76	5	56.7	0
14	1	1	3	1128.00	735.50	201.3	733.80	3	83.7	0
15	5	1	4	636.42	543.30	81.0	93.12	4	0	0
16	2	1	1	676.06	636.42	25.0	39.64	4	0	43.02
17	2	1	1	690.28	676.06	28.2	53.84	4	28.8	0
18	4	1	1	1211.11	735.5	330.5	1211.11	1	76.5	0
19	5	1	4	1162.10	1128.0	89.1	127.22	3	72.9	54.33
20	2	1	1	1192.75	1162.10	31.6	84.49	3	59.4	0

Table 5.5 (continued)

Step #	Work Cntr	Mch #	Last Set up Class	Next Avail-able	Operation Startup time	Total Setup Time	Total Time Used	Current Job #	Setup Sav-ings	Work in Queue
21	2	1	2	1247.08	1192.75	63.6	138.82	3	59.4	0
22	1	1	1	1637.71	1211.11	295.3	1160.40	1	83.7	0
23	6	1	1	1757.43	1637.71	40.0	185.48	1	63.0	112.20
24	6	1	1	1859.73	1757.43	41.1	287.51	1	72.9	83.56
25	6	1	1	1927.09	1859.73	42.9	354.87	1	89.1	0

Table 5.6 Changes to OPNALL Array

Step #	Rec-ord	Job #	Opn. #	Alt. #	Work Center	Mchn #	Set-up Class	Status	Total Time	Queue Entry Time	Pri or-ity
1	17	2	1	1	4	1	2	-1	95.20	0	0
1	2	2	2	1	1	1	1	0	116.00	95.30	0
2	2	2	2	1	1	1	1	-1	116.00	95.30	0
2	28	2	3	1	6	1	2	0	23.71	212.30	0
3	18	5	1	1	4	1	2	-1	99.20	0	0
3	33	5	2	2	8	1	1	0	332.65	118.0	0
3	1	5	2	1	1	1	3	0	108.00	118.0	0
4	19	4	1	1	4	1	1	-1	200.00	0	0
4	3	4	2	1	1	1	1	0	309.00	318.0	0
5	33	5	2	2	8	1	1	-1	332.65	118.0	0
5	1	5	2	1	1	1	1	-1	108.00	118.0	0
5	29	5	3	1	6	1	1	0	21.26	350.65	0
6	28	2	3	1	6	1	2	-1	23.71	212.30	0
6	26	2	4	1	6	1	1	0	18.08	226.01	0
7	25	2	4	1	6	1	1	-1	18.08	226.01	0
7	26	2	5	1	6	1	1	0	20.96	244.09	0
8	26	2	5	1	6	1	1	-1	20.96	244.09	0
9	3	4	2	1	1	1	1	-1	309.00	318.00	0
9	22	4	3	1	5	1	4	0	93.12	543.30	0
10	20	3	1	1	4	1	2	1	417.50	0	0
10	4	3	2	1	1	1	3	0	392.50	735.50	0
10	14	3	2	2	3	1	1	0	308.75	735.50	0

Table 5.6 (continued)

Step #	Record	Job #	Opn. #	Alt. #	Work Center	Mchn #	Set-up Class	Status	Total Time	Queue Entry Time	Priority
10	15	3	2	2	3	2	1	0	308.75	735.50	0
10	16	3	2	2	3	3	1	0	308.75	735.50	0
11	29	5	3	1	6	1	1	-1	24.49	750.65	0
11	24	5	4	1	6	1	1	0	15.96	360.74	0
12	24	5	4	1	6	1	1	-1	15.96	360.74	0
12	24	5	5	1	6	1	1	0	20.96	365.80	0
13	24	5	5	1	6	1	1	-1	20.95	365.80	0
14	4	3	2	1	1	1	3	-1	392.50	735.50	0
14	14	3	2	2	3	1	1	-1	308.75	735.50	0
14	15	3	2	2	3	2	1	-1	308.75	735.50	0
14	16	3	2	2	3	3	1	-1	308.75	735.50	0
14	23	3	3	1	5	1	4	0	107.0	1128.0	0
15	22	4	3	1	5	1	4	-1	93.12	543.30	0
15	6	4	4	1	2	1	1	0	39.64	636.42	0
15	10	4	4	1	2	2	1	0	39.64	636.42	0
16	6	4	4	1	2	1	1	-1	39.64	636.42	0
16	10	4	4	1	2	2	1	-1	39.64	636.42	0
16	7	4	5	1	2	1	1	0	43.02	676.06	0
16	11	4	5	1	2	2	1	0	43.02	676.06	0
17	7	4	5	1	2	1	1	-1	43.02	676.06	0
17	11	4	5	1	2	2	1	-1	43.02	676.06	0
18	21	1	1	1	4	1	1	-1	475.60	0	0

Table 5.6 (continued)

Step #	Rec-ord	Job #	Opn. #	Alt. #	Work Center	Mchn #	Set-up Class	Status	Total Time	Queue Entry Time	Pri-or-ity
18	5	1	2	1	1	1	1	0	426.60	1211.11	0
19	23	3	3	1	1	1	1	0	107.60	1128.0	0
19	9	3	4	1	2	1	1	0	61.25	1162.10	0
19	13	3	4	1	2	2	1	0	61.25	1162.10	0
20	9	3	4	1	2	1	1	-1	61.25	1162.10	0
20	13	3	4	1	2	2	1	-1	61.25	1162.10	0
20	8	3	5	1	2	1	2	0	54.33	1192.75	0
20	12	3	5	1	2	2	2	0	54.33	1192.75	0
21	8	3	5	1	2	1	2	-1	54.33	1192.75	0
21	12	3	5	1	2	2	2	-1	54.33	1192.75	0
22	5	1	2	1	1	1	1	-1	426.60	1211.11	0
22	32	1	3	1	6	1	1	0	126.02	1637.71	0
23	32	1	3	1	6	1	1	-1	126.02	1637.71	0
23	31	1	4	1	6	1	1	0	112.20	1757.43	0
24	31	1	4	1	6	1	1	-1	112.20	1757.43	0
24	30	1	5	1	6	1	1	0	83.56	1859.73	0
25	30	1	5	1	6	1	1	-1	83.56	1859.73	0

3. Step 2

From the set of available machines, it is seen that the earliest available machine is machine 1 in workcenter 1. There is one active job in the queue which is the second operation of the second job and so it is scheduled. This results in changes to the MCHSTS array as shown in Table 5.5 and changes to the OPNALL array as shown in Table 5.6.

4. Step 3

The earliest available machine is machine 1 in workcenter 4 which is available at 95.3 time units. Machine 1 in workcenter 1 is available only at time 212.3 and so is machine 1 in workcenter 6. The other machines in other workcenters although available earlier have no jobs in queue to be scheduled. From the jobs that can be scheduled on machine 1 in workcenter 4, job 5 is chosen. The necessary changes are reflected in Tables 5.5 and 5.6. Since the setup class on the machine is the same as the operation to be scheduled, the actual setup time is assumed to be only 10% of the standard setup time resulting in savings in setup time. Since the second operation of job 5 has two alternates, both the alternates are activated to active status as shown in Table 5.6.

5. Step 4

Scanning the set of available machines, the choice has to be made from machine 1 in workcenter 4, and machine 1 in workcenter 8 since they are both available at 118.0 time units. Arbitarily choosing machine 1 in workcenter 4, the first operation of job 4 is scheduled and the consequent changes are shown in Tables 5.5 and 5.6.

Proceeding along similar lines all the operations of all the jobs can be completed as shown in Tables 5.5 and 5.6. The throughput time from Table 5.5 is 1927.09. In the example explanation of the basic logic, the following steps that are considered in the actual model were neglected.

1) Every 440 time units (3 shifts) the work-in-process statistics are collected. The work in each queue, the total material cost of all jobs in queue, and the total labor cost incurred thus far by all the jobs in the various queues are determined and output to temporary files.
2) In the determination of completion times for operations on machines, consideration is given to whether the machine is available during a particular shift. This permits machines to be shut down for finite periods of time on a predetermined basis.
3) For computational efficiency, once an operation is completed it is moved to the end of the OPNALL array and the MCHOP arrays are adjusted. This reduces the number of rows to be scanned as simulation progresses.
4) In an array JOBSTS, the completion time as well as the actual setup time encountered for each job is monitored.

The raw WIP data are written to a temporary file as they are generated. After all operations on all jobs have been completed the following information is written to temporary files.

1) The JOBSTS array data.
2) For each machine in each workcenter:
    a) the total actual setup time on the machine,
    b) the setup savings realized on this machine,

c) and the total time the machine was utilized.

The above raw statistics are used in the output module to generate overall statistics. To generate the schedule as each operation is completed, basic information is written to another temporary file used by the output module to detail the actual schedule.

D. The Output Module

In the output module, overall machine and job statistics are derived using data generated from the logic module. The following statistical and schedule related information can be output from the module.

1) Original job data set.
2) Machine cell information.
3) Overall job statistics.
4) Overall machine cell statistics.
5) Work-in-process statistics.
6) Actual schedule.

Excepting overall job statistics and overall machine cell statistics, all the other output is generated only on user specification. The overall logical flow in this module is shown in Figure 5.5. A sample output may be found in Appendix B. The following provides a detailed description of the 6 output segments.

1. Original job data set

The following is output for every job:

1) Lot size.
2) Due time in minutes.

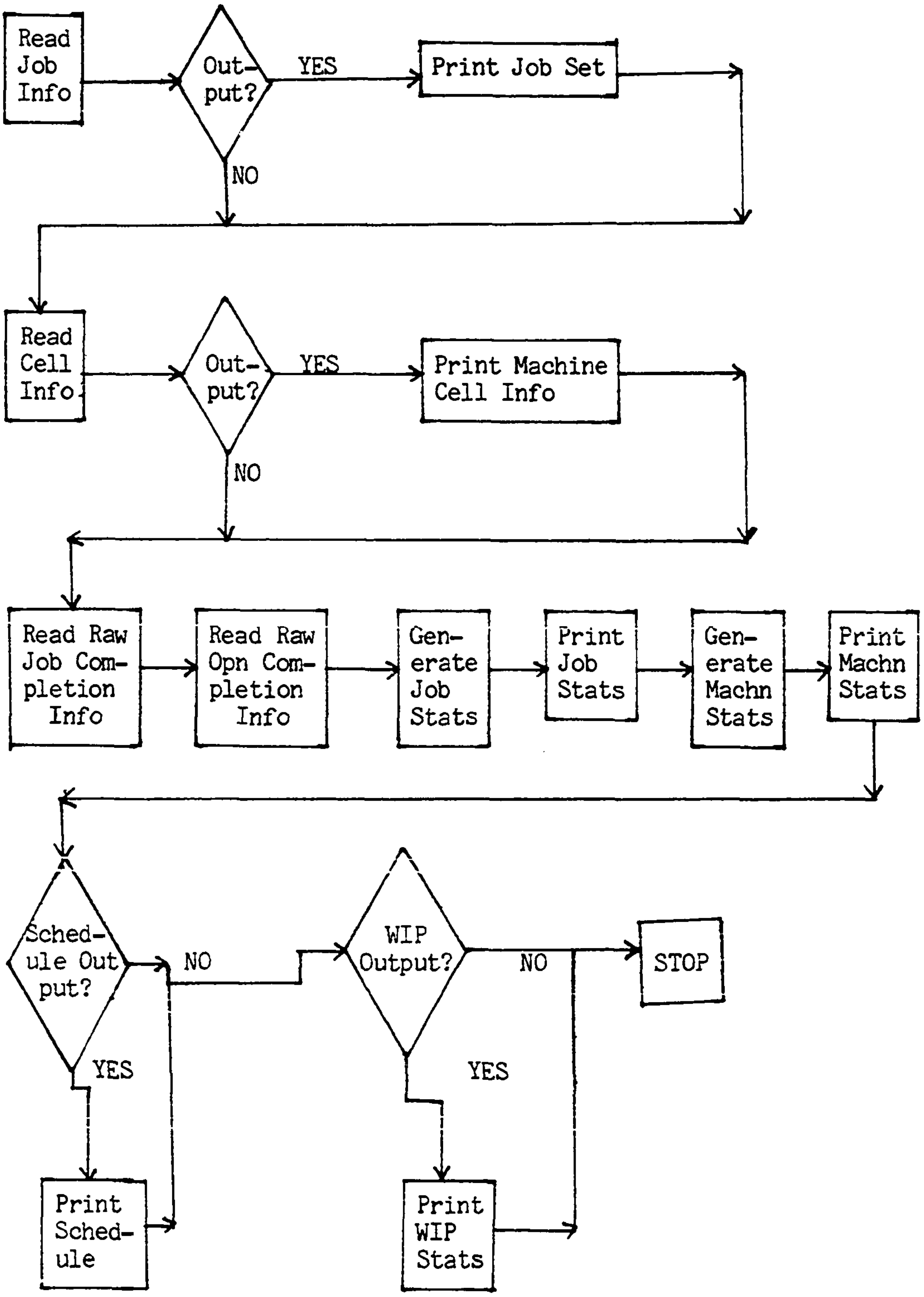


Figure 5.5 The Output Module

3) Whether it is a make-to-stock job, represented by 's', or make-to-order job, represented by 'n'.
4) Material cost for the lot.
5) For each operation:
    a) the operation number,
    b) the alternate number,
    c) the setup time in minutes for the entire lot,
    d) the process time/piece in minutes.

