Order Number 9020786

A comparative evaluation of some group-technology scheduling heuristics

Metwally, Amal Fathy, Ph.D.

City University of New York, 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A COMPARATIVE EVALUATION

OF SOME

GROUP-TECHNOLOGY SCHEDULING HEURISTICS

BY

AMAL FATHY METWALLY

A dissertation submitted to the Graduate Faculty in Business in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1990

This manuscript has been read and accepted for the Graduate Faculty in Business in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/15/90
Date

Chair of Examining Committee

1-22-90
Date

Executive Officer

Dr. George Schneller

Dr. Georghios Sphicas

Dr. Emre Veral

Supervisory Committee

The City University of New York

ABSTRACT

A COMPARATIVE EVALUATION OF SOME GROUP-TECHNOLOGY SCHEDULING HEURISTICS

by

Amal Fathy Metwally

Advisor : Professor David G. Dannenbring

Group technology is an innovative approach to increase efficiency and productivity of small-, and medium-batch production by grouping similar parts together in families, and arranging the machines required to process these families into cells.

By doing this, GT can achieve economic advantages similar to those associated with continuous flow-line production. GT implementation can reduce setup times, simplify materials flow, increase capacity, and improve production planning and control. But the great potential of GT is the role it can play in integrating modern manufacturing- and information-related technologies and systems, such as integrating Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) in Computer-Integrated Manufacturing (CIM). GT also provides the cellular manufacturing concept which is applied in Flexible Manufacturing Systems.

Because of this important role that GT can play, and for its expected advantages, this study concentrates on one of the major areas of GT application, which is production scheduling. The scheduling problem is greatly simplified, and its scope is also reduced by using GT.

The study investigates and compares the performance of five different scheduling heuristics in a GT flow line to minimize the maximum flow time (makespan). Three of these heuristics were originally developed for use in the conventional flow shop, and they have been adjusted in this study for use in a static, deterministic group-technology flow line. The other two heuristics are new versions of a conventional flow shop scheduling heuristic.

Besides evaluating performance of these five heuristics, the effects of some operational factors are also tested, including the impact of problem size in general, and in particular, the impact of the number of families, the number of jobs within each family, and the number of machines on performance of the heuristics. The impact of the range of group setup time on the relative performance of heuristics is also tested.

Computational results for a variety of small and large problems are presented. Conclusions and directions for future work are also reported.

ACKNOWLEDGEMENT

[Thanks be to Allah for finishing this dissertation.]

This dissertation could not have been completed without the help, direction, support, and encouragement of many people.

First, I would like to express my deep appreciation to my committee chairman and advisor, Professor David Dannenbring for his guidance, encouragement, and interest in this research.

I am delighted to thank the other members of my committee, Professor Georghios Sphicas, Professor George Schneller IV, and Professor Emre Veral for their time, effort, and thoughts. I am sincerely grateful to Professor G. Sphicas, who has been a great source of support and encouragement throughout my doctoral studies. Special thanks is given to Professor Schneller for his encouragement and support during my research assistantship with him and after that.

I am also sincerely grateful to Professor Donald Vredenburgh the executive officer of the Ph.D. program in Business for his support. I owe many thanks to Prof. Mona F. Metwally (my sister), and Prof. Hassan of Ein Shams University (Egypt) for their assistance.

Many thanks to the staff and consultants of the EEC

Micro Computer Lab at Baruch College. I also wish to thank Jaquelyn Stasiak, Rona Ostrow, Joe Echelle and all the staff working at the Graduate Business Resource Center at Baruch for their help.

Special thanks to my friends, Azza and Momen Elmasry, Sherifa and Mostafa Mohammad, Samia Madkour, Nadia and Bassam Elahmady, Nadia, her mother, and Maher Elrayes for their warm friendship, support and incredible cheerfulness during my frustrating moments in my studies.

Infinite thanks to my parents, who have always given me love, support, and inspiration for which I am truly indebeted to them. Many thanks to my uncles, aunts, sister, brothers, cousins, mother-in-law and their families.

Finally, and the most important, many thanks to my husband Ahmed Fouad Salem, and my daughter Yasmin, who have suffered with me, and have sacrificed for me more than any body else. Very special thanks to my husband Ahmed for his advice, comments, encouragement, and real help. Many thanks for allowing me to devote the time required to finish this work. His sacrifices to finish my research first before his own, is greatly appreciated.

To my family Ahmed and Yasmin, and to my grand family "EGYPT", I lovingly dedicate this dissertation with heartfelt thanks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES IN THE APPENDICES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

In this chapter, we present the scope of the study, its significance, and an overview of the dissertation.

1.1 Scope and Limits of the Study:

This study is a computer-based investigation and comparison of different heuristics to solve the scheduling problem for a static deterministic group technology flow line, using the total flow time (makespan) as an objective function.

The study compares the performance of the following five heuristics:

1 - SOI (Slope Order Index) heuristic developed by Palmer, 1965. (104)

2 - NEH, a heuristic developed by Nawaz, Enscore, and Ham, 1983. (97)

3 - CDS, a heuristic developed by Campbell, Dudek,and Smith, 1970. (20)

4 - MOD1, the first modified heuristic, and

5 - MOD2, the second modified heuristic.

The last two heuristics have not been previously reported, and are newly modified versions of NEH. The first three heuristics were chosen from the literature as a basis of comparison for the new heuristics. They were chosen because they proved in previous comparative studies to perform better in general than the others (as will be shown in the literature review).

The study also investigates the impact of the problem size (represented by number of families, number of jobs within each family , and number of machines), besides the impact of group setup time on the performance of these heuristics.

The heuristics are tested over a large number of problems of different sizes. The computational results are analyzed using different statistical tests.

1.2 Significance of the Study:

The importance of the study can be summarized by a few major points, each of which clarifies the importance of choosing a specific factor or variable to be included in the study. These points are listed in the next subsection. (*)

1.2.1 Significance of Group Technology:

Group Technology is a philosophy and a technique to increase production efficiency in small- and medium-batch manufacturing (job shops).

(*) The references of the following subsection are mentioned in the related chapters.

Within the next few years, a higher percentage of the industrial production all over the world is expected to be on a small-lot basis.

The need to increase the productivity of job shops (batch production) in the dynamic, competitive environment increases the focus on group technology (GT).

GT is one of the main approaches to reform this section of industry by bringing some advantages of the flow-shop production into the small-batch production.

GT achieves many benefits in the short run, such as reducing setup time, simplifying material flow, and improving production planning and control. But the most important is the long-term potential through applying the modern concepts and technology, such as CIM (Computer-Integrated Manufacturing), and FMS (Flexible Manufacturing Systems), and in general the automated factory of the future.

1.2.2 Significance of the Flow Line as a GT Layout:

Out of the three forms of arrangement or layout of group technology (GT center, GT cell, and GT flow line), the flow line is considered the most rational. The GT flow line can bring all the possible advantages of the flow shop into small-batch production.

Viewing a GT cell as a flow shop reduces most of the job-shop production-planning and control problems. Thus most of the mathematical models available for planning the flow shop can be applied to the GT cell.

1.2.3 Significance of the Scheduling in GT Application:

The scheduling decision in general has a great effect on three principal types of costs: the inventory cost, the facility utilization cost, and the lateness cost. These costs imply a wider effect on the performance of the company as a whole.

One of the major areas for GT applications is production scheduling, because the scope of the scheduling problem is greatly simplified by applying GT, as will be discussed in subsection (3.3.1). Based on this simplification, the available mathematical models that have been developed for the conventional flow shop can be applied (with some modifications) to the group-scheduling problems.

Also when the advantages of both (better scheduling decisions and successful GT implementations) are combined, this will result in a significant cost reduction.

When the scheduling in GT is performed accurately, this will lower the need to maintain large quantities of parts inventory. Instead of that, each part can be manufactured just before it is needed, which can lead to applying the just-in-time concept, as will be discussed later in subsection (3.3.1).

1.2.4 Significance of Using Heuristics in Scheduling:

Most of the flow-shop scheduling problems including three machines or more are said to be NP-complete, i.e. no simple rule has been offered for determining the optimal schedule.

Although the general-purpose methodologies for solving combinatorial problems such as dynamic programming, and branch and bound can be used in solving this problem, in most of the cases, the computation time, and the memory required to keep track of the calculation is prohibitive, even for small problems.

Though heuristics do not necessarily provide the optimal solution to the problem, they are, for the most part, an efficient and economical way of getting a good solution to the problem.

1.3 Overview of the Dissertation:

The dissertation includes six chapters that cover the conceptual bases of the study, the evaluation procedures, the results of the evaluation, and the conclusions.

In chapter I, the scope of the study, and its significance are presented. In the significance of the study, we consider the importance of each factor or variable chosen to be included in the study. Four basic points are described including the importance of GT concept in general, and in particular, its flow-line form, also the integral role played by scheduling in GT applications, and the reasons for using the heuristic approach to solve the scheduling problem.

The group technology definition, its historical evolution, basic layout forms, and the expected benefits of implementing GT, are discussed in chapter II. Then, the chapter is concluded with the potential of GT.

Chapter III is a conceptual presentation for the scheduling problem. The three sections of this chapter cover the general scheduling problem, the flow-shop scheduling problem, and the different scheduling techniques available for GT application.

In the general-scheduling problem section, the importance of the scheduling decision, the description of the problem, and the different performance measures of scheduling are discussed. Then, three different approaches to solve the flow shop problem are presented including the mathematical, theoretical, and heuristic approaches. Thirteen different heuristic procedures for flow-shop scheduling are discussed. Chapter three also includes the literature review for the GT scheduling area, including the GT center, the GT cell, and the GT flow line.

Chapter IV is devoted to the evaluation procedures. We begin with the mathematical definition of the problem, its assumptions, and procedures of the five evaluated heuristics, followed by the eight hypotheses of the study, and their testing procedures. Then we conclude with the measures or ratios used in evaluating the heuristics performance, and the bases of the parametric and nonparametric statistical tests used in the analysis.

Chapter V contains the results of the evaluation grouped according to the hypotheses or research questions. The results of using different statistical tests for each hypothesis are presented with the results for other evaluation measures (not used in the statistical analysis).

The most important result of the study is that there is no significant difference, statistically, between performance of the two heuristics first reported here (MOD1, and MOD2), and performance of the best heuristic, chosen from the literature.

The summary of this study, the conclusions, and suggestions for future research and developments are offered in chapter VI.

CHAPTER TWO

THE GROUP TECHNOLOGY CONCEPT

We consider in chapter two all the issues related to the GT concept. It covers GT's definition, its evolution, differences between functional and GT layouts, and different layout forms of GT. The chapter is concluded with the expected benefits of GT, including its potential in the factory of the future.

2.1 Introduction:

Interest in Group Technology (GT) has recently increased in the field of batch production. The need for increased productivity in small- and medium-batch manufacturing has brought focus to the concept of Group Technology.

The importance of batch-manufacturing productivity stems from the fact that, within the next few years, a higher percentage of industrial production in the U.S.A. and throughout the world is expected to be on a small-lot basis. Groover (48) states that the estimates for manufactured parts in lots of fifty or fewer will constitute 75% of the overall manufactured parts in the years to come. This necessitates the development of systematic procedures which will simplify and rationalize the production planning and control activities in an integrated way in batch manufacture.