2. Machine cell information

The following is output for every machine in every workcenter.

1) Workcenter number.
2) Machine number.
3) Shift availability. A '1' indicates that it is available and a '0' that it is unavailable for that shift.

3. Overall job statistics

As part of the job statistics, the following is output. In all cases, the total for all operations of all jobs and the mean is also output.

1) Standard setup time. This is the total setup time of all jobs and average setup time per job as per the process routing sheets.
2) Actual setup time. This is the actual total setup time encountered for all the jobs in a particular job set, and the average per job.
3) Savings in setup time. This is the total savings in setup time realized due to the scheduling procedure, for all the jobs and

the average setup savings/job.

4) Cell resident time. This is the total time spent by all jobs in the cell, and average time spent by any job in the cell.
5) Waiting time. This is the total time each job spends in all the queues, and the average/job.
6) Waiting time/OPN. This is the waiting time for each operation of a job.
7) Lateness. This is the total and mean lateness per job. Lateness as defined here is the arithmetic sum of tardiness and earliness.
8) Tardiness. This is the total and mean tardiness for all the jobs.
9) Earliness. This is the total time by which the jobs that are completed early are ahead of schedule. The mean earliness is also provided.

In addition, the total number of jobs in the data set and the number of jobs delayed is also output.

4. Overall machine statistics

The following statistics are generated and output for all machines in all workcenters.

1) Total setup time: The total time the machine was used for setup purposes.
2) Total utilization: The total time the machine was actively in production. This includes both setup and processing.
3) Setup savings: The total savings in setup realized on this machine. This is difference between the standard setup time and actual setup time of all operation of all jobs processed on this

machine.

4) Queue length: This gives the mean queue length in front of the machine.
5) Mean time: This gives the average waiting time encountered in front of the machine.

5. Work-in-process statistics

At the end of every day in the simulation, a snapshot of the cell is taken and raw WIP data collected. On user request, the following is output at the end of each day.

1) Jobs finished: This is the number of jobs completed.
2) Work remaining: This is total amount of work remaining that is yet to be completed. This is the algebraic sum of the total standard operation time of all operations of all jobs.
3) Total material cost: This is the total material cost of all jobs being processed, or waiting to be processed.
4) Total labor cost: This is the total time expended thus far on all completed operations of all jobs being processed or waiting to be processed.
5) For each machine center the following is output:
    a) The actual queue length.
    b) The work in queue in minutes.

6. Job schedule

The schedule is printed by job number. For each job, the output lists the operation number, the workcenter number in which it is scheduled, and the individual machine number on which the operation will be processed. In addition the start and stop times are also output.

In the previous chapter the scheduling procedures were detailed. In this chapter the software used to simulate using the approaches is detailed. The next chapter details the results of simulating the procedures.

VI. ANALYSIS AND RESULTS

A. The Framework

1. Job sets

A total of 15 different job sets are randomly generated from a calling population of 102 different process plans. These process plans are actual process plans used for a cell in a manufacturing plant located in central Iowa. Thus, all data sets used in the simulation are based on actual data. The ratio of make-to-stock and make-to-order process plans to the total in the calling population is 0.55 and 0.45 respectively. The 15 different job sets are of the following composition.

1) Three different job sets in which the ratio of make-to-order jobs to the total in the job set is approximately 65%.
2) Three different job sets in which the ratio of make-to-order jobs to the total in the job set is approximately 70%.
3) Three different job sets in which the ratio of make-to-order jobs to the total in the job set is approximately 75%.
4) Three different jobs sets in which the ratio of make-to-order jobs to the total in the job set is approximately 80%.
5) Three different jobs sets in which there is no fixed ratio for the make-to-order jobs to the total in the job set.

2. Number of jobs

For each job set, the number of different jobs in the job set is randomly determined from a uniform distribution U(40, 65). Having determined the total number of jobs in the job set, the number of make-to-stock and make-to-order orders are calculated based on the ratio of make-to-order jobs to the total in the job set. In choosing individual jobs from the calling population, it is assumed that equal probabilities exist in the choice. However, the probability that a particular process plan will be chosen for the second time is one half the probability that it will be chosen for the first time. This reduces the probability of finding multiple orders for the same order in the job set.

3. Due date

The due date for the various jobs in the job set was assigned randomly based on an uniform distribution defined as U(6, 16), where the time is denoted in days.

4. Procedures tested

The 15 different job sets were all simulated, based on the following procedures:

1) Shortest processing time (SPT).
2) Six operation scheduling procedures.
    a) ONEFOR: A procedure that chooses a job from a queue as per Equations 4.16 and 4.17 when N(S{Similar setup job}) > 1 and Equation 4.24 when N(S{Similar setup jobs}) = 0.
    b) ONEFIV: A procedure that chooses a job from a queue as per Equations 4.16, and 4.17 when N(S{Similar setup jobs}) > 1 and Equation 4.25 when N(S{Similar setup job}) = 0.
    c) TWOFOR: A procedure that chooses a job from a queue as per Equations 4.19, 4.20 and 4.21 when N(S{Similar setup job}) > 1 and Equation 4.24 when N(S{Similar stup job}) = 0.
    d) TWOFIV: A procedure that chooses a job from a queue based

on Equations 4.19, 4.20 and 4.21 when N(S{Similar setup jobs}) > 1 and Equation 4.25 when N(S{Similar setup jobs}) = 0.

e) THRFOR: A procedure that chooses a job from a queue based on Equations 4.22 and 4.23 when N(S{Similar setup jobs}) > 1 and Equation 4.26 when N(S{Similar setup jobs}) = 0.

f) THRFIV. A procedure that chooses a job from a queue based on Equations 4.22 and 4.23 when N(S{Similar setup jobs}) = 0.

3) Five job-grouping procedures: These 5 procedures are duplicated in two sets. The first set consists of 5 procedures tested on subgroups formed by setting the threshold value THRESH as defined in Equation 4.30 where the value of C was assumed to be taken as 0.3. The second set consists of the same 5 procedures tested on subgroups formed by setting the threshold value THRESH as defined in Equation 4.30 plus a constant 0.2. The value of C is 0.3. The 5 procedures in the first set are denoted by G_1, G_2, G_3, G_4, and G_5, and those in the second set by G_6, G_7, G_8, G_9 and G_{10}.

a) G_1: A job-grouping procedure where the subgroups are formed as per Equation 4.30 with C = 0.3 and subgroup sequencing as per Equation 4.31.

b) G_2: A job-grouping procedure where the subgroups are formed as per Equation 4.30 with C = 0.3 and subgroup sequencing as per Equation 4.2.

c) G_3: A job-grouping procedure where the subgroups are formed as per Equation 4.30 with C = 0.3 and subgroup sequencing as per Equation 4.33.

d) G_4: A job-grouping procedure where the subgroups are formed as per Equation 4.30 with C = 0.3 and subgroup sequencing as per Equation 4.34.

e) G_5: A job-grouping procedure where the subgroups are formed as per Equation 4.30 with C = 0.3 and subgroup sequencing as per Equation 4.35.

f) G_6: A job-grouping procedure where the subgroups are formed as per Equation 4.30 plus 0.2, with C = 0.3 and subgroup sequencing as per Equation 4.31.

g) G_7: A job-grouping procedure where the subgroups are formed as per Equation 4.30 plus 0.2, with C = 0.3 and subgroup sequencing as per Equation 4.32.

h) G_8: A job-grouping procedure where the subgroups are formed as per Equation 4.30 plus 0.2, with C = 0.3 and subgroup sequencing as per Equation 4.33.

i) G_9: A job-grouping procedure where the subgroups are formed as per Equation 4.30 plus 0.2, with C = 0.3 and subgroup sequencing as per Equation 4.34.

j) G_{10}: A job-grouping procedure where the subgroups are formed as per Equation 4.30 plus 0.2, with C = 0.3 and subgroup sequencing as per Equation 4.35.

5. Criteria

The following job and machine related statistics are used for comparison purposes.

a) Throughput time.

b) Setup savings/job.

c) Cell resident time/job.

d) Wait time/job.

e) Lateness.

f) Tardiness.

g) Earliness.

h) Percent jobs late.

i) Cell utilization.

Cell utilization is defined as follows:

$$\text{Total utilization} = \sum_{i=1}^{\text{NMCH}} \sum_{j=1}^{\text{NUMMCH(I)}} \text{Utilization}_{ij}$$

$$\text{Mean utilization} = \frac{\text{Total utilization}}{\sum_{i=1}^{\text{NMCH}} \sum_{j=1}^{\text{NUMMCH(I)}} 1}$$

$$\text{Cell utilization} = \left(\frac{\text{mean utilization}}{\text{throughput time}}\right) \times 100$$

B. Operation Scheduling Results

Table 6.1 presents results for the shortest processing time and the six operation scheduling rules.

With respect to throughput time, all the operation scheduling procedures perform better than shortest processing time. The average throughput time for all the 6 procedures in only 91% of that for SPT. The range is from 81% for TWOFIV to 91% for THRFIV.

Table 6.1 SPT and Operation Scheduling

Operation Scheduling Results							
Criteria	SPT	ONEFOR	ONEFIV	TWOFOR	TWOFIV	THRFOR	THRFIV
Throughput	40208	38369	38691	32826	32377	38546	38752
Setup Savings	195	230	228	220	223	224	223
Cell Resident Time	4599	5711	7444	19556	18421	8495	8462
Waiting Time	2762	3940	5666	17774	16647	6711	6673
Average Lateness	10361	9249	7517	-4594	-3460	6263	6140
Tardiness	963	1343	2505	8690	7861	2731	2653
Earliness	11325	10593	10023	4095	4401	9199	9152
% Job Late	9.50	11.64	16.73	60.08	56.85	24.03	23.91

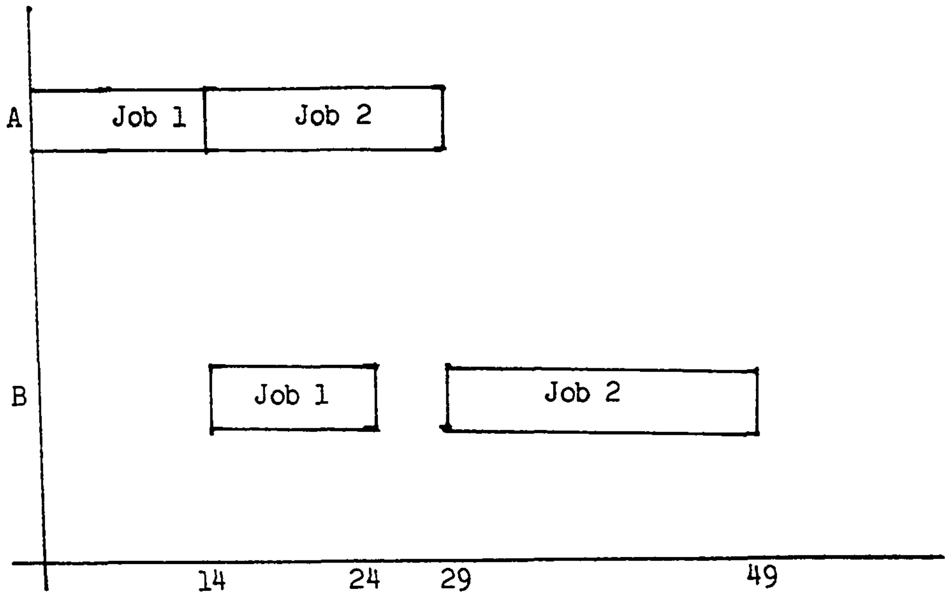
All six operation scheduling procedures perform better than SPT with respect to total savings in setup time. The improvement is approximately 15% per job.

However, the cell resident time is much higher. This is only to be expected. SPT by virtue of choosing the shortest processing time results in quick processing of several jobs at the start. This same effect can also be seen for waiting time. To illustrate this, consider an example of two jobs 1 and 2 with two operations to be done on two machines A and B in that order. Table 6.2 gives the visitation sequence and the total processing time for the operations.

Table 6.2 Example Job Set

Job #	Operation number			
	1		2	
	Machine	Time	Machine	Time
1	A	14	B	10
2	A	15	B	20

The Gantt chart using the SPT rule is shown in Figure 6.1. The throughput time, the cell resident time and the waiting time, are calculated from the Gantt chart data shown on the figure. Figure 6.2 is a Gantt chart for the same job set using the longest processing time rule. Comparing Figures 6.1 and 6.2, it is clear that all though the throughput time is greater for SPT, the cell resident time and waiting time/job are higher for LPT. From Table 6.1, a similar phenomenon can be observed in

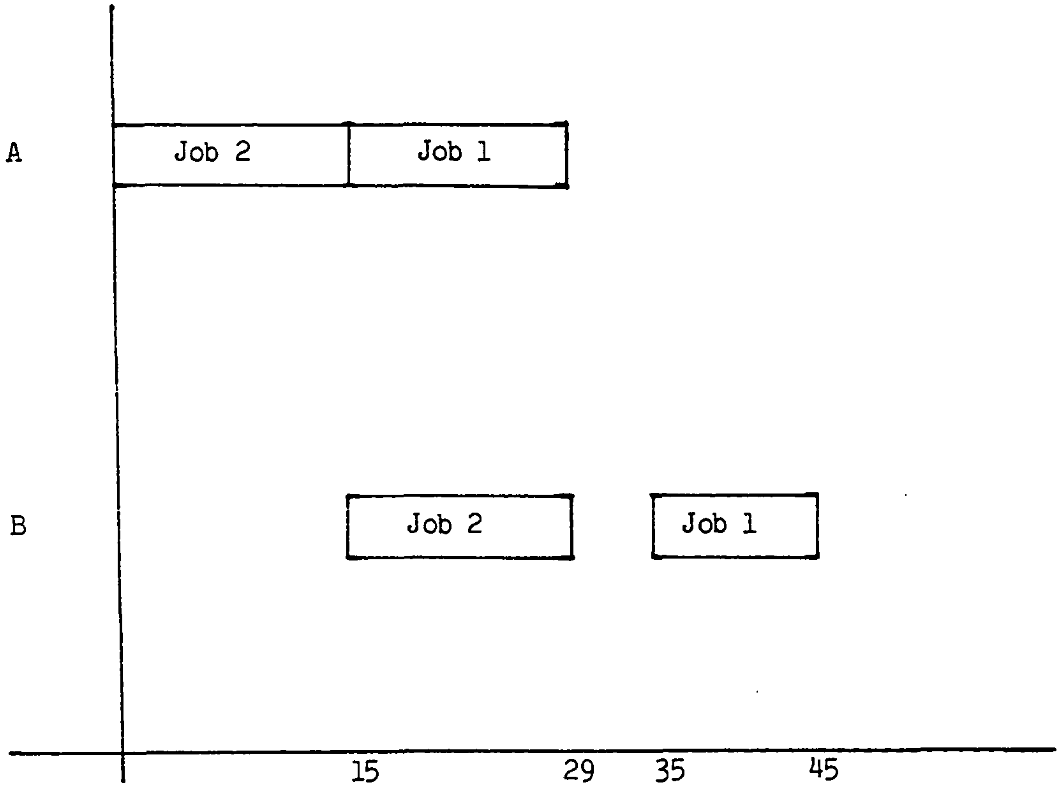


Throughput time = 49

Cell Resident Time/Job = (24 + 49)/2 = 36.5

Waiting Time/Job = (0 + 14)/2 = 7

Figure 6.1 Gantt Chart per SPT



Throughput time = 45

Cell Resident Time/Job = (35 + 45)/2 = 40

Waiting Time/Job = (21 + 0)/2 = 11.5

Figure 6.2 Gantt Chart per LPT

the case of operation scheduling procedures against SPT. Since the cell resident time is higher, it is not surprising that average lateness is lower and percent of jobs late is higher.

From Table 6.1, it is clear that the four best operation scheduling procedures are ONEFOR, ONEFIV, THRFIV, and THRFOR. The four best are chosen based more on a process of elimination rather than on a clear cut choice. From Table 6.1, it can be seen that the percent of jobs late for TWOFOR and TWOFIV are clearly out of normally acceptable range. This is so, in spite of having the lowest throughput time.

How do the four best compare against SPT? All the four operation scheduling procedures are better than SPT with respect to throughput time. They also realize greater savings in setup time per job. However, the cell resident time is higher which implies higher WIP oriented costs. Some of these determinental costs can be offset by make-to-stock orders which when completed under SPT lower WIP costs but increase finished goods costs. The reduction in throughput time and the increased savings in setup time will also help defray the higher WIP costs. Another important statistic is the percent of jobs late. Here again the operation scheduling procedures fare worse than SPT. In the worst case, the percent of jobs late is 24.1 for THRFOR against 10% for SPT. Assuming an average of 52 jobs per job set, the number of jobs late for SPT is 5 and for THRFOR it is 12. The savings in setup time for SPT is 10140 time units (52 x 195) and that for THRFOR it is 11648. Thus, the costs of 7 extra jobs delayed will have to be offset by setup savings of 1508 time units and overall reduction in throughput time of 1456 time units.

How does SPT fare against the best of operation scheduling procedures? In this case, an extra 1 job is delayed. However, there is a reduction in throughput time of 1839 time units and setup savings of 1820 time units.

Among the various operation scheduling procedures, ONEFOR seems to fare the best on all criteria except throughput time. The six procedures can be ranked in following order of preference based on Table 6.1.

1) ONEFOR
2) ONEFIV
3) THRFIV
4) THRFOR
5) TWOFIV
6) TWOFOR

In general, it seems that setup oriented procedures fare better than slack oriented approaches. It is interesting to note that the setup oriented approaches fare better than slack oriented approaches even in lateness, tardiness, earliness and percent of jobs delayed.

The only plausible explanation is the savings realized by choosing similar setup jos in the long run help earlier completion of jobs than does the choice of the "hottest" job. The choice of the "hottest" job seems to be shortsighted and detrimental in the long run. This hypothesis is again verified in the combination schemes THRFIV and THFOR which fare worse than ONEFOR or ONEFIV.

ONEFIV uses a setup oriented heuristic when N(S{Similar setup jobs}) > 1 and a slack oriented when N(S{Similar setup jobs}) = 0, TWOFOR uses a slack oriented heuristic when N(S{Similar setup job}) > 1 and a setup oriented heuristic when N(S{Similar setup jobs}) = 0. From Table 6.1, it is clear that ONEFIV is a far better procedure than is TWOFOR. This can be due to only one reason. The condition when N(S{Similar setup jobs}) > 1 is much more prevalent than the condition when N(S{Similar setup jobs}) = 0. It could be conceivably argued that the vice versa is true and consequently slack oriented approaches are better than setup oriented approaches. However, this argument is defeated since ONEFOR which is completely setup oriented fared the best and TWOFOV which is completely slack oriented fared the worst.

Another criterion that has not been discussed thus far is cell utilization. How does SPT and the operation procedures fare with regards to machine cell utilization. Table 6.3 details the utilization statistic.

Table 6.3 Cell Utilization

Procedure	Utilization %
SPT	15.02
ONEFOR	15.30
ONEFIV	15.20
TWOFOR	18.04
TWOFIV	18.25
THRFOR	15.30
THRFIV	15.26

Interestingly, the cell utilization does not vary significantly between various procedures. It is not surprising that TWOFOR and TWOFIV the slack oriented procedures have a slightly higher utilization. In both cases, the throughput time is lower without a significant decrease in total time that machines are actually processing the jobs. On the contrary, there will be an increase in this value due to lower savings in setup time compared to ONEFOR or ONEFIV.

The low utilization may raise eyebrows. However, there are logical reasons for this.

1) The average number of jobs in a job set is not large enough. Consequently, some machines are never used.
2) The throughput time was taken to be the time for machine shutdown. Consequently, a machine that has completed all operations in all possible jobs much earlier than the throughput time is still kept in an available state until the completion of throughput time.

Because of the relatively few different job sets for the various ratios of make-to-order orders to total orders, conclusive statements cannot be made. However, the limited experience indicated that the higher the number of make-to-order orders in the cell, the greater the savings in setup time and also the lower the numer of jobs delayed.

In summary, the following can be said with caution.

1) Operation scheduling procedures produce better results than SPT in setup savings and throughput time.
2) Not all operation scheduling procedures are effective.
3) Among operation scheduling procedures, the setup oriented ones fare better than the slack oriented approaches.
4) The machine cell utilization does not differ significantly between procedures, for underloaded cells.

C. Job-Group Scheduling Results

Tables 6.4 and 6.5, present the results for job grouping procedures G_1 to G_5 and G_6 to G_{10} respectively. From Tables 6.4 and 6.5 it can be seen that job-grouping procedures perform better than SPT with respect to the throughput time. The same is true for the average setup savings per job.

However, on every other criterion, the grouping techniques rate disfavorably to SPT. The key to reducing waiting time is to reduce the cell resident time. In job grouping techniques, the cell resident time is high because the procedure promotes the complete scheduling of a subgroup before it commences with the next. Thus, jobs in subgroups to be required later wait in the cell for a longer period of time. This consequently tends to increase the cell resident time per job. If the jobs belonging to a certain subgroup are not released until the first operation of any job in that subgroup is processed, then the cell resident time and the waiting time can be drastically reduced. In one set where this was attempted a 60% reduction in cell resident time/job was achieved.

In the generation of job sets, due dates for all jobs were assigned from the same distribution. In the formation of the subgroups, no consideration was given to the due dates. Thus, any job X would be in the same subgroup Y provided that SIM(X, Y) is close to 1. This is true even though X might be "hot" and Y with considerable slack. Now further, if this subgroup containing jobs X and Y is the last to be sequenced, job X will be delayed considerably. If this argument is extended to several jobs belonging to subgroups that are sequenced later, it partially answers the question "Why is % jobs delayed higher in job-group scheduling

Table 6.4 Job-Grouping Scheduling - Set 1

Criteria	SPT	G1	G2	G3	G4	G5
Throughput	40208	37691	40060	38911	36576	37964
Setup Savings	195	225	228	227	227	225
Cell Resident Time	4599	13536	7994	10284	14221	10100
Waiting Time	2762	11748	6187	8488	12434	8306
Average Lateness	10361	1426	6967	4677	4861	741
Tardiness	963	5144	2078	3087	3482	6101
Earliness	11325	6569	9045	7764	8343	6841
% Jobs Late	9.5	39.30	18.08	29.57	26.13	40.82

Table 6.5 Job-Group Scheduling - Set 2

Criteria	SPT	G6	G7	G8	G9	G10
Throughput	40208	37636	39728	37190	36936	38009
Setup Savings	195	227	225	217	228	216
Cell Resident Time	4599	12742	7300	10069	13738	9142
Waiting Time	2762	10941	5515	8243	11956	7330
Average Lateness	10361	2219	7661	4892	1223	5819
Tardiness	963	4436	1673	3299	5761	2635
Earliness	11325	6656	9335	8192	8454	6984
% Jobs Late	9.5	36.55	17.65	25.88	22.93	37.33

procedures?"