Also the trend of increasing labor wages, accompanied with a limited supply of skilled workers, as well as increased costs of energy, and other important factors such as competition, necessitate better and more efficient manufacturing systems. (118)

As Ashton and Cook (9) stated :

> "American industry is beginning to put its manufacturing house in order. Companies building high-volume standardized products are learning from their rivals and emulating their commitment to ongoing product and process innovation.
>
> But one important segment of U.S. industry (job-shop manufacturers) continue to postpone needed reforms. Many of these companies have been sheltered from foreign competition by their products' custom nature and lack of mass market potential.
>
> Without an urgent competitive threat, job-shop managers have been slow to overhaul their manufacturing operations despite problems of mediocre quality , excessive lead times, unreliable delivery, and high costs."

GT implementation is considered one of the most important approaches to reform job-shop manufacturing. The concept stems from the need to improve conditions in the small- and medium-batch situation. The problem to be resolved can be described in one way by the relationship which exists between finished products and the parts from which they are made, as shown in fig.(1). GT is related to the modular production concept. This means specializing in components of more than one product (part standardization). So, in general the GT concept is applicable mainly to parts production, and not to final products production. (85, 127)

The idea of GT started with the aim of reducing setup time while sequencing parts on a single machine, and it became a whole intgrated philosophy for production, called (GT), or cellular manufacturing system.

GT offers great potential for the traditional job shop. Heretofore, these types of gains have usually been attainable through a movement towards production in large batches. The short-term benefits will be discussed in detail, but the most important long-term benefit is the introduction of new technology such as CAD (Computer-Aided Design), CAM (Computer-Aided Manufacturing), and FMS (Flexible Manufacturing Systems), etc. that help to implement a manufacturing strategy aimed at greater automation. (139)

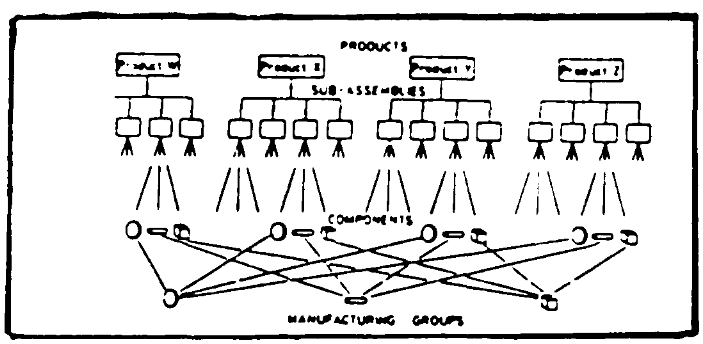


Fig.(1) The relationship between products, components and grouping for manufacture. (42)

The introduction of GT techniques has allowed firms like GE, Lockheed, and Boeing to deal with the enormous problems of classifying and designing hundreds of thousand of parts. Today GT is viewed as an essential step in the move towards factory automation. (49)

2.2 Group Technology Definition:

A wide variety of names has been used to express the concept of group technology such as: cellular manufacturing (CM), machine cells (MC), Part(s) family(s), Group production, group technology (GT), Family-parts-line production, and many others.

GT is an innovative integrated approach or philosophy to manufacturing, which seeks to rationalize medium- and small-batch production by capitalizing on similarirties which exist among component parts. It groups parts or components having similar manufacturing sequences into families. The machines required to process a set of related families are then arranged together in a cell to process them. (48, 72) Fig.(2) shows the difference between functional and cell layouts.

T: Turning

M: Milling

D: Drilling

G: Grinding

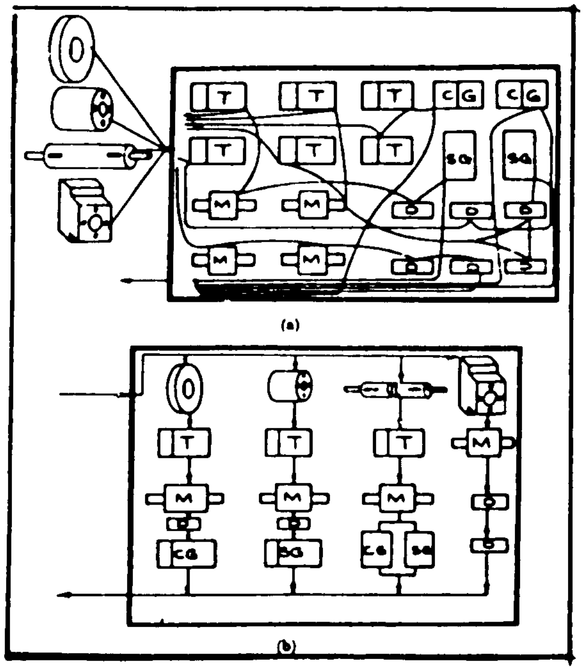


Fig.(2) a- Functional and b- cell layouts (5)

2.3 <u>The Historical Evolution</u>:

Group technology's origin goes back to the U.S.S.R., when Sokolovski mentioned in 1937, that similar parts should be manufactured in the same way, with standardized technological processes. (68) But it is known that Mitrofanov (U.S.S.R.) was the first one to write about GT, although his work was based on Sokolovski's idea. In 1958, Mitrofanov published the first book about GT (Scientific Principles Of Group Technology). His book was translated later in 1966 into English. (93)

Throughout the years 1957-1965, interest in GT peaked in the U.S.S.R., in both studies and applications of the concept. (46) Also during the sixties, some other Eastern European countries showed an interest in studying and applying GT, those countries being Czechoslovacia, East Germany, Rumania, Hungary, and Poland. In the early sixties, West Germany, The Netherlands, Norway, Sweden, France, and Switzerland started their efforts. (69)

In the mid sixties, the U.K. showed great interest in the concept. The British Government sponsored many research centers, many other academic and research institutes were running seminars and offering different courses about GT. A tremendous number of papers were published in the U.K.. By that time the U.K. had more applications of GT than any other country in the world. But by the end of 1970's, the situation in the U.K. had changed, as it was described by Burbidge (the British scholar): "It was not GT that failed, it has not disappeared, it has only emigrated". (16)

In contrast to the European countries, little interest was shown by the U.S.A. researchers and companies to study or implement the concept. But we can say that the U.S.A. showed some early interest in the classification and coding, but the interest in GT only emerged later.

Literature on the applications of GT in the U.S.A. had been scarce with only few summaries (91, 131) until 1977 when reports about GT applications in some American companies were published (57, 70). After that the results of many other applications were published. (67, 73, 81, 90, 115, 116, 125, 136).

Most of the industrial countries, including Germany, Japan, and The U.S.A. have now set GT into proper context both as a main stream contributor to improve productivity, and as natural partner alongside the applications of robotics, Computer-Aided Design (CAD), and Computer-Aided Manufacturing (CAM). (42, 141, 142, 143)

2.4 Part Family Formation:

Finding families of parts is one of the first steps in applying the GT concept. Grouping parts into families is considered also the biggest obstacle in changing over from a traditional-production shop to a group-production system. The part family is a collection of parts that are similar in either geometric or manufacturing attributes. (48) Grouping parts into families can be done through :

1- Visual selection (eyeballing):

This is the easiest and the least expensive, but also the least accurate or comprehensive way of forming groups or families of parts. It should be carried out by skilled manufacturing engineers acquainted with parts and processes. (12, 48, 68)

2- Classification and coding system:

This method is based on classifying parts by their features, and coding these features to facilitate grouping the parts having the same code number. Parts can be classified according to the shape, function, manufacturing operations, tooling, or materials used (12, 17, 48)

3- Production flow analysis (PFA):

This is an alternate method developed by J.L.Burbidge for identifying parts families and their associated groupings of machine tools. (13, 14, 15, 126) This method uses the route sheets of the parts. Parts having similar operations sequence and machine routings are grouped together to form a family, to be processed on a group of machines forming a cell. (17, 41, 47, 75)

2.5 Basic Layout Forms of GT:

Group layout can be classified into three different forms (4, 5) as shown in fig.(3).

1. GT center (single machine):

This is the first degree of rationalization of GT manufacturing system, and it is similar to the functional layout. It is used for parts that require processing on one machine only, such as milling, turning...etc. (5, 47, 86) Some of the authors do not consider this form as a GT layout, but they consider it as process standardization in the functional layout. (68)

2. GT flow line:

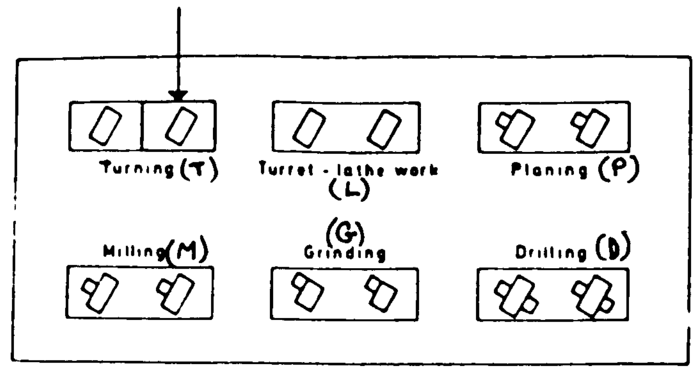
This form can be used when each member of the part families to be processed has almost the same processing route, resulting in the same production flow. This is the highest level of rationalization of GT manufacturing system, and it is close to the mass production flow line. Sometimes it is called "batch production flow line". (86) This form requires a higher level of similarity by having the same sequence, and a uni-directional flow.

3. GT cell:

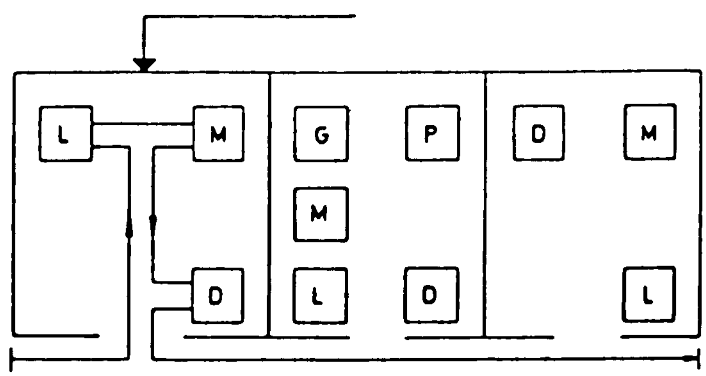
In this case the production flow for each part in one family or more is not identical, therefore a GT flow line is not established. This is the second degree of rationalization, where machines are grouped in cells, each of which processes a family or group of families. This form allows flexible operation sequences depending on the type of parts. (5, 31, 48, 109)

2.6 Advantages of Applying GT:

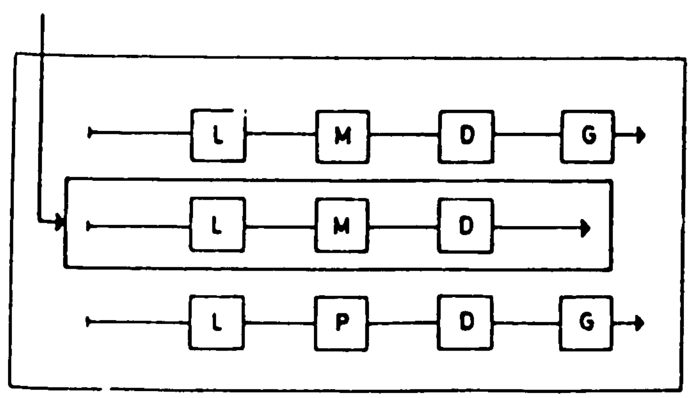
The advantages of implementing the GT concept are mainly reported in three categories: operational, economic, and social benefits. (6, 17 25, 35, 123, 140) The publications have covered these advantages in different ways, either literaly as expected benefits of GT implementation, or as actual results of specific case studies, or as results of surveys handling different applications. Fig.(4) shows some of these advantages.



GT-Centre



GT-Cell



GT-Flow Line

Fig.(3) Different GT Layout Forms (5)

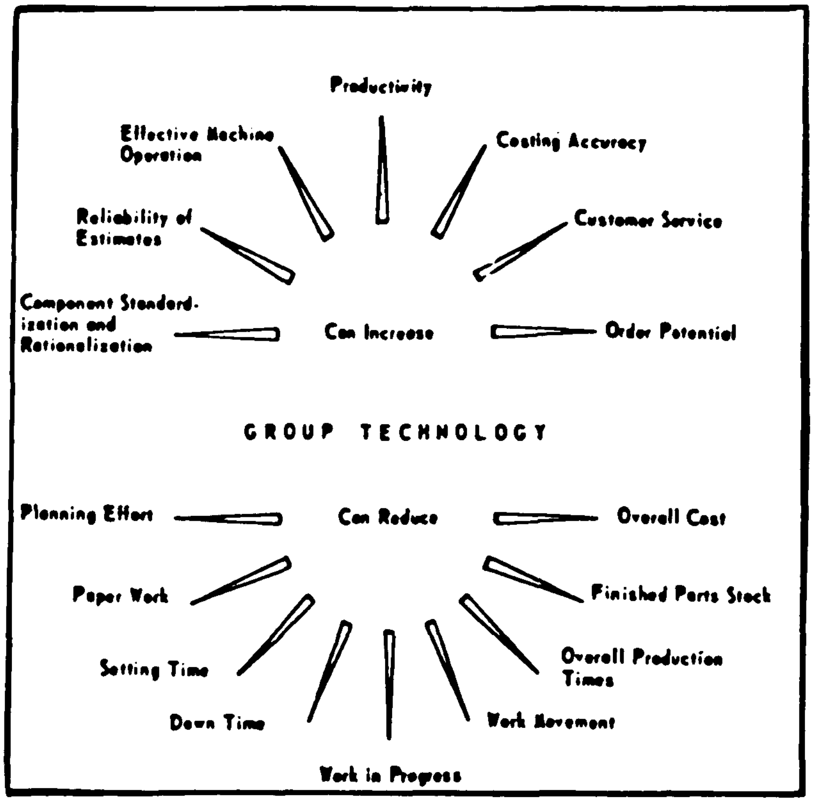


Fig. (4) General Achievements of Group Technology (5)

The following are some of the reported benefits of implementing GT:

1- Increased capacity due to reduced setup time.

2- Simplification of material flow, where all or most of the processes will be performed in one cell which improves the production control, materials handling, throughput time, and work in progress.

3- With GT, different responsibilities of the central planning and control departments are assigned to the supervisor of the cell and the team of workers he is supervising. They become responsible for completing the required parts by their due dates, and at the required level of quality. They also become responsible for scheduling their work.

4- Improving the marketing performance due to the flexibility to adapt to the market changes, and due to improvements in the quality and delivery of the products.

5- Introducing a new design or machine becomes easier, since the change is related to a specific family or cell and not to the whole shop.

6- Reduced data processing and paper work.

7- Other social and behavioral benefits regarding improving the work environment, which increases the output per employee. (18, 26, 47, 48, 88, 89, 99)

A good example of the operational improvement that can be achieved by GT, has been described by Schaffer (115) of a project at Sealol, in Providence, Rhode Island, U.S.A.. The MICLASS coding and analysis system was used to identify 324 bar-type parts, which were using 22 machine tools, and needed 115 routes. The functional flows shown in fig.(5), were rationalized into the single cell using seven machine tools, and the 70 routes shown in fig.(6). The average batch output produced per day increased from 2.2 to 3.85, and the production time was almost halved. (42)

2.7 Potential of GT:

Manufacturing- and information-related technologies in recent years have created numerous subsystems in the design, planning and scheduling of production, and in materials handling. The company seeking the factory of the future must integrate all these areas through CIM (Computer-Integrated Manufacturing). (39, 82, 83, 120) CIM includes CAD (Computer-Aided Design), CAM (Computer-Aided Manufacturing), robots, CAPP (Computer-Aided Process Planning), MRP-II (Manufacturing Resources Planning), and GT. (50, 92)

GT is the practical approach to implement CIM, it integrates both design and manufacturing information (CAD-CAM) as the main componenets of CIM. (59, 80)

GT is also a basic concept in developing Flexible Manufacturing Systems (FMS), that are built on the cellular production concept. (21, 61, 114, 135)

Applying GT for part design and manufacturing and for process planning requires using a good classification and coding system that can identify families of similar parts sharing certain geometrical attributes (e.g., size, shape, and feature configuration) or manufacturing attributes (operations required and their sequence). (51, 70)

In general, the classification and coding system used with GT can provide the following benefits: (60).

1- Effective design and manufacturing information retrieval.

2- Effective grouping for parts families.

3- Standardization and simplification (for both design and manufacturing).

4- Elimination of design and route proliferation.

5- Design cost reduction for more economic manufacturing.

6- And the most important point is providing a data base for integrating CAD/CAM in CIM. (43, 66, 71)

With GT data base, we can store, retrieve, analyze, and standardize design and manufacturing data in the corporate CIM data base. (112)

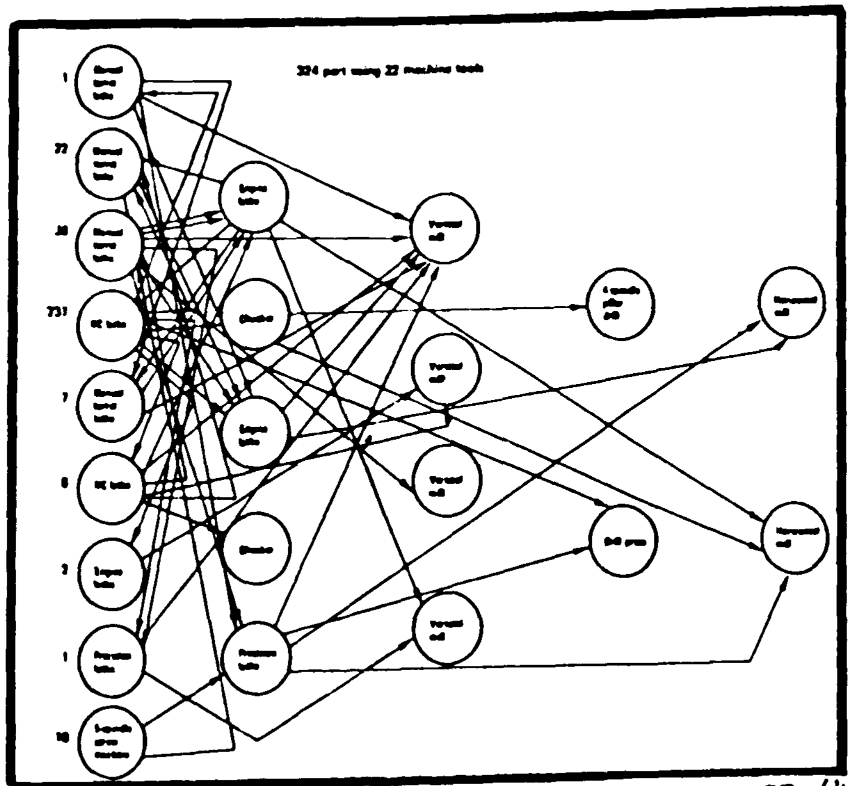


Fig.(5) Functional Flows at Sealol before GT (42)

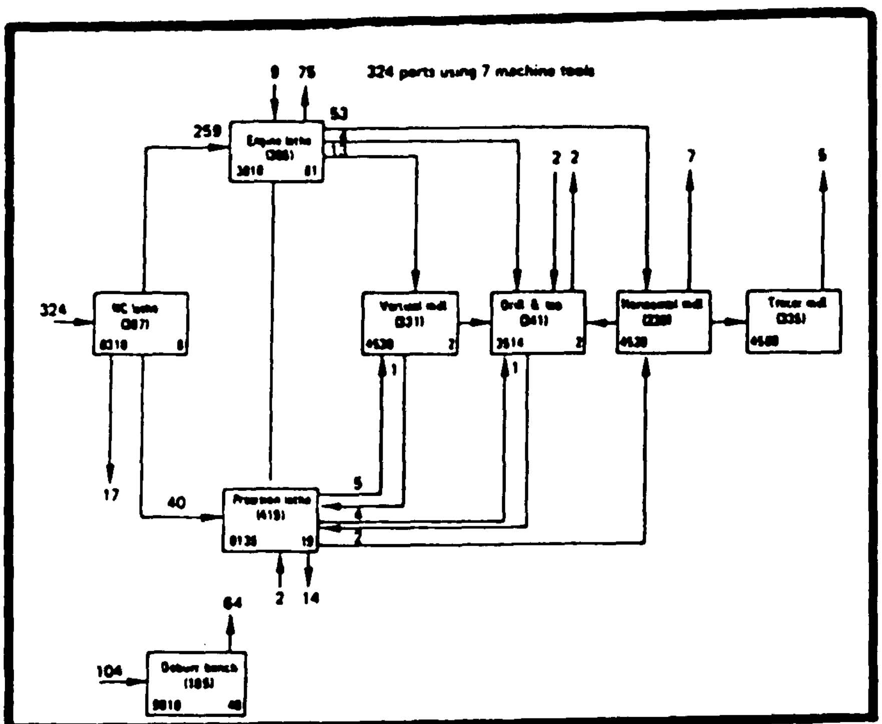


Fig.(6) Flows at Sealol in the GT Cell (42)

CHAPTER THREE

THE SCHEDULING PROBLRM

In chapter three, we discuss the scheduling problem in general (its importance, description, and performance measures), and the flow-shop scheduling problem in particular (its aefinition, solution approaches, and a brief review for the mathematical and the heuristic approaches). The chapter ends with a literature review for the scheduling techniques in different GT layout forms (GT center, cell, and flow line).

3.1 The General Scheduling Problem:

The sequencing problem involves defining an order or priority for a set of jobs or tasks as they proceed from one machine (processor) to another. Thus it determines the position of job (j) relative to all other jobs. These types of problems occur in many different environments.