The discussion thus far regarding the performance of job-group scheduling techniques points to their limitation for use as a detailed level scheduling technique.

Even among the job-grouping techniques two questions need to be addressed.

1) Of what impact is the threshold value THRESH?
2) Is there any difference among the 5 subgroup sequencing techniques?

In Table 6.4, the subgroups were determined based on a threshold value THRESH defined as follows.

$$\mathrm{THRESH}_{\mathrm{Basic}} = \left| \sum_{\substack{\mathrm{all}\\ j}} \sum_{\substack{\mathrm{all}\\ i}} \{S(i, j) \mid S(i, j) > 0.3\}/E \right|$$

where $E = E + 1$ for every case when $S(i,j) > 0.3$

In Table 6.5 the threshold value THRESH is defined as follows.

$$\mathrm{THRESH} = \mathrm{THRESH}_{\mathrm{Basic}} + 0.2$$

In Table 6.5 the threshold value is higher than in Table 6.4, thereby resulting in more but tightly knit jobs within a subgroup. The higher threshold value gives consistently better results for all subgroup sequencing techniques and for most criteria. Encouraged by this, for a limited number of data sets the threshold equation was redefined as follows.

$$\mathrm{THRESH} = \mathrm{THRESH}_{\mathrm{Basic}} + 0.35$$

This did not give better results, than those provided in Table 6.5.

It is obvious that there is an "optimum" threshold value, though this might vary from job set to job set.

Among the five subgroup sequencing techniques, the technique that is a variation of SPT as defined in Equation 4.30 produces the best result. Surprisingly, the slack oriented schemes G_1 and G_6 do not fare well in job-group scheduling. This was also true in operation scheduling techniques.

Total setup time based on setup oriented procedures G_3, G_4, and G_9, G_{10} fared the worst as is evidenced in Tables 6.4 and 6.5. However, similarity based setup oriented schemes G_5 and G_{10} are the second best and lag behind the processing time oriented schemes G_2 and G_7.

Comparing G_2 and G_5 in Table 6.4 both schemes result in the same setup savings. However, the throughput time is lower for G_5. G_2 is better with respect to cell resident time and waiting times. This can be intuitively seen since G_2 is total processing time oriented. Similar to SPT, this scheme has a tendency to get more jobs completed at the very outset and delay time consuming subgroups to the very last. G_5 also has more jobs late than G_2. A similar discussion is applicable for G_7 and G_{10} in Table 6.5.

Table 6.6 gives the machine cell utilization statistics for SPT, G_2, G_5, G_7 and G_{10}. As in operation scheduling, there does not exist significant difference in machine cell utilization which would make a procedure better than the other.

In summary, job grouping techniques as used in this research, have limitations for use in detailed level scheduling. There is not a significance difference among the techniques with respect to overall machine cell utilization.

Table 6.6 Machine Cell Utilization

Rule	Utilization %
SPT	15.02
G_2	14.68
G_5	15.52
G_7	15.59
G_{10}	17.45

D. Operation Scheduling vs Job-Group Scheduling

Table 6.7 presents the results for two operation scheduling procedures and two job-group scheduling procedures. From Table 6.7, operation scheduling procedure ONEFOR gives the best results for the thruput time, savings in setup time and number of jobs delayed. In terms of machine utilization, there is not an appreciable difference between the various procedures.

In general, operation scheduling procedures fare better than job-group scheduling procedures for detailed level scheduling. This is particularly true in the areas of cell resident time, waiting time and tardiness. In terms of setup savings, there is not appreciable difference between ONEFOR and G7.

Although job-grouping scheduling as used in this research does take a second place to operation scheduling for detailed level scheduling, it offers much potential to general group scheduling within a GT cell.

Table 6.7 Operation Scheduling and Job-Group Scheduling

Criteria	ONEFOR	ONEFIV	G7	G10
Throughput	38369	38691	39728	38009
Setup Savings	230	228	225	216
Cell Resident Time	5711	7444	7300	9142
Waiting Time	3940	5666	5515	7330
Average Lateness	9249	7517	7661	5819
Tardiness	1343	3505	1673	2635
Earliness	10593	10023	9335	8454
Percent Jobs Late	11.64	16.73	17.65	22.93
Machine Utilization	15.30%	15.20%	15.59%	17.45%

In Figure 1.1, there are two levels of scheduling. The first level identifies subgroups of jobs that will result in minimal number of setup changes and the second level relates to detail level scheduling. Job-group scheduling is ideally suited to perform the first level via the similarity matrix and clustering technique presented in Chapter 4. For the second level, operation scheduling procedures ONEFOR and ONEIV offer potential. Thus, job-group scheduling in conjunction with operation scheduling provides a means to perform scheduling for a completely automated cell. Such a procedure will result in minimal setup changeovers.

In a likewise manner in an MRP system, orders to be released within a period of time can be analyzed for setup similarities and clustered together and scheduled in subgroups. This schedule then determines approximate release times and due dates. This schedule will also result in good savings in setup time. Based on the savings in setup time and the release times and due dates, the cost tradeoff decision can be made so as to enable when actual release will take place. Such an analysis is a quantative part of the MRP-GT interface. For a make-to-order order the clustering procedure helps determine the due date. This can be accomplished by clustering the existing orders with the new make-to-order order and determine the sub-group the make-to-order belongs to. Having established this, the time of completion of the last operation of the last job belonging to the subgroup helps determine an approximate due date for the make-to-order order.

VII. CONCLUSIONS

A. Research Highlights

1) Operation scheduling procedure ONEFOR and ONEFIV fare better than SPT in throughput time, savings in setup time, but not as well in flow time related measures. Other operation scheduling procedures do not fare as well as SPT.
2) Among operation scheduling procedures, those that are biased more towards setup similarities, i.e., ONEFOR, ONEFIV fare better than those that are biased towards slack like TWOFOR and TWOFIV. This deduction is strengthened since the combination schemes THRFOR and THRFIV fare better than the slack oriented schemes.
3) For further investigation of the practical limitations of using operation scheduling procedures for detailed level scheduling, ONEFOR and ONEFIV are recommended.
4) Job group scheduling procedures do not fare as well for detailed level scheduling as do operation scheduling procedures.
5) Job-group scheduling procedures are good for clustering jobs that are similar in setup. This preliminary subgrouping can then be combined with operation scheduling at the detailed level.
6) Combining both job-group scheduling and operation scheduling MRP-GT interactions can be analyzed particularly the cost tradeoff in preparing and postponing release times.
7) Job-grouping permits an approximate technique to establish due dates for make-to-order orders.

8) Job-grouping presents one way to establish orders to be released to a completely automated cell, with minimum setup changeover being a primary goal.
9) For underloaded cells, the overall machine cell utilization does not vary much between procedures.
10) The software developed is efficient and can be used to study MRP-GT interactions quantitatively.

B. Contributions

The following are the contributions of this research.

1) Setup oriented procedures have been developed and investigated for detailed level scheduling in a GT cell.
2) Two such procedures appear to offer potential for practical use.
3) A methodology to recognize setup similarity among jobs and subgroup them prior to scheduling has been developed using clustering analysis.
4) A procedure for scheduling parts in a completely automated cell with minimal setup changeover has been developed.
5) A relatively efficient and simple software has been developed for scheduling simulation and also for studying MRP-GT interactions. From experience in this research, 60 x 11 problems can be solved under 10 CPU seconds and problems up to 150 x 15 can be solved within estimated 512K region. The structure permits large scale problems to be solved in conjunction with direct access disk files. This same structure permits its adaptability to solve small problems on stand alone desk top computers with floppy disk capabilities.

C. Suggestions for Further Study

1) In job-group scheduling, the similarity identification was restricted to one attribute, the setup. In such an approach, jobs were subgrouped irrespective of their due dates. This resulted in a higher number of jobs being delayed. Future research should incorporate due date as an attribute and perform clustering based on both the attributes.
2) In this research little attention was paid to achieving a level load. In job-grouping, further research needs to be conducted to develop methods of tempering the subgroups so as to attain a level load.
3) The software needs to be expanded to provide interactive capabilities. This will aid in using it as a training device for shop personnel.
4) Using job-grouping approaches or better efficient approaches, MRP-GT interactions need to be studied.
5) In scheduling, a methodology needs to be developed that will measure the impact of a job order on the schedule with respect to cell performance. Such a processs will not only aid in GT cell scheduling but also in job-shop scheduling. The software developed coupled with job-group scheduling approaches provides through simulation a potential way for a crude determination of this impact.

6) In operation scheduling, a linear combination of slack and setup parameters was tested for detailed level scheduling. Although the research results were not encouraging, nonlinear schemes that bias more towards setup needs to be researched.

7) A limitation of this study is the limited number of data sets on which the procedures were tested, due to limitations in computer time and related resources. The inferences made from this research need to be statistically verified with a more comprehensive set of job data sets.

VIII. BIBLIOGRAPHY

1. Aggarwal, S. C. 1972. A Note on "The Influence of Setup Time on Job Shop Performance" by Willbrecht and Prescott. Management Science 19:1332-1333.

2. Ashour, S. 1967. A Decomposition Approach for the Machine Scheduling Problem. International Journal of Production Research 6:109:122.

3. Ashour, S. 1969. A Modified Decomposition Algorithm for Scheduling Problems. International Journal of Production Research 8:231-285.

4. Bae, J. M. 1972. A Heuristic solution to the general job shop scheduling problem. Unpublished Master's Thesis. Iowa State University, Ames, Iowa.

5. Baker, K. R. 1968. Priority Dispatching in the Simple Channel Queue with Sequence Dependent Setups. Journal of Industrial Engineering 9:203:206.

6. Barankin, E. W. 1952. The Scheduling Problem as an Algebraic Generalization of Ordinary Linear Programming. Discussion Paper No-18. Industrial Logistics Research Project, University of California, Los Angeles, California.

7. Bennet, M., and J. H. F. Sawyer. 1971. The Uses of a Simulation Model in Studying a Sequencing Problem in a Batch/Flow Environment. International Journal of Production Research 9:437:442.

8. Bulkin, M. H., J. L. Colley, and H. W. Steinhoff. Jr. 1966. Load Forecasting, Priority Sequencing and Simulation Job Shop Control Systems. Materials Science 13:B29:B51.

9. Carrie, A. S. 1973. Numerical Taxonomy Applied to Group Technology and Plant Layout. International Journal of Production Research 11:399-416.

10. Conway, R. W. 1964. An Experimental Investigation of Priority Assignment in a "Job Shop". Memorandum RM:3789. Rand Corporation, Santa Monica, California.

11. Conway, R. W., W. L. Maxwell, and L. W. Miller. 1967. Theory of Scheduling. Addison-Wesley, Cambridge, Massachusetts.

12. DeWitte, J. R. 1980. Use of Similarity Coefficients in Production Flow Analysis. International Journal of Production Research 18:503-514.

13. Eilon, S., and J. K. King. 1967. *Industrial Scheduling Abstracts.* Oliver and Boyd, Edinburgh, Scotland.

14. Evans, L. 1973. *Production Technology Advancements: A Forecast to 1988.* Industrial Development Division, Institute of Science and Technology, University of Michigan, Ann Arbor, Michigan.

15. Fisher, W. D. 1969. *Clustering and Aggregation in Economics.* Johns Hopkins Press, Baltimore.

16. Gallagher, C. C., and W. A. Knight. 1973. *Group Technology.* Butterworth, London.

17. Gavett, W. J. 1965. Three Heuristic Rules for Sequencing Jobs to a Single Production Facility. *Management Science* 11:B166-B176.

18. Gere, W. S. 1966. Heuristics in Job Shop Scheduling. *Management Science* 13:167-190.

19. Gupta, J. N. D. 1971. M-Stage Scheduling - A Critical Appraisal. *International Journal of Production Research* 9:267-281.

20. Gupta, J. N. D., and A. R. Maykut. 1973. Flow Shop Scheduling by Heuristic Decomposition. *International Journal of Production Research* 11:105-111.