"A problem could involve jobs in a manufacturing plant, aircraft waiting for landing clearances, or programs to be executed at a computer center".(97)

3.1.1 Importance of Scheduling Decisions :

Conway et.al.(24) grouped the costs that can be affected by the decisions of pure sequence into three principal types. These are the costs of inventory, utilization , and lateness.

1- Much has been written on the costs of inventory. But in general there is an interest in reducing the average inventory. Being able to develop a sequence or a schedule that minimizes the average flow time, we can by this reduce the average inventory. Minimizing the average flow time will also provide a competitive sales advantage.

2- Facility utilization is a very important economic consequence of sequencing decisions. The ability to compact the busy intervals and produce a low mean-flow-time schedule simply implies a procedure that will permit a given facility to do more work. Conversely an efficient scheduling procedure will permit a given work load to be accomplished with a smaller aggregate damand on facilities. In the short run, this is reflected in the costs of overtime, additional shift operation, and overload subcontracts. In the long run it is reflected in the required capacity or the amount of business that can be accommodated.

3- In some situations, especially construction projects, the cost of lateness is obvious and explicit. In manufacturing situations, lateness cost may be obvious in the production-to-order circumstances. It has its long term effect through the customers' displeasure which has implications for future business.

3.1.2 Description of the Scheduling Problem:

Conway et.al.(24) used a four-parameter notation to classify individual scheduling problem, written as A/B/C/D.

A: describes the jobs arrival process.

- For dynamic problems,"A" will identify the probability distribution of the times between arrivals.
- For static problems, it will specify the number of jobs, assumed to arrive simultaneously unless stated otherwise.
- When "n" is given as the first term, it denotes an arbitrary, but finite number of jobs in a static problem.

B: describes the number of machines in the shop.

C: describes the flow pattern in the shop. The principal symbols are:

(F): for the flow shop, where the machines order for all jobs is the same.

(G): for general job shop, where there is no restriction on the form of the technological constraint (machines order).

(-): in the case of a shop with a single stage, there is no flow pattern, and the third parameter of the description is omitted.

D: describes the criterion by which the schedule is to be evaluated, as will be discussed in detail next.

3.1.3 Scheduling Performance Measures:

The purpose of flow-shop scheduling techniques is to determine the schedule that minimizes (or in general optimizes) some well-defined objective function. The following is a presentation of the objective functions or the performance measures that are most commonly optimized in solving the scheduling problem.

Conway et.al.(24) consider all these criteria as a function of one variable (W). The sequencing decision is considered simply to determine when each job should be done, or when each operation of each job should be done. This way of looking at the matter is equivalent to determining how long each operation of each job should wait before processing begins, which is denoted by "W". "W" is the time that the job must wait after the completion of the previous operation before beginning the current operation. So the total waiting time for a job is the sum of the waiting times for all operations of that job.

A schedule is completely specified by giving a set of Ws. All other variables used are functions of the W. In the final analysis every comparison of schedules is based on the comparison of different sets of Ws. The goodness of a schedule is completely a consequence of the values of the W.

The most important measures that may be derived from the W are:

C_j: the completion time of job (j). The time at which processing of the last operation of the job is completed.

F_j: the flow time of job (j). The total time that the job spends in the shop. It equals the sum of the processing times and the waiting times for the job.

L_j: the lateness of job (j). It equals the difference between the due date and the completion date of the job.

Two schedules for a particular problem are identical if and only if they have identical sets of Ws.

In almost all the theoretical work on scheduling, very simple measures of performance have been employed. These have been the average or the maximum of the values of completion time, lateness, or tardiness. All these measures can be expressed as a function of the jobs completion times.

$$\text{Measure (M)} = f\ (C_1\ ,\ C_2, \ldots\ldots\ ,\ C_J)$$

A measure increases only if at least one of the completion times increases.

Hax and Candea (62) divided the performance measures into two groups :

1- Performance measures relating to the individual job, such as flow time, lateness, and earliness of a job. The schedule which minimizes one of these measures is optimum with respect to it.

2- performance measures relating to the shop, such as the measures reflecting the work-in-process inventory, or the average number of jobs in the systems, etc.

Another classification divides these measures into another two groups :

1- criteria aimed at improving customer service, such as average flow time, average waiting time, or percentage of late jobs.

2- criteria aimed at improving resource utilization including labor, machines, or in-process inventory.

French (40) classified these criteria into :

1- Criteria based upon completion times, such as :

F_{max}: Minimizing total flow time. Maximum flow time is essentially saying that a schedule's cost is directly related to its longest job.

C_{max}: Minimizing maximum completion time.
Note: when all ready times are zero, C_{max}, and F_{max} are identical, and can be called the total production time or makespan.

$\overline{F}$: Minimizing mean flow time implies that a schedule's cost is directly related to the average time it takes to process a single job.

$\overline{C}$: Minimizing mean completion time. It is equivalent to minimizing $\overline{F}$, i.e., a schedule which attains the minimum $\overline{C}$ also attains the minimum $\overline{F}$, and vice versa.

2- Criteria based upon due dates:

Since the cost of a schedule is usually related to how much we miss target dates by, so obvious measures of performance are:

L_{max}: maximum lateness. (Lateness is defined as the difference between completion time and due date). It can take either a positive value (tardiness), or a negative value (earliness).

$\bar{L}$: mean lateness.

T_{max}: maximum tardiness. (Tardiness is the positive value of lateness), $T_i = \max \{ L_i , 0 \}$

$\bar{T}$: mean tardiness.

Minimizing $\bar{L}$, or L_{max} is appropriate when there is a positive reward for completing a job early, and the reward is larger the earlier the job is. Minimizing $\bar{T}$, or T_{max} is appropriate when early jobs bring no reward ,only penalties for late jobs.

Makespan is considered the most common criterion. Gupta (53, 54) clarified that the purpose of the flow-shop scheduling technique is to determine the schedule that minimizes or maximizes some well-defined objective function. The most common measure of a schedule performance is the makespan, defined as the total production time or the total flow time in which all jobs complete processing on all machines. While makespan is often an appropriate measure of the schedule performance, it is by no means the only appropriate measure. However, the computational difficulties and the desirability of maximizing the production rate generally make makespan the most acceptable schedule performance criterion. (84) Also Ham et.al. (60) stated that makespan is the most appropriate criterion employed in production scheduling. This is why it was used as an objective function in this study.

3.2 The Flow-Shop Scheduling Problem :

In this section, there are five subsections covering the definition of flow-shop scheduling problem, and the solution techniques including the mathematical approach (with a brief presentation for branch and bound), the theoretical approach, and the heuristic approach. The literature for flow-shop scheduling heuristics is reviewed in this section.

3.2.1 Definition of the Flow-shop Scheduling Problem:

A flow shop is one in which all jobs follow essentially the same path from one machine to another. The shop contains (M) different machines, each job consists of (M) operations, each of which requires a different machine. The flow shop is characterized by a flow of work that is unidirectional, i.e. a flow shop contains a natural machine order (10), as will be shown in the assumptions of the study.

To find an optimum sequence of (J) jobs on a single machine, it may be necessary to examine (at least implicitly) each of the sequences corresponding to the (J!)

different permutations. And for a flow shop, to sequence (J) jobs on (M) machines, there are $(J!)^M$ possible combinations of operations sequences (assuming an arbitrary order for operations).

Because of the richness of the solution space for the general flow shop problem, solution approaches have normally added an important restriction to considerably simplify the problem, and reduce the search for an optimum. This restriction is to assure that not only do all the jobs follow the same sequence of operations on all machines, but also each machine has the same order of jobs. This is called the no-passing assumption. By adding this assumption, we restrict the size of the solution space to (J!) possible solutions, instead of $(J!)^M$. And in spite of this reduction, (J!) is still a very large number for practical problems. (27)

3.2.2 Solution Techniques:

The first major step towards developing a scheduling theory dates back to Johnson's work in 1954 (77), which described how to find the optimal solution for the two-stage flow-shop scheduling problem that minimizes the makespan. Most of the flow-shop scheduling procedures are mainly concerned with the criterion of minimum makespan, and are often extensions to or based upon Johnson's theorem. (60)

But in general, flow-shop scheduling problems including three machines or more are not easy to solve using an optimization technique. An important factor in solving the scheduling problems is the efficiency of the solution algorithm. The time complexity of an algorithm refers to its execution time for finding a solution, which for flow shop scheduling procedures is expressed as a function of the number of jobs. (60, 79)

Complexity theory deals with measuring and analyzing the difficulty of computing a solution to a given problem. The time complexity function of an algorithm is the amount of time that is necessary to execute the algorithm, as a function of the input length (n). In general, problems are divided into three main categories in terms of their solution difficulty: (77a)

1- Category "P" (polynomial): It consists of the problems that are solvable in polynomial time. If an algorithm can be found to solve a problem with a time complexity function that is polynomial in (n), then the problem is said to be tractable.

2- Category "U" (undecidable): This is the category of all problems for which it can be proven that there does not exist an algorithm giving a solution. This category contains the unsolvable problems based on the concept of "reducibility". Problem A is said to be reducible to problem B if, given a subroutine capable of solving problem B, one can construct an algorithm to solve problem A.

3- Category Non-Poly: This category in general contains all the problems which cannot be solved by polynomial-time algorithms. But according to Cook's definition (24a), NP class consists of all those problems for which a proposed solution can be checked in polynomial time. This means that he considers class P as a subset of class NP. Some authors distinguish between NP and Non-Poly classes, by saying that the Non-Poly class includes problems harder than those in NP class. So, NP is the area into which combinatorial problems typically fall. (77a)

Cook (24a) also defined an interesting category of problems called NP-complete problems. A problem is NP-complete if it lies in the class NP, and every problem in NP is polynomial-time reducible to it. This category is important because if any single NP-complete problem is ever shown to have a polynomial-time solution, then polynomial-time solutions can be constructed for all other problems in the category. (77a)

Most of the classical combinatorial problems (including the flow-shop scheduling problem) are NP-complete. Therefore no simple rule has been offered for determining the optimal solution in polynomial time. (23, 29) To cope with such problems in practice, one possible approach stems from the fact that near-optimal solutions will often be good enough.

But in general we can group the solution techniques into three categories, the mathematical approach, the theoretical approach, and the heuristic approach.

3.2.3 The Mathematical Approach:

One of the solution approaches consists of the general-purpose methodologies for solving the combinatorial problems, such as dynamic programming, and branch and bound. These methods have been applied to solve the flow-shop scheduling problems with three machines or more. But in most of the cases, the computation time and the memory required to keep track of the calculation are prohibitive even for small problems. (97)

3.2.3.1 The Branch and Bound Algorithm (B&B):

B & B is the solution technique used in solving many combinatorial problems. Like dynamic programming, it is an "implicit" enumeration technique, and again like dynamic programming it is an approach to optimization.

The basic B & B procedure was developed by Ignall, and Schrage in 1965. (74) The job sequence is constructed in a forward direction in proceeding down the branching tree. For each node on the tree, a lower bound on the makespan is obtained by considering the work remaining on each machine. Calculating the lower bound for a specific node will tell us whether to explore this branch of the tree any further or not, as will be discussed later in this subsection.

So, as its name implies, the approach consists of two fundamental procedures: branching: which is the process of partitioning a large problem into two or more subproblems, and bounding: which is the process of calculating a lower bound on the optimal solution of a given subproblem.

There are two main strategies that are mostly used in searching for the optimal solution. These are the frontier search (branching from the lowest bound), or Depth-first search. (10) Fig.(7) shows an example for both strategies.

(1) Frontier search, or branching from the lowest bound: if we have a problem of four jobs to be sequenced, first we simultaneously branch to each of the four nodes. Then we calculate the lower bound for each node. If we find for example that the node 2XXX has the least lower bound, we branch from this node to the next level, calculate the lower bounds, and so on. We always branch from the node with the current minimum lower bound.

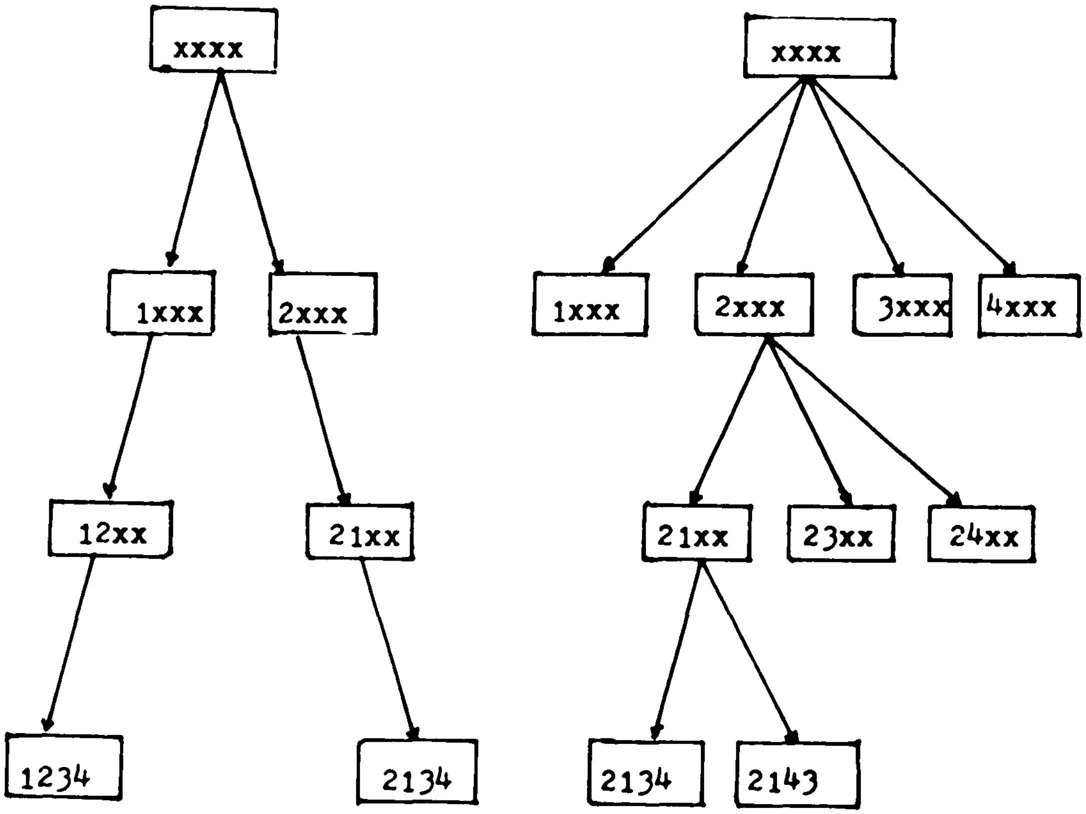
This strategy requires less calculation, because it chooses which branch to explore next in a more intelligent fashion. So, it usually finds an optimal solution faster. But this is bought at a cost of greater storage requirements.

(2) Depth-first search: In our search of the tree we select a branch and systematically work down it until we either eliminate it on the grounds of lower bound, or reach its final node, and consider its lower bound as the first "trial schedule". Then we proceed with the next branch, comparing

the lower bound at each node with the value of our "trial schedule". If it is greater than or equal to it, we know that we cannot improve upon the trial schedule by exploring this branch further, hence we eliminate it. But if the value of the lower bound of that node is less than the lower bound of the "trial schedule", we explore the branch further. When we arrive at a final node, and find that the schedule has a lower completion time, we consider this schedule as the new "trial schedule". We proceed in this process until tha last remaining "trial schedule" becomes the optimal schedule. This strategy requires very little computer storage, compared to the first one, but it may however require a great deal of computation.

French (40), mentioned that the number of operations required, and hence the time required to solve a problem is unpredictable, whatever search strategy is used. It might happen that the procedure has to fully explore virtually every node, in which case it would take as long as complete enumeration. Indeed, it might take longer because B & B involves more computation per node than complete enumeration. Theoretically, like dynamic programming, B & B always finds an optimal solution, but it may take prohibitively long to do so.

Ignall and Schrage (74) report that their algorithm for the $n/3/F/C_{max}$ problem requires an average about twice as much time for (n+1) jobs as it does for (n). Thus, if it



a. Depth-First Search

b. Frontier Search

Fig.(7) Search Strategies for the Optimal Solution in branch and bound takes one second to solve an n-job problem, it will take n^r to solve (n+r) problem (twelve days for r=20). (40)

Ashour(8), concluded from a comparative study that using branch and bound to solve a (7Jx4M) problem requires in some cases exploring 1322 nodes (as an observed maximum number in his experiment). And for (8Jx4M) problem, the number of explored nodes were 4884, and the number reached 6148 nodes to solve a problem of (8Jx3M), and 6392 nodes for a (10Jx3M) problem.

Baker (10) also mentioned that the branch and bound elimination approach has two inevitable disadvantages, which are typical of implicit enumeration methods. First the computation requirements will be severe for large problems. Second even for relatively small problems, there is no guarantee that the solution can be obtained quickly.

If the problem of sequencing J jobs on M machines using B&B is not easy to solve in the conventional flow shop, it will be much more complicated in the case of group scheduling. In this case we have to sequence F families, and to sequence J jobs within each of these families.

Figures (8, 9) show two examples to sequence three-family, two-job-each, and three-family,three-job-each problems to give an indication about the number of nodes that could be explored to solve these small problems using the B&B algorithm. This is also to show why the optimal solution is not used in this study. From the figures, we can see that increasing the number of jobs (within the family) from two to three jobs, leads to an increase in the number of nodes from 156 to 3825 nodes.

3.2.4 The Theoretical Approach:

The theoretical approach solves the flow-shop special-structure problems. Advances in scheduling techniques have shown that it is rather difficult to develop exact optimization techniques for the solution of the general scheduling problem of the flow shop. As a result of this awareness, several authors have considered special-structure flow-shop problems. The job processing times in these problems are not completely random, but bear a well-defined relationship to one another. (55) One of the most common examples of these problems is Johnson's generalization of his two-machine theorem to the three-machine case when the second machine is dominated. (77) A few authors (55, 57) developed several algorithms for solving special-structure flow-shop scheduling problems that have specific relationships among jobs processing times.

3.2.5 The Heuristic Approach:

The word "Heuristic" comes from the Greek word (heuriskein), meaning to discover. The term has been used by Simon and Newell, 1958 (122) to describe a particular approach to problem solving and decision making. (127)

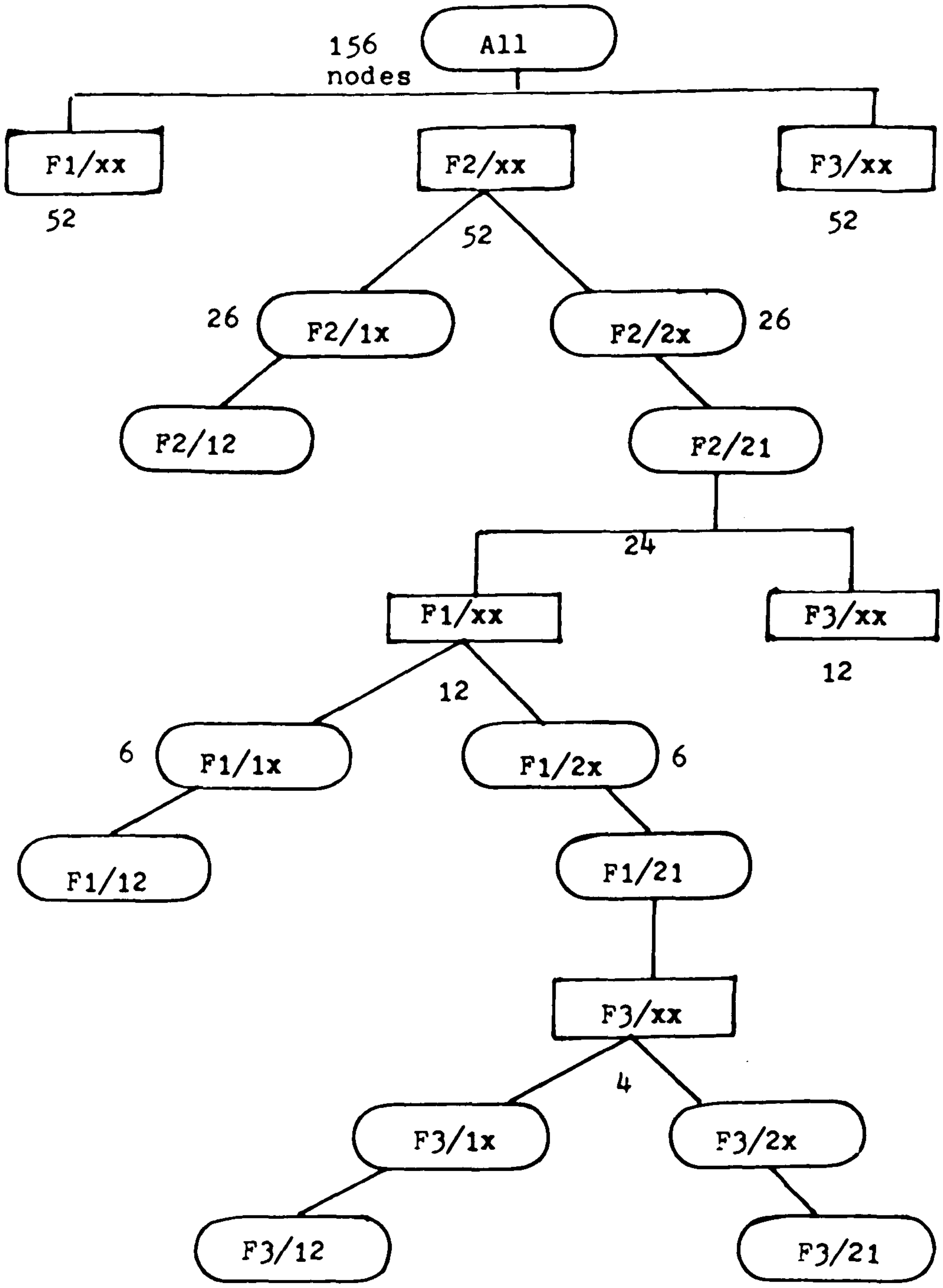


Fig.(8)