21. Haque, A. A. 1976. *A Generalized Scheduling Technique for the Multimachines, Multiple Facilities System with an Application in Truck Dispatching*. Unpublished Ph.D. Dissertation. Iowa State University, Ames, Iowa.

22. Hitomi, K., and I. Ham. 1977. Group Scheduling Technique for Multi-product, Multistage Manufacturing Systems. *Journal of Engineering for Industry* 10:759-765.

23. Holloway, C. T., and R. T. Nelson. 1973. Alternative Formulation of the Job Shop Problem with Due Dates. *Management Science* 20:65-76.

24. Igwilo, F. N. 1979. *Group Technology Approach to Manufacture and Layout*. Unpublished Master's Thesis. Iowa State University, Ames, Iowa.

25. Kennedy, J. N. 1974. A Reveiw of Some Cluster Analysis Methods. *A.I.I.E. Transactions* 6:211-223.

26. Khator, S., and C. Moodie. 1979. A Machine Loading procedure which considers part complexity and machine capability. *International Journal of Production Research* 17:33-44.

27. King., J. R. 1980. Machine-Component Grouping in Production Flow Analysis. An Approach Using a Rank and Clustering Algorithm. *International Journal of Production Research* 10:213-232.

28. Le Grande, E. 1963. The Development of a Factory Simulation Using Actual Operating Data. *Management Technology* 3:1-19.

29. Mahendra, B., S. Arora. 1969. The Sequencing Problem. *Management Science* 16:247-263.

30. Merchant, M. E. 1970. Delphi-Type Forecast of the Future of Production Engineering. *CIRP Annals* 18:5-11.

31. McAuley, J. 1972. Machine Grouping for Efficient Production. *Production Management* 51:53-57.

32. Mellour, R. 1965. A Review of Job Shop Scheduling. *Operations Research Quarterly* 17:161-171.

33. Moore, J. M., and R. C. Wilson. 1967. A Review of Simulation Research in Job Shop Scheduling. *Production and Inventory Management* 8:100-118.

34. Nayeri, M. D. 1981. *Johnson - Type Algorithm for M-Machine Flow Shop*. Unpublished Ph.D. dissertation. Iowa State University, Ames, Ia.

35. Nelson, R. T. 1964. Direct Resource Constrained Series Service System. *Operations Research* 16:324-326.

36. Niscanci, H. I., and R. J. Sury. 1980. Production Analysis by Simulation in a Shoe Manufacturing Facility. *International Journal of Production Research* 18:31-42.

37. Page, E. S. 1961. An Approach to the Scheduling of Jobs on Machines. *Journal of the Royal Statistical Society* 23:484-693.

38. Pai, A. R., and K. L. McRoberts. 1971. Simulation Research in Interchangeable Part Manufacturing. *Management Science* 17:732-743.

39. Panwalker, S. S., and W. Islander. 1976. A Survey of Scheduling Rules. *Sequencing Research Report. No. QT-101-74.*, Texas Technical University, Lubbock, Texas.

40. Rajagopalan, S., and J. L. Batra. 1975. Design of Cellular Production Systems. A Graph-Theoretical Approach. *International Journal of Production Research* 13:567-579.

41. Rao, C. R. 1952. *Advanced Statistical methods in Biometric Research.* Hafner Publishing Company, Damien, Connecticut.

42. Ross, G. J. S. 1969. Algorithms AS13, AS14, and AS15. Applied Statistics 18:103-110.

43. Rowe, A. J. 1960. Toward a Theory of Scheduling. Journal of Industrial Engineering 11:128-137.

44. Ruhn, J., and A. P. Friedman. 1967. A Cluster Analysis and Taxonomy System for Grouping and Classifying Data. IBM Scientific Center, New York.

45. Sokal, R. R., and P. H. A. Sneath. 1963. Principles of Numerical Taxonomy. Freeman and Co., San Francisco.

46. Spencer, D. L. 1969. An Investigation of the Relationship Between Sequencing and Scheduling. Unpublished Ph.D. dissertation. Texas Technical College, Lubbock, Texas.

47. Spencer, M. S. 1980. Scheduling Components for Group Technology Lines. Production and Inventory Management 21:47-50.

48. Taylor, F. W. 1947. Scientific Management. Harper and Brothers, New York.

49. Wallace, C. 1947. The Gantt Chart. Sir Isaac Putman and Sons, London.

50. White, C. H., and R. C. Wilson. 1977. Sequence Dependent Setup Times and Job Sequencing. International Journal of Production Research 15:191-202.

51. Wilbrecht, J. K., and W. Prescott. 1969. The Influence of Setup Time on Job Shop Performance. Management Science 16:274-280.

52. Yamamato, M. 1977. An Approximate Solution of Machine Scheduling Problems by Decomposition Method. International Journal of Production Research 15:599-608.

IX. ACKNOWLEDGEMENTS

I would like to thank Dr. Sposito and Dr. Oldhehoeft for taking the time to serve on my committee. I sincerely appreciate the help of Merla Oppy in the typing of this manuscript, and Martha Keenan and Mary Kaufman in data preparation and keypunching.

My heartfelt gratitude goes to Dr. Jim Thomas for his immense help in this research effort. I sincerely thank all his colleagues, who gathered the data and helped me understand their manufacturing operation. Without Jim and his colleagues this effort would not have been possible.

I sincerely thank Dr. Malstrom and Dr. G. W. Smith for consenting to serve on my committee and for all their support, encouragement and advice throughout my graduate study. In this regard, I appreciate the efforts of Dr. Cowles, Dr. Meeks and Dr. Lamp. I extend my gratitude to Tony Bible, Mary Beth Liebernecht and Jim Hoekstra of the ISU Computation Center for their help in software.

I thank all the students and colleagues who taught me during the last 6 years, particularly Ray Blom, Pete Gheresus, and Brad Meyer.

To Dr. Berger, I owe a lot. He nourished my interest in graduate school and by virtue of his confidence in me helped me gain confidence in myself. More importantly, I would like to thank him and his family for being concerned friends and providing the emotional support, not to mention the great dinners and good times.

What I am today professionally and all my minimal accomplishments thus far have been due to Dr. McRoberts. He has taken a personal interest in me and expended much effort in my development during the last six

years. I am greatly indebted to him for this. All this effort, although sincerely apreciated by me, takes second place to what his entire family has done for me. They have been a surrogate family to me and helped me through difficult times during the last six years. Truly words cannot express my gratitude to all of them for this so I say simply hoping they will understand the depth: thank you to you all.

To my family in India I dedicate this effort. They taught me something that not all the books and schools can teach me -- what it is to be a decent human being. I would not be what I am as a person but for them.

Last but of greatest importance, I would like to thank God in whatever name He may be addressed and in whatever form He be worshipped. Without His help, protection and benevolence, nothing, absolutely nothing would have been possible.

X. APPENDIX A. DICTIONARY

The following describes the notation of the various arrays used. The reader should review this carefully, for it is used considerably in Chapters 4 and 5.

COMPOP(I)	Number of operations completed for the i^{th} job $1 \leq I \leq$ NJOBS
DUE(I)	the due date for the i^{th} job $1 \leq I \leq$ NJOBS
GROUP(I,J)	The job identification of the j^{th} entry in the i^{th} subgroup $1 \leq I \leq$ NGRP $1 \leq J \leq$ GRPCNT(I)
GRPCNT(I)	Number of jobs in the i^{th} subgroup $1 \leq I \leq$ NGRP
JOBSTS(I,J)	Basic statistical information on each job I, $1 \leq I \leq$ NJOBS
J = 1	Entry time into cell
J = 2	Actual setup time spent on the job
J = 3	Conpletion time of last operation of the job
LOT(I)	The lot size for the i^{th} job $1 \leq I \leq$ NJOBS
MCHNUM (I,J)	Machine number of the j^{th} machine in the i^{th} workcenter $1 \leq I \leq$ NMCH $1 \leq J \leq$ NUMMCH (I)
MCHOP (I,J,K)	The starting and stopping index values for array OPNALL for the j^{th} machine in the i^{th} workcenter. $1 \leq I \leq$ NMCH $1 \leq J \leq$ NUMMCH (I)
k = 1	Starting index
k = 2	Ending index

MCHSTS (I,J,K)	Specifies the various characteristics of the j^{th} machine in the i^{th} workcenter
$k = 1$	Last setup class
$k = 2$	Time when the machine is next available
$k = 3$	Time when the last operation started
$k = 4$	Total actual setup time on this machine
$k = 5$	Total time machine has been used thus far
$k = 6$	Job number of current job or last job
$k = 7$	Savings in setup time realized thus far on this machine
$k = 8$	Work in queue in front of machine
$k = 9$	Operation number of last job or current job
MXGRP	Maximum number of subgroups
NGRP	Number of subgroups
NJOBS	Total number of jobs to be processed
NMCH	Number of workcenters
NOPALT (I,J)	The number of alternates for the j^{th} operation of the i^{th} job $1 \leq I \leq NJOBS$ $1 \leq J \leq NOPN(I)$
NOPN(I)	Number of operations for the i^{th} job $1 \leq I \leq NJOBS$
NUMMCH(I)	Number of machines within workcenter I $1 \leq I \leq NMCH$
NWKCNT (I,J,K)	The workcenter number in which the k^{th} alternate of the j^{th} operation of the i^{th} job will be processed $1 \leq I \leq NJOBS$ $1 \leq J \leq NOPN(I)$ $1 \leq k \leq NOPALT\ (I,J)$
OPNALL(I,J)	An array that defines various attributes of each operation in each job visiting a particular machine.
$J = 1$	Job number

J = 2	Operation number for the above job
J = 3	Alternate number for the above operation
J = 4	Workcenter number that the above alternate will be processed in
J = 5	Machine number within the above workcenter
J = 6	Setup class of the above alternate
J = 7	Status of the operation 0 - ready to be processed 1 - not ready but a likely future candidate -1 - already completed the operation
J = 8	Total processing time required
J = 9	The time the job entered the queue in front of the machine
PRSTIM (I,J,K)	Process time of the k^{th} alternative of the j^{th} operation of the i^{th} job $1 \leq I \leq NJOBS$ $1 \leq J \leq NOPN(I)$ $1 \leq K \leq NOPALT(I,J)$
SIM(I,J)	Similarity measure between the i^{th} and the j^{th} job
STPTIM (I,J,K)	Setup time of the k^{th} alternate of the j^{th} operation of the i^{th} job $1 \leq I \leq NJOBS$ $1 \leq J \leq NOPN(I)$ $1 \leq K \leq NOPALT(I,J)$
QUE(I,J,K)	Job index of the k^{th} job in queue in front of the j^{th} workcenter $1 \leq K \leq QUELNG(I,J)$ $1 \leq J \leq NUMMCH(I)$ $1 \leq I \leq NMCH$
QUELNG(I,J)	The queue length of the j^{th} machine in the i^{th} workcenter $1 \leq I \leq NMCH$ $1 \leq J \leq NUMMCH(I)$

XI. APPENDIX B. SAMPLE OUTPUT

JOB INFORMATION

Lot Size	Due Time	Stk/ Non	Material Cost	Opern #	Alt #	Work Cntr	Setup Time	Process Time
6	18720	S	134.00					
				20	1	180	84.00	19.40
				40	1	189	92.00	20.70
				45	1	431	16.00	8.49
				50	1	431	11.00	4.96
				60	1	431	18.00	3.26
22	21600	S	43.80					
				20	1	180	84.00	17.80
				40	1	189	94.00	15.10
				45	1	431	7.00	5.41
				50	1	431	11.00	4.60
				60	1	431	18.00	2.98
14	12960	S	40.60					
				75	1	431	11.00	4.50
				80	1	431	18.00	2.76
1	14400	N	70.70					
				10	1	293	118.00	1.57
				15	1	180	70.00	11.50
				20	1	293	220.00	2.67
				40	1	189	97.00	11.70
				45	1	431	16.00	5.03
				50	1	431	11.00	4.58
				60	1	431	18.00	2.96
5	11520	S	151.10					
				20	1	180	73.00	23.50
				40	1	189	74.00	16.70
				45	1	431	7.00	11.40
				50	1	431	11.00	5.02
				60	1	431	18.00	3.19
22	21600	S	43.80					
				20	1	180	84.00	17.80
				40	1	189	94.00	15.10
				45	1	431	7.00	5.41
				50	1	431	11.00	4.60
				60	1	431	18.00	2.98

MACHINE CELL INFORMATION

FACILITY	MACHINE	SHIFT AVAILABILITY		
		SHIFT #1	SHIFT #2	SHIFT #3
189	672	1	0	1
422	717	1	1	1
422	718	1	1	1
293	614	1	0	0
293	615	1	0	0
293	749	1	1	1
180	754	1	0	1
469	1017	1	1	1
431	1110	1	1	1
188	1348	1	1	0
295	1439	1	1	1

OVERALL JOB STATISTICS

CRITERIA	TOTAL	MEAN
Standard setup time	21095	340.000
Actual setup time	6013	96.000
Savings in setup time	15082	243.000
Cell resident time	1961661	31639.000
Waiting time	1807498	29153.000
Waiting time/opn	361426	5829.000
Average late/early	******	-17564.000
Lateness	1317946	21257.000
Earliness	228925	3692.000

The number of jobs is	62
The number of jobs delayed is	44
The percent of jobs delayed is	70.97

OVERALL MACHINE STATISTICS

Wkont #	Mchn #	Total Setup Time	Total Utlzn	Saving Setup Time	Mean # in Queue	Mean Waiting Time
189	672	411	28497	2929	13	18652.72
422	717	78	13358	408	0	0.00
422	718	121	12046	374	1	4032.33
293	614	1211	5609	1740	0	98.93
293	615	673	4285	725	1	4675.00
293	749	381	2018	1349	1	2785.08
180	754	456	24772	2811	26	18735.98
469	1017	177	3501	888	2	8441.07
431	1110	203	7750	1600	0	190.43
188	1348	39	402	358	1	176.78
295	1439	2263	4204	1634	0	215.47

INTERMEDIATE WIP STATISTICS

Current Time	Jobs Fnsh	Total Work Remaining	Total Matl Cost	Total Labor Cost	Mach Cntr	Queue Length	Work in Queue
0	0	70011	51678.00	0.00			
					189	0	0
					422	0	0
					293	9	3500
					180	30	7291
					469	0	0
					431	12	396
					188	2	125
					295	7	1922
1440	13	55299	33332.00	19382.00			
					189	2	370
					422	0	0
					293	1	1130
					180	20	8024
					469	0	0
					431	17	698
					188	0	0
					295	1	2330
2880	31	46041	26023.00	24535.00			
					189	0	0
					422	0	0
					293	0	6595
					469	0	0
					431	9	600
					188	0	0
					295	1	269
4320	42	41418	18829.00	23428.00			
					188	1	267
					422	1	41
					293	1	213
					180	9	5310
					469	0	0
					431	1	129
					188	0	0
					295	0	0

THE SCHEDULE

COL 1 -- JOBID
COL 2 -- OPERATION NUMBER
COL 3 -- MACHINE CENTER
COL 4 -- MACHINE #
COL 5 -- START TIME
COL 6 -- ENDTIME

1	20	180	754	22998	23122
1	40	189	672	48797	48930
1	45	431	1110	48937	48988
1	50	431	1110	48988	49018
1	60	431	1110	49018	49038
2	20	180	754	27176	27575
2	40	189	672	50362	50703
2	45	431	1110	53662	53781
2	50	431	1110	53781	53883
2	60	431	1110	53883	53949
3	75	431	1110	22	86
3	80	431	1110	86	125
4	10	293	749	3421	3433
4	15	180	754	21852	21870
4	20	293	614	21870	23052
4	40	189	672	47049	47069
4	45	431	1110	47069	47075