Number of Nodes for a Three-Family, Two-Job-Each Problem

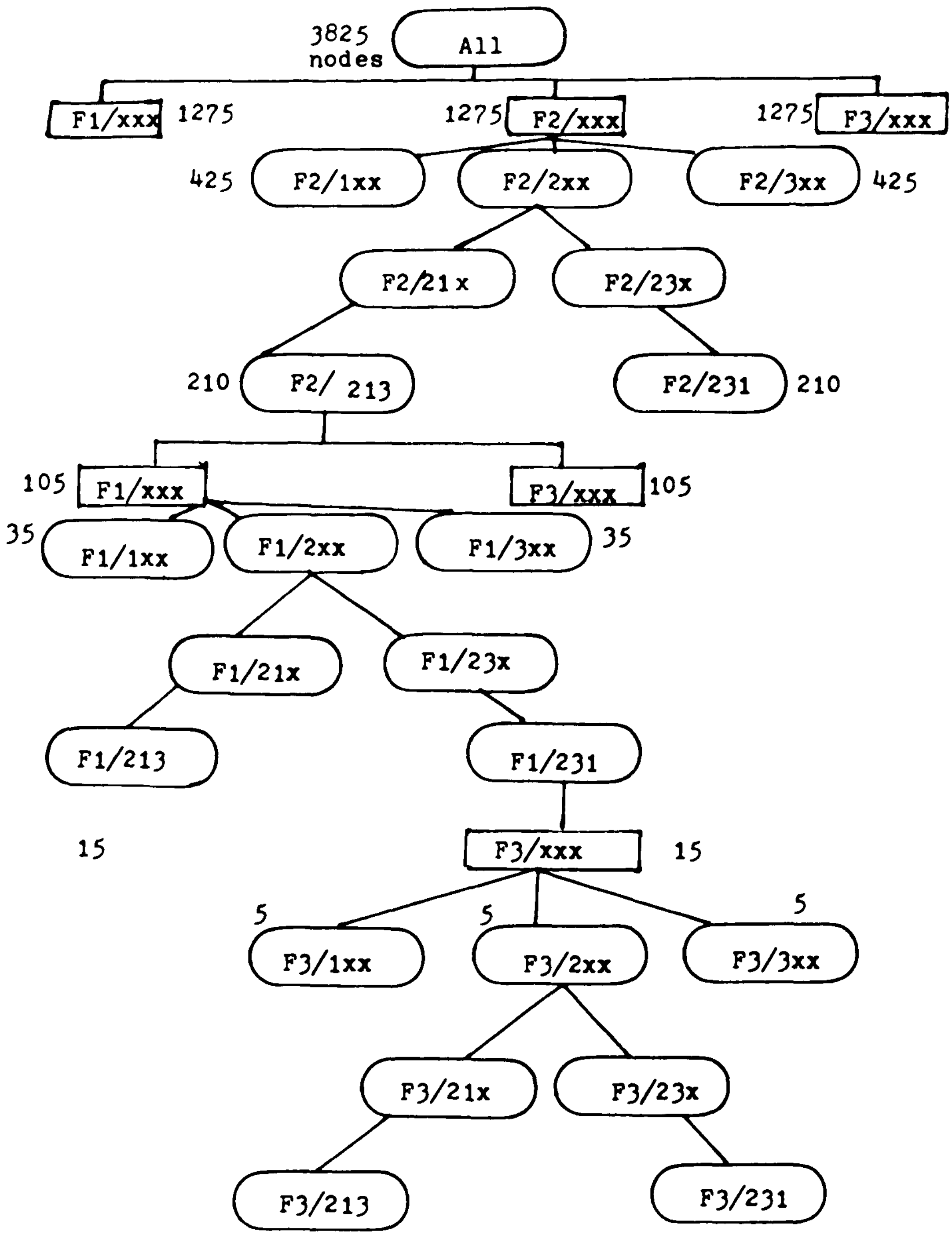


Fig.(9)

Number of Nodes for a Three-Family, Three-Job-Each Problem

Although both heuristics and optimizers use search reduction techniques to restrict the number of solutions that must actually be examined, optimizers guarantee that the area remaining to be searched (after reduction) includes an optimal solution, while heuristics do not. It is possible that heuristic search reduction may retain an optimum in the search area, but this is not guaranteed, as mentioned before. (27)

Knowing that most scheduling problems are NP-complete is valuable for directing efforts towards finding efficient approximation techniques for generating good feasible schedules rather than searching for possibly efficient algorithms that hold some promise of optimality. But the NP-hardness of a problem alone is not sufficient reason to resort to heuristic methods. It must also be so large that enumerative methods are intractable. So heuristic methods are not used when either constructive polynomial time solutions exist or implicit enumeration is computationally feasible. (40)

In practical situations, the management of a production facility may prefer using near-optimum algorithms rather than spending large amounts of money and computer time in search of optimum solutions that provide only a very slight improvement over the near-optimum solutions. (1)

Heuristic models utilize logic and common sense derived by observation and introspection. These models replace the mathematical ones, when formal analytical methods have little promise of being optimal. The essence of the heuristic approach is in the application of selective routines that reduce the size of the problem. Another kind of reduction is used in which a relatively simple rule is applied repeatedly until all decisions that must be made have been made. (121, 127)

Although heuristic procedures do not necessarily provide the optimal solution to the problem, they are for the most part an efficient and economical way of getting a good solution to the problem. (97)

3.2.5.1 Some Selected Heuristics:

French (40) in motivating Johnson's algorithm for the $n/2/F/F_{max}$ problem, argued that the jobs which were placed near the beginning of the schedule should have short processing time on machine one (m1) to minimize the idle time on machine two (m2). Similarly jobs which were placed near the end of the schedule should have short processing time on m2 to minimize the idle time on m1.

Some heuristics carry over this idea to the general $n/M/F/F_{max}$ problem. It is intuitively reasonable that jobs placed early in the schedule should have increasing processing times from machine to machine, whereas, those towards the end of the schedule should have times that tend to decrease in passing from machine to machine.

The following is a presentation of some of the most commonly used heuristics.

Petrov's method:

One of these heuristic methods that use the makespan as an objective function is Petrov's method (108). It is an extension of Johnson's algorithm for two stages to be applied to M stages, with M>2. The method transforms an M-machine flow-shop scheduling problem into a two-machine flow shop problem by dividing the matrix of job processing times into 2 parts.

The two fictitious processing times for each job are computed as follows:

$$P_j^A = \sum_{m=1}^{h} P_{j,m}$$

$$P_j^B = \sum_{m=\grave{h}}^{M} P_{j,m}$$

Where $h = \grave{h} = (M / 2)$, for even M, and

$h = \grave{h} = (M + 1) / 2$ for odd M.

Then Johnson's algorithm is used to solve the problem of n jobs, each of which has the two fictitious processing times, previously mentioned.

Palmer's heuristic:

Palmer (104) proposed another heuristic called the slope order index (SOI). It is based on the idea that jobs placed in early positions in the sequence should have increasing processing times from machine to machine, while late-position jobs have decreasing processing times to minimize the waiting time as will be seen in (4.2.1).

Gupta's heuristic:

In 1971, Gupta (53) suggested another heuristic, by extending Johnson's algorithm (the general case for three machines, where we can apply his two-machine algorithm if, and only if the second machine is dominated by either the first or the last machine).

Gupta calculates a slope index for each job to be used in sequencing jobs. His heuristic is similar to Palmer's except that he defined the slope index in a slightly different manner.

$$S_j = e_j \; / \min_{1 \leq k \leq M-1} (P_{j,k} + P_{j,k+1})$$

$$\text{where } e_j = \{ \; 1 \qquad \text{if } P_{j,1} < P_{j,M}$$

$$= \{ \; -1 \qquad \text{if } P_{j,1} \geq P_{j,M}$$

Ashour's heuristic:

Ashour (7) proposed and tested a heuristic called linear branch and bound (LBB), which is the first feasible solution found by a branch and bound procedure.

Page's heuristic:

Page (103) suggested that the methods commonly used for sorting (in data processing) might be successfully applied to the sequencing problem. Based on these sorting techniques, he developed four different heuristics:

1- The merging procedure first divides the list of jobs into (n) strings, each containing one job. Each successive pair of strings is then merged into a single ordered string containing two jobs. The ordering is based on the best makespan for the pair of jobs. Continuing this process, the number of jobs per string increases, and the number of strings decreases, until a single ordered string of n items is formed.

2- The pairing procedure is identical to that of merging, the only difference is to keep adjacent jobs permanently adjacent in any larger strings, once a string is formed.

3- The individual exchanging procedure starts with a given job order. It then checks whether each successive pair of adjacent jobs should remain in their order or they should exchange positions based on the makespans. The process continues until we do not find any improvement by changing positions of any adjacent jobs.

4- The group exchanging procedure is similar to the previous one except that groups of jobs are exchanged rather than individual jobs. The procedure starts by dividing the n jobs into two groups. An attempt is made to improve the makespan by switching the order of the two groups. Then each of the groups is split into two halves, and group exchanges are attempted again. Every time the number of jobs in a group is halved, which doubles the number of groups. The last phase applies a single exchange procedure to the current best sequence.

CDS's heuristic:

Campbell, Dudek, and Smith (20) generalized Johnson's 3-machine algorithm to generate a set of M-1 artificial 2-machine problems from the original M-machine One. Each of these is solved using Johnson's algorithm. The procedures of the heuristic will be discussed in detail in (4.2.2).

Dannenbring's heuristics:

Dannenbring (27, 28) developed three new heuristic procedures which generate a good starting solution using a weighting scheme (similar to that for the SOI and CDS methods) to form a single 2-machine subproblem, to be solved using Johnson's algorithm. Then he improves this solution by means of a fine tuning device through transposing adjacent jobs. The following are three heuristic procedures that have been developed by Dannenbring.

1- Rapid access procedure uses an index similar to that of slope order's, and it is developed the same way as CDS's. A single two-machine subproblem is formed, where the processing times are determined as follows:

$$P_{i,1} = {}^{m}_{j=1} \; (m-j+1)\, t_{ij}$$

$$P_{i,2} = {}^{m}_{j=1} \; (j)\, t_{ij}$$

where t_{ij} : processing time for the ith job on the jth machine in the main problem.

$P_{i,j}$: subproblem processing time for the ith job on the jth machine.

2- Rapid access with close order search improves the previous solution by transposing each pair of adjacent jobs. Each of these (n-1) neighbors are examined for possible improvement in makespan.

3- Rapid exchange with extensive search does not terminate the previous process after one set of interchanges. It uses the best immediate neighbor to generate further neighbors through additional interchanges. The process continues until no improvement can be found.

NEH's heuristic:

Nawaz, Enscore, and Ham (97) proposed an algorithm that is based on the assumption that a job with more total processing time on all machines should be given a higher priority than a job with less total processing time, as will be shown in (4.2.3).

3.2.5.2 Comparative Studies of Heuristics:

The following are some of the studies that compared different flow-shop scheduling heuristics:

Campbell et.al.(20) tested their algorithm (CDS) and examined its performance as compared to Palmer's heuristic in several problems. They found that the CDS's heuristic was generally more effective for both small and large problems. In addition, the computer times required were of the same order of magnitude for $N \leq 20$. Only in somewhat larger problems would the question of trading-off solution value for computing time arises.