XII. APPENDIX C. PROGRAM LISTING

```
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
     *)
C
       COMMON /TWOA/ NJOBS,NOPN(75),LOT(75),DUE(75),MATCST(75),CPNUM(75,
     *10),STPTIM(75,10,2),PRSTIM(75,10,2),STPCLS(75,10,2),NOPALT(75,10)
     *,NWKCNT(75,10,2)
C
C
      COMMON /TWOB/ STNST(75)
C
C
      COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),COMPOP(75),Q
     *UELNG(10,3)
C
      COMMON /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
       CHARACTER*4 MCHCDS,MCHNUM
       CHARACTER*1 MCHDES,MANAUT
C
      CHARACTER*1 STNST
      CHARACTER*4 WCNT(3)
C
C
       INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
C
      INTEGER NJOBS,NOPN,LOT,DUE,OPNUM,STPCLS,NWKCNT,NOPALT
C
       INTEGER OPNALL,JOBOP,COMPOP,MCHOP,QUELNG
      INTEGER MCHSTS,JOBSTS
C
      REAL STPTIM,PRSTIM,MATCST
C
      DOUBLE PRECISION DSEED
C
```

```
C
      DSEED = 123457.D0
      DO 10 I = 1,10
           DO 20  J = 1,3
                DO 30 K = 1,9
                     MCHSTS(I,J,K) = 0
 30             CONTINUE
 20        CONTINUE
 10    CONTINUE
      CALL CELINF
       CALL JOBINF
      CALL REDCPN(NJOBS)
      CALL SPTSCH(DSEED)
      STOP
      END
      SUBROUTINE CELINF
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
      *)
 C
 C
        CHARACTER*4 MCHCDS,MCHNUM
        CHARACTER*1 MCHDES,MANAUT
 C
 C
        INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
 C
 C
        READ(11,7000) NMCH
        DO 10 I=1,NMCH
               READ(11,7010) NUMMCH(I),MCHCDS(I)
                I1 = NUMMCH(I)
               DO 20 J=1,I1
```

```
                    READ(11,7020) MCHNUM(I,J),(MCHDES(I,J,K),K=1,35),MCH
     *CAP(I,J),(MCHAVL(I,J,K),K=1,3),MCHEFF(I,J),MANAUT(I,J)
 20             CONTINUE
 10       CONTINUE
 7000       FORMAT(I2,78X)
 7010       FORMAT(I1,2X,A4,73X)
 7020       FORMAT(A4,2X,35A1,2X,I2,3I2,I3,A1,25X)
C
C
         RETURN
         END
       SUBROUTINE JOBINF
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
     *)
C
       COMMON /TWOA/ NJOBS,NOPN(75),LOT(75),DUE(75),MATCST(75),CPNUM(75,
     *10),STPTIM(750,10,2),PRSTIM(75,10,2),STPCLS(75,10,2),NOPALT(75,10)
     *,NWKCNT(75,10,2)
C
C
      COMMON /TWOB/ STNST(75)
C
C
      COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),COMPOP(75),Q
     *UELNG(10,3)
       CHARACTER*4 MCHCDS,MCHNUM
       CHARACTER*1 MCHDES,MANAUT
C
C
      CHARACTER*1 STNST
      CHARACTER*4 WCNT(3)
C
```

```
 C
        INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
 C
 C
       INTEGER NJOBS,NOPN,LOT,DUE,OPNUM,STPCLS,NWKCNT,NOPALT
 C
        INTEGER OPNALL,JOBOP,CCMPCP,MCHOP,QUELNG
 C
       REAL STPTIM,PRSTIM,MATCST
 C
 C
       READ (12,7030) NJOBS
       DO 30 I=1,NJOBS
           READ(12,7040) NOPN(I),LOT(I),DUE(I),STNST(I),MATCST(I)
           L=NOPN(I)
           JOBOP(I,1) = MATCST(I)
           JOBOP(I,2) = 0
           DO 40 J=1,L
               READ(12,7050) OPNUM(I,J),NOPALT(I,J)
                 I1 = NOPALT(I,J)
               DO 50 K=1,I1
                 READ(12,7060) WCNT(K),STPTIM(I,J,K),PRSTIM(I,J,K),STPCLS
     *(I,J,K)
                  DO 60 M=1,NMCH
                      IF ((WCNT(K).EQ.MCHCDS(M))) THEN
                          NWKCNT(I,J,K)=M
                          GO TO 50
                      ENDIF
   60             CONTINUE
   50         CONTINUE
   40     CONTINUE
  30     CONTINUE
 C
 C
```

```
 7030  FORMAT(I3,77X)
 7040  FORMAT(32X,I2,2X,I4,2X,I6,2X,A1,F7.2)
 7050  FORMAT(I3,2X,I1,74X)
 7060  FORMAT(A4,2(2X,F7.2),2X,I2)
C
C
      RETURN
      END
C
      SUBROUTINE REDOPN(NJOBS)
      COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
      COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
     *)
C
      COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),COMPCP(75),Q
     *UELNG(10,3)
C
      COMMON /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
      CHARACTER*4 MCHCDS,MCHNUM
      CHARACTER*1 MCHDES,MANAUT
C
C
      INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
C
      INTEGER OPNALL,JOBOP,COMPCP,MCHOP,QUELNG
C
      INTEGER MCHSTS,JOBSTS
```

```
C
C
      DO 10 I=1,NJOBS
C         READ(3,7040) JOBOP(I,1),JOBOP(I,2)
          COMPCP(I)=0
  10  CONTINUE
          DO 5 I=1,NMCH
             I1=NUMMCH(I)
             DO 6 J=1,I1
                MCHCP(I,J,1)=0
                MCHOP(I,J,2)=0
                QUELNG(I,J) = 0
   6         CONTINUE
   5      CONTINUE
      DO 20 I=1,1200
         READ(14,7050,END=2000) (OPNALL(I,J),J=1,11)
         MCHOP(OPNALL(I,4),OPNALL(I,5),2)=I
         IF (OPNALL(I,7).EQ.0) THEN
             QUELNG(OPNALL(I,4),OPNALL(I,5))=QUELNG(OPNALL(I,4),OPNALL(I
     *,5))+1
C     PRINT*,'17110',OPNALL(I,4),OPNALL(I,5),QUELNG(OPNALL(I,4),OPNALL(I
C    *,5)),I
              MCHSTS(OPNALL(I,4),OPNALL(I,5),8) = MCHSTS(OPNALL(I,4),OPN
     *ALL(I,5),8) + OPNALL(I,8)
         ENDIF
         IF (I .EQ. 1) THEN
             MCHOP(OPNALL(I,4),OPNALL(I,5),1) = 1
             I1 = OPNALL(I,4)
             I2 = OPNALL(I,5)
          ENDIF
         IF (OPNALL(I,4) .NE. I1) THEN
              MCHOP(I1,I2,2) = I - 1
              MCHOP(OPNALL(I,4),OPNALL(I,5),1) = I
              I1 = OPNALL(I,4)
              I2 = OPNALL(I,5)
         ELSE
```

```
              IF (OPNALL(I,5) .NE. I2) THEN
                 MCHOP(I1,I2,2) = I - 1
                 I2 = OPNALL(I,5)
                MCHOP(I1,I2,1) = I
               ENDIF
           ENDIF
   20      CONTINUE
 2000  CONTINUE
       K=NUMMCH(1)
C7040  FORMAT(I4,2X,I4,70X)
 7050  FORMAT(I3,2X,6(I2,2X),2(I7,2X),2(I2,2X))
       RETURN
       END
      SUBROUTINE SPTSCH(DSEED)
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
     *)
C
       COMMON /TWOA/ NJOBS,NOPN(75),LOT(75),DUE(75),MATCST(75),OPNUM(75,
     *10),STPTIM(750,10,2),PRSTIM(75,10,2),STPCLS(75,10,2),NOPALT(75,10)
     *,NWKCNT(75,10,2)
C
C
       COMMON /TWOB/ STNST(75)
C
       COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),CUMPOP(75),Q
     *UELNG(10,3)
C
       COMMON /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
        CHARACTER*4 MCHCDS,MCHNUM
        CHARACTER*1 MCHDES,MANAUT
```

```
C
C
      CHARACTER*1 STNST
      CHARACTER*4 WCNT(3)
C
       INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
C
       INTEGER OPNALL,JOBOP,COMPCP,MCHOP,QUELNG
      INTEGER MCHSTS,JOBSTS
C
      INTEGER WIPTIM,FLGEND,NOJOB,IFAC,IMCH
C
      INTEGER NJOBS,NOPN,LOT,DUE,OPNUM,STPCLS,NWKCNT,NOPALT
C
C
      REAL STPTIM,PRSTIM,MATCST
C
       DOUBLE PRECISION DSEED
C
C
C     DO 10 I=1,NMCH
C        I1=NUMMCH(I)
C        DO 20 J=1,I1
C           DO 30 K=1,7
C               MCHSTS(I,J,K)=0
C 30        CONTINUE
C 20      CONTINUE
C 10  CONTINUE
      DO 40 I=1,NJOBS
          DO 50 J=1,3
              JOBSTS(I,J)=0
 50       CONTINUE
 40   CONTINUE
      IFAC=0
      ITIM=0
```

```
      FLGEND=0
      NOJOB=0
      WIPTIM=0
      IMCH=0
      READ (5,7000) IHRST,ITNSFR
 2000 CALL MCHSLC(FLGEND,NOJOB,IFAC,IMCH,ITIM)
C     PRINT*,'SPT',IFAC,IMCH,ITIM,FLGEND,WIPTIM
      IF (FLGEND.EQ.0) THEN
         IF (WIPTIM .LE. ITIM) THEN
              CALL WIPSTAT(WIPTIM)
C      PRINT*,'WIPSTAT,WIPTIM',WIPTIM
C             WIPTIM=WIPTIM+1440
         ENDIF
         CALL CPNSCH(FLGEND,NOJOB,IFAC,IMCH,ITIM,IHRST,ITNSFR,DSEED)
      ELSE
         GO TO 2010
      ENDIF
      GO TO 2000
 2010 DO 60 I=1,NJOBS
         WRITE(19,8020) JOBSTS(I,2),JOBSTS(I,1),JOBSTS(I,3)
   60 CONTINUE
      DO 70 I=1,NMCH
         I1=NUMMCH(I)
         DO 80 J=1,I1
            WRITE(19,8030) MCHSTS(I,J,4),MCHSTS(I,J,5),MCHSTS(I,J,7)
   80    CONTINUE
   70 CONTINUE
 7000 FORMAT(2(I1,2X),74X)
 8020 FORMAT(3(I7,2X))
 8030 FORMAT(3(I7,2X))
      RETURN
      END
C
C
```

```
      SUBROUTINE MCHSLC(FLGEND,NOJOB,IFAC,IMCH,ITIM)
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(1),3
     *)
C
      COMMCN /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
       CHARACTER*4 MCHCDS,MCHNUM
       CHARACTER*1 MCHDES,MANAUT
C
C
       INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
C
      INTEGER MCHSTS,JOBSTS
C
      INTEGER WIPTIM,FLGEND,NOJOB,IFAC,IMCH
C
      INTEGER ARRAY(30,3),TEMP
C
C
      INDEX=0
      DO 10 I=1,NMCH
         II=NUMMCH(I)
         DO 20 J=1,II
            INDEX= INDEX+1
            ARRAY(INDEX,1)=MCHSTS(I,J,2)
            ARRAY(INDEX,2)=I
            ARRAY(INDEX,3)=J
 20       CONTINUE
 10    CONTINUE
```

```
C
       DO 30 I=1,INDEX
           DO 40 J=2,INDEX
              JJ=J-1
              IF (ARRAY(J,1).LT.ARRAY(JJ,1)) THEN
                DO 50 K=1,3
                   TEMP= ARRAY(JJ,K)
                   ARRAY(JJ,K)= ARRAY(J,K)
                   ARRAY(J,K)= TEMP
  50            CONTINUE
              ENDIF
  40      CONTINUE
  30  CONTINUE
C
C
      FLGEND=1
      NN=NOJOB
      DO 60 I= 1,INDEX
          IF (ARRAY(I,1) .GE. 0) THEN
              FLGEND= 0
              IF (NOJOB .EQ.0) THEN
                  ITIM=ARRAY(I,1)
                  IFAC=ARRAY(I,2)
                  IMCH=ARRAY(I,3)
                  GO TO 2010
              ELSE
                  NN=NN-1
                  IF (NN .EQ. -1) THEN
                      ITIM= ARRAY(I,1)
                      IFAC= ARRAY(I,2)
                      IMCH= ARRAY(I,3)
                      GO TO 2010
                  ENDIF
              ENDIF
          ENDIF
```

```
   60   CONTINUE
         FLGEND = 1
  2010  RETURN
        END
C
C
        SUBROUTINE WIPSTAT (WIPTIM)
         COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
       *NMCH
C
         COMMON /ONES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
       *)
C
         COMMON /TWOA/ NJOBS,NOPN(75),LOT(75),DUE(75),MATCST(75),OPNUM(75,
       *10),STPTIM(750,10,2),PRSTIM(75,10,2),STPCLS(75,10,2),NOPALT(75,10)
       *,NWKCNT(75,10,2)
C
C
        COMMON /TWOB/ STNST(75)
C
C
        COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),COMPOP(75),Q
       *UELNG(10,3)
C
        COMMON /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
         CHARACTER*4 MCHCDS,MCHNUM
         CHARACTER*1 MCHDES,MANAUT
C
        CHARACTER*1 STNST
        CHARACTER*4 WCNT(3)
C
C
         INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
C
```

```
      INTEGER NJOBS,NOPN,LOT,DUE,OPNUM,STPCLS,NWKCNT,NOPALT
C
       INTEGER OPNALL,JOBOP,COMPCP,MCHOP,QUELNG
      INTEGER MCHSTS,JOBSTS
C
      INTEGER TOTAL(15,2),WIPTIM,TOTWIP
C
      REAL STPTIM,PRSTIM,MATCST
C
C
       DO 5 I = 1,NMCH
           I1 = NUMMCH(I)
           DO 6 J = 1,I1
               IF (MCHSTS(I,J,2) .GT. WIPTIM) GO TO 6
                ISTART = MCHOP(I,J,1)
                IEND   = MCHOP(I,J,2)
                DO 7 K = ISTART,IEND
                    IF (OPNALL(K,7) .EQ. 0) THEN
                        RETURN
                    ENDIF
 7               CONTINUE
 6          CONTINUE
 5      CONTINUE
      DO 10 I= 1, NMCH
         ISTART= MCHOP(I,1,1)
         IEND= MCHOP(I,1,2)
         TOTAL(I,1)=0
         TOTAL(I,2)=0
         DO 20 J= ISTART, IEND
          IF (OPNALL(J,7) .EQ. 0) THEN
              IF (OPNALL(J,9) .GT. WIPTIM) GO TO 20
C         PRINT*,'QUE',J,OPNALL(J,1),OPNALL(J,2),I,OPNALL(J,9)
              TOTAL(I,1)= TOTAL(I,1) + OPNALL(J,8)
              TOTAL(I,2) = TOTAL(I,2) + 1
          ENDIF
```

```
  20      CONTINUE
  10  CONTINUE
C
C
      TOTWIP=0
      RAWMAT = 0
      RLABOR = 0
      NCOMP = 0
C
C
      DO 30 I= 1, NJOBS
          IF (COMPOP(I) .EQ. NOPN(I)) THEN
              NCOMP = NCOMP + 1
              GO TO 30
          ENDIF
          I1 = COMPOP(I) + 1
          I2 = NOPN(I)
          DO 40 J= I1,I2
             TOTWIP = TOTWIP + STPTIM(I,J,1) + PRSTIM(I,J,1)*LOT(I)
             RAWMAT = RAWMAT + JOBOP(I,1)
             RLABOR = RLABOR + JOBOP(I,2)
  40      CONTINUE
  30  CONTINUE
        WRITE(17,8010) WIPTIM,TOTWIP,NCOMP,RAWMAT,RLABOR
      DO 50 I = 1,NMCH
           I1 = NUMMCH(I)
           WRITE(17,8020) TOTAL(I,1),TOTAL(I,2)
 50   CONTINUE
      WIPTIM = WIPTIM + 1440
C
C
 8010 FORMAT(3(I7,2X),2(F9.2,2X),31X)
 8020 FORMAT(I7,3(2X,I3),48X)
 2000 RETURN
      END
```

```
C
C
      SUBRCUTINE UPNSCH(FLGEND,NCJOB,IFAC,IMCH,ITIM,IHRST,ITNSFR,DSEED)
C
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       CONMCN /CNES/ MCHCDS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
     *)
C
       CCMMON /TWOA/ NJOBS,NOPN(75),LOT(75),DUE(75),MATCST(75),CPNUM(75,
     *10),STPTIM(750,10,2),PRSTIM(75,10,2),STPCLS(75,10,2),NOPALT(75,10)
     *,NWKCNT(75,10,2)
C
C
      COMMCN /TWOB/ STNST(75)
C
C
      COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),COMPOP(75),Q
     *UELNG(10,3)
C
      CCMMCN /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
       CHARACTER*4 MCHCDS,MCHNUM
       CHARACTER*1 MCHDES,MANAUT
C
      CHARACTER*1 STNST
      CHARACTEP*4 WCNT(3)
C
C
       INTEGER NUMMCH,MCPCAP,MCHAVL,MCHEFF
C
      INTEGER NJOBS,NOPN,LOT,DUE,OPNUM,STPCLS,NWKCNT,NOPALT
C
       INTEGER OPNALL,JOBOP,COMPOP,MCHOP,QUELNG
```

```
      INTEGER MCHSTS,JOBSTS
C
       INTEGER FLGEND,NOJOB,IFAC,IMCH,ITIM
C
      REAL STPTIM,PRSTIM,MATCST
C
      DOUBLE PRECISION DSEED
C
C
      FLGEND = 1
C     PRINT *,'46010,IF,IM,I',IFAC,IMCH,I
       IF (MCHOP(IFAC,IMCH,1) .EQ. 0) THEN
           MCHSTS(IFAC,IMCH,2) = -10
           NOJOB = NOJOB + 1
           RETURN
       ENDIF
      CALL SELEC (I,FLGEND,IFAC,IMCH,ITIM,IHRST,ITNSFR,DSEED)
C       PRINT *,'46101',FLGEND,IFAC,IMCH,ITIM,OPNALL(I,9),I
C     PRINT*,'46110,IF,IM,I,FLG',IFAC,IMCH,I,FLGEND
      IF (FLGEND .EQ. 0) GO TO 2030
      ISTART = MCHOP(IFAC,IMCH,1)
      IEND   = MCHOP(IFAC,IMCH,2)
      DO 20 I = ISTART,IEND
          IF (OPNALL(I,7) .EQ. 1) THEN
              FLGEND = 0
              NOJOB = NOJOB + 1
              RETURN
          ENDIF
 20   CONTINUE
      MCHSTS(IFAC,IMCH,2) = -10
C      PRINT*,'47710',IFAC,IMCH,ITIM,MCHSTS(IFAC,IMCH,2)
      FLGEND = 0
      RETURN
 2030 FLGEND = 0
      IF (MCHSTS(IFAC,IMCH,2) .LT. OPNALL(I,9)) THEN
          MCHSTS(IFAC,IMCH,2) = OPNALL(I,9)
```

```
C           PRINT*,'48325',IFAC,IMCH,ITIM,MCHSTS(IFAC,IMCH,2)
            RETURN
      ENDIF
      IF (STPCLS(OPNALL(I,1),OPNALL(I,2),OPNALL(I,3)) .EQ. MCHSTS(IFAC,I
     *MCH,1)) THEN
          PRNT = 0.10
       ELSE
          PRNT = 1.0
      ENDIF
      MCHSTS(IFAC,IMCH,1) = STPCLS(OPNALL(I,1),OPNALL(I,2),OPNALL(I,3))
      COMPCP(OPNALL(I,1)) = COMPCP(OPNALL(I,1)) + 1
      OPNALL(I,7) = -1
      JOBSTS(OPNALL(I,1),2) = JOBSTS(OPNALL(I,1),2) + STPTIM(OPNALL(I,1)
     *,OPNALL(I,2),OPNALL(I,3))*PRNT
C     QUELNG(IFAC,IMCH) = QUELNG(IFAC,IMCH) - 1
C     IF (QUELNG(IFAC,IMCH) .LE. 0) QUELNG(IFAC,IMCH) = 0
C     PRINT*,'49201',IFAC,IMCH,QUELNG(IFAC,IMCH)
      NOJOB = 0
C     I1 = NUMMCH(IFAC)
C     DO 25 J = 1,I1
C         QUELNG(IFAC,J) = QUELNG(IFAC,IMCH)
C     PRINT*,'49241',IFAC,IMCH,QUELNG(IFAC,IMCH)
C25   CONTINUE
      MCHSTS(IFAC,IMCH,3) = ITIM
      MCHSTS (IFAC,IMCH,4) = STPTIM(OPNALL(I,1),OPNALL(I,2),OPNALL(I,3))
     **PRNT + MCHSTS(IFAC,IMCH,4)
      MCHSTS(IFAC,IMCH,8) = MCHSTS(IFAC,IMCH,8) - OPNALL(I,8)
      MCHSTS(IFAC,IMCH,9) = OPNUM(OPNALL(I,1),OPNALL(I,2))
      MCHSTS(IFAC,IMCH,6) = OPNALL(I,1)
      ISTTIM = ITIM
      IENTIM = ISTTIM + OPNALL(I,8) -(1-PRNT)*STPTIM(OPNALL(I,1),OPNALL(
     *I,2),OPNALL(I,3))
      ITOTIM = OPNALL(I,8) - (1-PRNT)*STPTIM(OPNALL(I,1),OPNALL(I,2),OPN
     *ALL(I,3))
      MCHSTS(IFAC,IMCH,5) = MCHSTS(IFAC,IMCH,5) + ITOTIM
```

```
      JOBOP(OPNALL(I,1),2) = JOBOP(OPNALL(I,1),2) + ITOTIM * 1.00
      MCHSTS(IFAC,IMCH,7) = MCHSTS(IFAC,IMCH,7) +(1-PRNT)*STPTIM(OPNALL(
     *I,1),OPNALL(I,2),OPNALL(I,3))
      ISHIFT = (ISTTIM/480) - (((ISTTIM/480)/3)*3) + 1
C       PRINT*,'50110',ISHIFT,ITOTIM,MCHAVL(IFAC,IMCH,ISHIFT)
 2040 IF (ITOTIM .GT. 0) THEN
          IF (MCHAVL(IFAC,IMCH,ISHIFT) .EQ. 1) THEN
              I1 = ((ISTTIM/480) + 1)*480
              IF (I1 .GT. IENTIM) THEN
                  GO TO 2060
              ELSE
                  ITOTIM = ITOTIM + I1 - ISTTIM
                  ISTTIM = I1
                  ISHIFT = ISHIFT + 1
                  GO TO 2050
              ENDIF
          ELSE
              I1 = ((ISTTIM/480) + 1)*480
              IENTIM = IENTIM + I1 - ISTTIM
              ISTTIM = I1
              ISHIFT = ISHIFT + 1
          ENDIF
      ENDIF
 2050 IF (ISHIFT .GT. 3) ISHIFT = 1
      GO TO 2040
 2060 MCHSTS(IFAC,IMCH,2) = IENTIM
C       PRINT*,'52210',IFAC,IMCH,ITIM,MCHSTS(IFAC,IMCH,2)
      IF (OPNALL(I,2) .EQ. NOPN(OPNALL(I,1))) JOBSTS(OPNALL(I,1),3) = IE
     *NTIM
      I1 = NOPALT(OPNALL(I,1),OPNALL(I,2))
      DO 30 J = 1,I1
           I2 = NUMMCH(NWKCNT(OPNALL(I,1),OPNALL(I,2),J))
           DO 40 K = 1,I2
                ISTART = MCHOP((NWKCNT(OPNALL(I,1),OPNALL(I,2),J)),K,1)
                IEND   = MCHOP((NWKCNT(OPNALL(I,1),OPNALL(I,2),J)),K,2)
```

```
               DO 50 L = ISTART,IEND
                    IF (OPNALL(L,1) .EQ. OPNALL(I,1)) THEN
                        IF (OPNALL(L,2) .EQ. OPNALL(I,2)) THEN
                            IF (OPNALL(L,3) .EQ. J) THEN
                                IF (OPNALL(I,3) .NE. J) THEN
                                    OPNALL(L,7) = -2
                                ELSE
                                    OPNALL(L,7) = -3
                                ENDIF
                                QUELNG(OPNALL(L,4),K) = QUELNG(OPNALL(L
     *,4),K) - 1
                                IF (QUELNG(OPNALL(L,4),K) .LE. 0) QUELN
     *G(OPNALL(L,4),K) = 0
C      PRINT*,'53810',I1,J,K,I2,L,ISTART,IEND,OPNALL(L,7)
                              MCHSTS(OPNALL(L,4),K,8) = MCHSTS(OPNALL(
     *L,4),K,8) - OPNALL(L,8)
                            ENDIF
                        ENDIF
                    ENDIF
 50             CONTINUE
 40         CONTINUE
 30     CONTINUE
      OPNALL(I,7) = -1
      I1 = NOPN(OPNALL(I,1))
      IF (OPNALL(I,2) .NE. I1) THEN
          I1 = OPNALL(I,2) + 1
          I2 = NOPALT(OPNALL(I,1),I1)
          DO 60 J = 1,I2
               I3 = NUMMCH(NWKCNT(OPNALL(I,1),I1,J))
               DO 70 K = 1,I3
                    ISTART = MCHOP((NWKCNT(OPNALL(I,1),I1,J)),K,1)
                    IEND   = MCHOP((NWKCNT(OPNALL(I,1),I1,J)),K,2)
                    DO 80 L = ISTART,IEND
                        IF (OPNALL(L,1) .EQ. OPNALL(I,1)) THEN
                            IF (OPNALL(L,2) .EQ. I1) THEN
```

```
                                    IF (OPNALL(L,3) .EQ. J) THEN
                                        OPNALL(L,7) = 0
C      PRINT*,'56310',I1,I2,I3,J,K,L,ISTART,IEND,OPNALL(L,7)
                                        OPNALL(L,9) = IENTIM
                                        QUELNG(OPNALL(L,4),K) = QUELNG(OPN
     *ALL(L,4),K) + 1
                                        MCHSTS(OPNALL(L,4),K,8) = MCHSTS(O
     *PNALL(L,4),K,8) + OPNALL(L,8)
C      PRINT*,'56310',OPNALL(L,4),OPNALL(L,5),QUELNG(OPNALL(L,4),OPNALL(L
C    *,5)),L
                                    ENDIF
                                ENDIF
                            ENDIF
 80                   CONTINUE
 70              CONTINUE
 60         CONTINUE
       ENDIF
         WRITE (18,8010) OPNALL(I,1),OPNALL(I,2),OPNALL(I,3),OPNALL(I,4)
     *,OPNALL(I,5),OPNALL(I,9),MCHSTS(IFAC,IMCH,3),MCHSTS(IFAC,IMCH,2)
C       PRINT *,'57302, IFAC,IMCH,I,',IFAC,IMCH,I,OPNALL(I,1)
      IEND = MCHOP(IFAC,IMCH,2)
       ISTART = MCHOP(IFAC,IMCH,1)
 8010 FORMAT (5(I3,2X),3(I7,2X))
      RETURN
      END                                                    /
       SUBROUTINE SELEC (I,FLGEND,IFAC,IMCH,ITIM,IHRST,ITNSFR,DSEED)
       COMMON /ONE/ MCHCAP(10,3),MCHEFF(10,3),MCHAVL(10,3,3),NUMMCH(10),
     *NMCH
C
       COMMON /ONES/ MCHCOS(10),MCHDES(10,3,35),MCHNUM(10,3),MANAUT(10,3
     *)
C
       COMMON /TWOA/ NJOBS,NOPN(75),LOT(75),DUE(75),MATCST(75),OPNUM(75,
     *10), STPTIM(750,10,2),PRSTIM(75,10,2),STPCLS(75,10,2),NCPALT(75,10)
     *,NWKCNT(75,10,2)
```

```
C
C
      COMMON /TWOB/ STNST(75)
C
C
      COMMON /FIV/ OPNALL(600,11),JOBOP(75,2),MCHOP(10,3,2),COMPOP(75),Q
     *UELNG(10,3)
C
      COMMON /SIX/ MCHSTS(10,3,9),JOBSTS(75,3)
C
       CHARACTER*4 MCHCDS,MCHNUM
       CHARACTER*1 MCHDES,MANAUT
C
      CHARACTER*1  STNST
      CHARACTER*4  WCNT(3)
C
C
       INTEGER NUMMCH,MCHCAP,MCHAVL,MCHEFF
C
      INTEGER NJOBS,NOPN,LCT,DUE,OPNUM,STPCLS,NWKCNT,NOPALT
C
       INTEGER OPNALL,JOBOP,COMPOP,MCHOP,QUELNG
      INTEGER MCHSTS,JOBSTS
C
      INTEGER TEMP(50),FLGEND
C
      REAL STPTIM,PRSTIM,MATCST
C
      DOUBLE PRECISION DSEED
C
C
      FLGEND = 1
      ISTART = MCHOP(IFAC,IMCH,1)
      IEND = MCHOP(IFAC,IMCH,2)
      GO TO (2000,2100,2200),IHRST
```

```
2000 I1 = ITIM
      I2 = 2147483647
     IFLG = 1
     DO 10 J = ISTART,IEND
         IF (OPNALL(J,7) .EQ. 0) THEN
             IF (OPNALL(J,9) .LE. I2) THEN
                 I2 = OPNALL(J,9)
                IFLG = 0
                I = J
                FLGEND = 0
             ENDIF
             IF (OPNALL(J,9) .LE. I1) THEN
                FLGEND = 0
                I = J
                RETURN
             ENDIF
         ENDIF
 10   CONTINUE
      RETURN
 2100 I1 = ITIM
      I2 = 2147483647
       IFLG = 1
      DO 20 J = ISTART,IEND
          IF (OPNALL(J,7) .EQ. 0) THEN
             IF (IFLG .NE. 0) THEN
                IF (OPNALL(J,9) .LT. I2) THEN
                   I2 = OPNALL(J,9)
                   FLGEND = 0
                   I = J
                ENDIF
             ENDIF
             IF (OPNALL(J,9) .LE. I1) THEN
                FLGEND = 0
                I = J
                I1 = OPNALL(J,9)
                IFLG = 0
```

```
               ENDIF
           ENDIF
 20     CONTINUE
        RETURN
 2200   I1 = ITIM
        I2 = 0
        I3 = 2147483647
        IFLG = 1
        DO 30 J = ISTART,IEND
             IF (OPNALL(J,7) .EQ. 0) THEN
C       PRINT *,IFAC,IMCH,ITIM,I2,J
                 IF (IFLG .NE. 0) THEN
                     IF (OPNALL(J,9) .LT. I3) THEN
                         I3 = OPNALL(J,9)
                         FLGEND = 0
                         I = J
                     ENDIF
                ENDIF
C              PRINT *,'J,J9,I1,ITIM',J,OPNALL(J,9),I1,ITIM
                IF (OPNALL(J,9) .LE. I1) THEN
                     FLGEND = 0
                    I2 = I2 + 1
                    TEMP(I2) = J
                    IFLG = 0
                ENDIF
            ENDIF
 30     CONTINUE
        IF (FLGEND .EQ. 1) THEN
            RETURN
        ELSE
            IF (IFLG .EQ. 1) RETURN
            CALL RANDCM(YFL,DSEED)
```

```
               V1 = 1.0/(I2*1.0)
            V2 = 0.0
            DO 40 J = 1, I2
                V3 = J*V1
                IF ((YFL .GE. V2) .AND. (YFL .LT. V3)) GO TO 2300
                V2 = V3
 40         CONTINUE
        ENDIF
 2300   I = TEMP(J)
C       PRINT*,'I,YFL,I2,V2,V3TEMJ',I,YFL,I2,V2,V3
        FLGEND = 0
        RETURN
        END
C
C
C
        SUBROUTINE RANDOM(YFL,DSEED)
        DOUBLE PRECISION DSEED
        DSEED = (7**5)*DSEED
        X = 2147483647
        DSEED = MOD(DSEED,DBLE(X))
        YFL = DSEED/X
        RETURN
        END
```