Dannenbring (27, 28) compared the performance of different heuristic procedures and came up with a conclusion that his heuristic (rapid access with extensive search) is the least biased (the lowest average error), and the most consistent (the lowest average square error) of the procedures tested on the small problems (3,4,5,6 jobs combined with 3,4,5,6,7,8,9, and 10 machines). He also concluded that the CDS's heuristic achieves optimality roughly 55% of the time, and SOI heuristic achieves optimality about 30% of the time.

Nawaz et.al.(97) mentioned that in a recent study by Setiaputra (119), five well-known heuristics (CDS's, Dannenbring's (Rapid Access), Gupta's, Palmer's, and Petrov's) were statistically evaluated for performance, and the CDS's procedure was shown to be superior.

In the same study by Nawaz et.al.(97) they evaluated their proposed heuristic by solving a total of 2764 problems (with number of machines and jobs being set at various levels between 4 and 25). They compared the results with those obtained by applying the CDS's procedure. They concluded that their proposed heuristic (NEH) performs extremely well in comparison with the CDS's, in terms of the number of times it gave better solutions than CDS's.

In other studies by Park (105), and by Park et.al.(106), they compared the performance of different flow-shop scheduling heuristics and they stated that:

> "It is clear that NEH is the least biased, and the best-operated of the heuristics tested on the small static flow shop problems (3 to 9 jobs combined with 4 to 20 machines), with respect to makespan, and that CDS is the next best.
> "as in the case of small-size problems, it is clear from the test results that for minimizing makespan in large static problems (15 to 30 jobs combined with 4 to 20 machines), NEH is the least biased and most effective..."

From the results of various comparative studies of flow-shop scheduling procedures in a static and deterministic environment, (20, 27, 97, 105, 124), it is clear that CDS's and NEH's heuristics proved to be of the most efficient heuristics that use the makespan. And this is why they are included in this study to be tested for GT flow-line scheduling problem in comparison with the other two newly modified heuristics (MOD1, and MOD2).

3.3 GT Scheduling Techniques: (Literature Review)

We consider in this section, the importance of accurate scheduling in GT applications, and we review the literature of scheduling for the GT center, the GT cell, and the GT flow line.

3.3.1 Integration of Scheduling and GT:

One of the major areas for GT application is production scheduling. Production scheduling is greatly simplified by GT. The scope of the problem is reduced from that of a large portion of the shop to a small group of machines in the group technology cell, and to a flow shop in case of GT flow line. (130)

Job-shop manufacturing is characterized by a complex work-flow structure and highly variable manufacturing times, which complicate the production planning and control activities. (100)

High throughput times, extensive delays, high work-in-process inventory, low utilization of labor and machinery demanding supervision and control of production are the basic indicators related to the problems encountered in production planning and control activities in job-shop manufacturing. (100)

The obvious problems of the functional job-shop layout can be drastically reduced through the implementation of a GT layout, which allows more control over the flow of a part through the shop (cell). (32) Once a part is classified into a distinct part family, in effect the decision has been already made as where it will travel in the shop, and what operations must be performed. (11)

Proper scheduling is an integral part of GT, because good scheduling combined with reduced setup time and reduced transportation will result in a significant cost reduction. The most obvious benefit is reduced total production time. With this reduction, production can more closely match demand, so that inventory can be reduced and parts can be produced on schedule. (19, 130)

GT requires the production cycles to be short, and hence the run quantity (or batch size) to be small. Due to the short cycles, only demanded parts are produced in the preceding cycle of their assembly. This policy allows more flexibility to adapt to changes in demand, and also reduces the work-in-process inventory. So, when scheduling is performed accurately, it can help in implementing this policy. Based on that, quantities of inventory are greatly reduced, since each part is manufactured just before it is needed (just in time). (2, 52, 98, 117)

The literature review on scheduling for GT is divided into three parts: scheduling for a GT center, for a GT cell, and for a GT flow line.

3.3.2 Scheduling for a GT Center:

One of the early works in this area is by Emmons (34). He analyzed the scheduling problem for a single stage, considering that each group consists of only one job, to minimize total tardiness. This approach of using the conventional scheduling model to be applied in the GT case without modification has been used by many researchers.

Nakamura, Yoshida, and Hitomi (96) developed efficient algorithms to find the optimal or near-optimal group schedules. Regarding job sequence, they considered 2 cases: predetermined job sequences, and not predetermined job sequences to minimize total tardiness.

Foo and Wager (38) differentiated between cyclic scheduling where the production of an entire family is to be repeated sequentially (the solution can be sought by looking at the problem as a travelling salesman problem) and the acyclic problem which is defined as the case of having only selected parts from a family to be made, and the family is not to be repeated. They solved the problem of acyclic case for a single family on a single facility.

Ozden, Egbelu, and Iyer (102) used dynamic programming to solve the group scheduling problem on a single facility.

3.3.3 Scheduling for a GT Cell:

A study has been carried out by PERA (Production Engineering Research Association) (107), using a variety of scheduling strategies in three cells to test different scheduling rules. The scheduling rules which gave better schedules on average were SIO (shortest imminent operation), and COVERT (cost over time value).

Vaithianathani, and McRoberts (132) used the decomposition approach based on setup similarities to identify subgroups. Scheduling among subgroups used different criteria in assigning priorities. Scheduling within subgroups was done using the SPT (shortest processing time). They concluded after solving several problems that scheduling among subgroups based on the remaining slack was the best overall performer.

Sundaram (128) used some well-known flowshop scheduling algorithms for GT cell scheduling applications. But his procedure has a limitation of handling only 3 machines.

Elgomayel and Nader (33) presented a research project that uses a computer program called OPSSP, "Optimization of Setup and Scheduling of Parts", to optimize the choice of toolings, tooling setups, and scheduling of component parts to be machined on different kinds of turning machines.

Sundaram (129) proposed 2 heuristics for scheduling jobs in a GT cell. The results of these heuristics were compared with the optimal solution obtained using integer programming.

One of the most important studies in the area of scheduling in a GT cell was carried out by Mosier, Elvers, And Kelly (94). A simulation model for each of a GT shop, and non-GT job shop was developed. They evaluated different scheduling rules for GT manufacturing in a dynamic case, by comparing them with each other and with other job-shop scheduling rules. In general, the family scheduling procedures were better than their corresponding job rules in terms of mean flow time and mean lateness, which was expected with GT implementation. (95)

Jacobs, and Bragg (76) modified Mosier's idea, and developed what they called the "repetitive lots" procedure. It is a scheduling procedure that capitalizes on the sequence dependency of setup times in shops, where the queue of waiting jobs is scanned seeking a job identical to the job that was just processed on a machine (to eliminate setup time).

Flynn (36) extended the same previous idea to model the job shop and GT cell. A computer simulation was designed to compare a shop configured as a GT shop, with the same shop configured as a traditional job shop, and as a hybrid (which combined features of both). The study investigated the repetitive lots procedures as an approach to improve performance in a GT shop. (37)

3.3.4 Scheduling for a GT Flow-Line:

The published studies in the area of scheduling for a GT flow line are only few compared with the large number of publications about the GT cell. Although many papers have been published about the deterministic, static GT flow line, there is no complete comparative study to evaluate the different solution techniques available. The following is a brief presentation for the relevant studies.

Hitomi and Ham (63) extended Petrov's method that transforms a K-machine problem into a 2-machine problem, by dividing the job processing times matrix into 2 parts to apply Johnson's algorithm. They solved in their paper a numerical example using Petrov's modified heuristic, and compared the results with the optimal solution obtained by branch and bound.

Hitomi, Nakamura, Yoshida, and Okuda (65) investigated the effect of flow patterns (job shop, flow shop, and near flow shop) on flow time performance under dynamic conditions by simulation. The results showed no significant difference and indicated the importance of the ratio of setup time to processing times.

Hitomi and Ham (64) used the branch and bound algorithm to solve the problem of a static flow line GT shop to minimize the total flow time. Their study also determined the optimal machining speed for all the jobs on all the stages.

Yoshida, and Hitomi (138) discussed in their paper the two-machine flow-shop scheduling problem where setup times are separated from processing times. A simple decision rule which is an extension of Johnson's theorem is used to minimize the total elapsed time. Then in an attempt to compare two optimal schedules, with setup time separated, and with setup time included, a numerical example is given.

Ham, Hitomi, Nakamura, and Yoshida (58) incorporated in their group scheduling model variable processing times and cost depending on the machining condition. The optimal group schedules are first determined so as to minimize the number of tardy jobs. Because having large due dates leads to many optimal schedules, they used the total flow time as a secondry criterion. Once a group schedule minimizing the total flow time with the minimum number of tardy jobs is decided, the optimal machining conditions which minimize production cost are determined by utilizing the idle times of the schedule. They used the modified branch and bound algorithm to solve the problem. They solved only one numerical example to verify the effectiveness of the proposed algorithm.

Taylor and Ham (130) used Petrov's modified method to sequence a single part family and a set of part families in a flow line using a computer program written in BASIC.

Cho, Enscore, and Ham (22) reported an algorithm for minimizing the total tardiness in a multi-stage group scheduling application.

Radharamanan (110) used the same modified method of Petrov to sequence jobs of a single part family in a static GT flow line (using a computer program written in FORTRAN).

Vakharia and Wemmerlov (133, 137) investigated the performance of various static and dynamic scheduling procedures applied in a dynamic stochastic flow-line environment. The investigated rules included two static flow-shop heuristics (NEH and CDS), and two dynamic dispatching rules for job shop (FCFS and SLACK). The same rules were also applied for family application with three performance measures being: flow time, lateness, and ratio of early/late jobs. The results showed that the family rules performed better, particularly the CDS with respect to flow time.

From the previous studies, we can conclude that no studies have been done so far to evaluate the efficiency of different scheduling heuristics in the GT flow-line application, in a static and deterministic situation. The only complete simulation study carried out in this field is Vakharia, and Wemmerlov's study.

The difference between their study and the current one is that the former assumes a dynamic stochastic case, and the current study considers the static deterministic case. Sometimes, in the stochastic case, if the distribution of a variable cannot be determined correctly, assuming a deterministic situation may provide more accurate results.

Vakharia and Wemmerlov's study is based on a simulation model for a five-stage pure flow-line cell (a queuing model). The variables considered in their study are number of families, mean time between job arrivals, and family setup-to-job processing time ratio.

The current study is a computer-based investigation and comparison of five flow-shop scheduling heuristics. Two of these heuristics are major contribution of this study as new versions of NEH. The variables considered in the study are problem size, range of family/ group setup time, number of families, number of machines, and number of jobs.

CHAPTER FOUR

HEURISTICS EVALUATION PROCEDURES

This chapter covers the procedures of heuristics evaluation including the problem (its mathematical definition, assumptions, and notations), the procedures of selected heuristics, and the evaluation measures of solutions. The chapter also contains the hypotheses of the study, the procedures to test them, and the statistical tests used.

4.1 The Problem to be Studied:

This subsection discusses the mathematical definition of the problem, the assumptions of group sechduling and of the static deterministic flow shop. The notations used in heuristics procedures are also given.

4.1.1 The Mathematical Definition of the Problem:

Consider the problem of scheduling a number of groups or families of parts ($F = \{f_1, f_2, \ldots., f_F\}$), each family (f) consisting of a number of jobs ($J = \{j_1, j_2, \ldots., j_J\}$), each job (j) to be processed on a flow line that consists of a number of stages or machines (M), as will be discussed in subsection (4.1.2).

The scheduling problem for a GT application can be called "group scheduling" problem, and it can be solved by finding the best group sequences (that include both family and job sequences) which minimize the total flow time or makespan (MS).

The performance criterion used is the makespan which can be defined as the total elapsed time, or the total flow time or the maximum completion time. It equals the completion time of the last job (in the last family) on the last machine, and includes both processing and waiting times. The formula for calculating the makespan is given in subsection (4.2.6).

4.1.2 Assumptions of the Study:

These assumptions are divided into two parts, group-technology assumptions, and the general assumptions of the static deterministic flow shop.

(1) GT scheduling assumptions:

1- Jobs are already grouped into families.

2- Machines in the cell are arranged in a flow line.

3- Changing over between families require setup.

4- Setup times are independent of the sequence.

5- Family sequence and job sequence are identical on all stages of the manufacturing system (no passing).

6- Family processing time consists of the family/ group setup time plus the sum of the processing times for the jobs contained in this family.

(2) General assumptions: (static and deterministic flow shop)

1- Jobs are available simultaneuosly at time zero (static case).

2- Each machine is continuously available for processing jobs at any time.

3- Jobs consist of a strictly ordered sequence of operations (production technological order).

4- Time required to complete processing a job = setup time + machining time, i.e. job processing time includes any required setup time. Processing time is deterministic, known in advance.

5- Each operation can be performed by only one machine.

6- There is only one machine of each type in the work shop.

7- Preemption is not allowed : once processing begins on a job, no interruption is allowed.

8- The processing time of successive operations of a particular job may not be overlapped.

9- Each machine can handle at most one operation at a time.

10- Intermediate transportation times are ignored in processing times.

11- Each job requires some processing on each machine (straight flow shop).

4.1.3 Notations:

The following are the notations used in the procedures of different selected heuristics, and their definitions.

F : total number of families.

f : family index ($f=1,2,3.....,F$).

J_f : total number of jobs, members of group or family (f).

j : job index ($j= 1,2,......,J_f$).

M : total number of machines in the flow shop.

m : machine index ($m=1,2,3,....,M$).

P_{fjm} : processing time of job (j), a member of family (f), on machine (m).

S_{fm} : setup time of family (f) on machine (m).

K : an index for the ordered list (used in NEH, MOD1, and MOD2).

U : the index of subproblems ($U=1,2,......,M-1$) (used in CDS).

D_f : difference between processing time of family (f) on the last machine M, and its processing time on the first machine (used in MOD1).

D_{fj} : difference between processing time of job j (a member of family f) on the last machine, and its processing time on the first machine (used in MOD1).

4.2 Procedures of the Heuristics :

In this subsection, the procedures of the five heuristics are presented. These heuristics are SOI, CDS, NEH, MOD1, and MOD2.

4.2.1 Palmer's Heuristic (Slope Order Index) :

Palmer's idea (104) was to give each job a slope index which gives its largest value to those jobs having the strongest tendency to progress from short to long processing times as they pass from machine to another i.e. jobs placed early in the sequence should have increasing processing times (from machine to machine), while jobs assigned to late positions should have decreasing processing times.

The problem can be viewed much like a jigsaw puzzle, where the order leads to a perfect match. As seen in fig.(10), when the job order is reversed the pieces no longer fit well together, and the result is no longer the total processing time of the first case. (27)

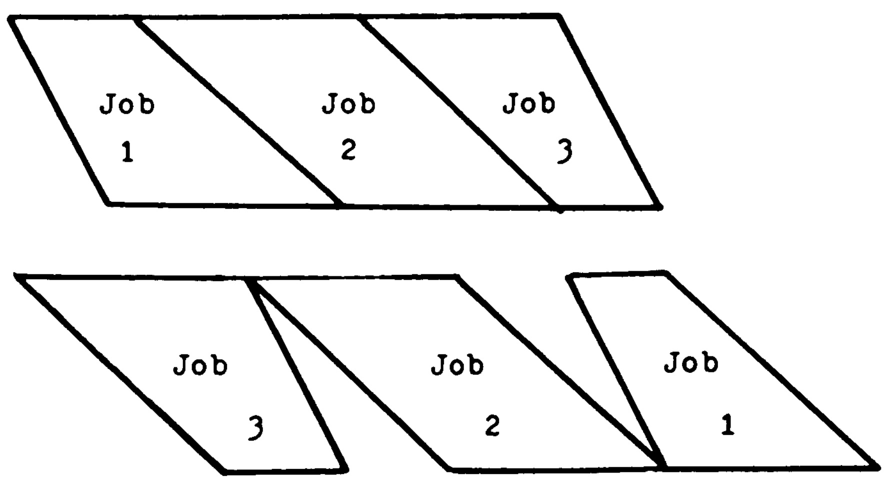


Fig.(10) Effects of Different Sequences on Makespan (27)

Fig.(10) Effects of Different Sequences on Makespan (27)

Thus, when the jobs are scheduled in a decreasing order of slope index, we might expect to find a near-optimal schedule.

The slope index (SI) for a job (j) in general, is given by:

$$SI_j = -(M-1)P_{j1} - (M-3)P_{j2} - \ldots\ldots + (M-3)P_{jM-1} + (M-1)P_{jM}$$

$$SI = -\sum_{m=1}^{M} \left[\left(\frac{M-(2m-1)}{2} \right) * P_{jm} \right]$$

In general, the same concept will be used to sequence the families first, then it will be used to sequence the jobs within each family.

Creation of family sequence:

To find the best sequence for the (group scheduling problem), using this heuristic, the procedure starts with sequencing the families first. The families are sequenced as if they were individual jobs, with family processing time on each machine that equals the sum of its jobs' processing times on that machine, plus its setup time on that machine.

The following are the steps to be followed first in sequencing families, and then (for the best family sequence), in sequencing the jobs within each family.

1- For each family, compute total required time on each machine (m), that equals the sum of all its jobs processing times on machine (m) plus the group/ family setup time on machine (m).

$$P_{fm} = \sum_{j=1}^{J_f} P_{fjm} + S_{fm}$$

2- Calculate SI for each family, using the following equation:

$$SI_f = -\sum_{m=1}^{M} \left[\left(\frac{M - (2m-1)}{2} \right) * P_{fm} \right]$$

3- Arrange the families in a descending order of their slope index (SI_f). This order will give the best family sequence. Then the next step is to sequence jobs within each family.

<u>Creation of job sequence (within a family)</u> :

1- For each family from f = 1 to F, do the following:

2- Compute SI_{fj} for each job in the family

$$SI_{fj} = -\sum_{m=1}^{M} \left[\left(\frac{M - (2m-1)}{2} \right) * P_{fjm} \right.$$

3- Sequence the jobs of family (f) in a descending order of their slope index (SI_{fj}).

4- Compute the makespan (MS) for the previous final sequence (groups/families and jobs), taking into consideration the setup times of the groups, using the formula given in subsection (4.2.6).

4.2.2 CDS's Heuristic (Campbell, Dudek, and Smith) :(20)

The CDS's heuristic corresponds to a multi-stage use of Johnson's rule (for two machines) applied to a new problem, derived from the original one. CDS's heuristic generates a set of (M-1) artificial two-machine problems from the original M-machine problem. Each of these problems is then solved using Johnson's algorithm. The best of the (M-1) solutions becomes the heuristic solution to the M-machine problem.

In general, we can say that the original problem is divided into (M-1) stages or subproblems as follows:

For U=1, (subproblem 1): Johnson's rule is applied to the first and the last (Mth) operations only, as the two machines. (other intermediate operations are ignored).

For U=2, (subproblem 2): Johnson's rule is applied to the sums of the processing times for the first two and the last two operations.

For U = 3, 4, ..., (M-1), the same procedure is followed to form the subproblem, adding one additional operation in each case, then Johnson's rule is used to solve it. Out of these (M-1) solutions, the one that gives the shortest makespan becomes the solution for the original M-stage problem.

<u>Creation of family sequence:</u>

For each subproblem (U) $(U=1,2,\ldots,M-1)$

1- For each family $(f=1,2,\ldots F)$, compute $t_f^{u,1}$ & $t_f^{u,2}$

$t_f^{u,1}$:artificial processing time of family f on the first stage for the Uth subproblem.

$t_f^{u,2}$:artificial processing time of family f on the second stage for the Uth subproblem.

$$t_f^{u,1} = \sum_{m=1}^{u} \left(\sum_{j=1}^{J_f} P_{fjm} + S_{fm}\right)$$

$$t_f^{u,2} = \sum_{m=M-u+1}^{M} \left(\sum_{j=1}^{J_f} P_{fjm} + S_{fm}\right)$$

2- Solve each subproblem using Johnson's algorithm, and compute the makespan.

3- Select the final sequence for the families or groups based on the shortest makespan for the M-1 subproblems considered.

<u>Creation of job sequence (within each family)</u> :

For each family $(f=1,2,\ldots,F)$, do the following :

1- For each subproblem U,

compute the artificial processing times for each job

$$t_{fj}^{u,1} = \sum_{m=1}^{u} P_{fjm} \quad \text{,and}$$

$$t_{fj}^{u,2} = \sum_{m=M-u+1}^{M} P_{fjm}$$

3- Solve each subproblem using Johnson's method, and compute its makespan.

4- Select the sequence having the shortest makespan.

4.2.3 NEH's Heuristic (Nawaz, Enscore, and Ham): (97)

This heuristic is based on the assumption that a job with more total processing time on all machines should be given a higher priority. It builds successively larger sequences by entering a new job in all possible positions without disturbing the relative order of the previous partial sequence.

It starts with the first two jobs (in the list) having the highest total processing times. The best partial sequence for these two jobs is found by computing the makespan (MS) for the two possible partial schedules. The relative positions of these two jobs with respect to each other are fixed in the remaining steps.

Next, the job with the third highest total processing time is selected, and placed at the beginning, middle, and end of the partial sequence found. Then, these three partial sequences are tested based on their makespan. The best partial sequence will fix the relative positions of these three jobs for the remaining steps.

The process is repeated until all jobs are fixed, and a complete sequence is found.

Creation of family sequence:

1- For each family compute total required time on all machines

$$P_f = \sum_{m=1}^{M} [\sum_{j=1}^{J_f} P_{fjm} + S_{fm}]$$

2- Arrange the families in a descending order of P_f.

Let K be an index for the ordered list.

3- Pick the first and second families in the list, find the best sequence for these two families by calculating the makespan for the two possible sequences. Let K = 3.

4- If K > F, stop, else go to (5).

5- Pick the family in the Kth position in the list, find the best sequence by placing it at all possible K positions in the partial sequence found in the previous step (without changing the relative positions of the families already assigned).

Select the sequence that yields the minimum makespan.

6- Set K = K+1, go to (4).

Creation of job sequence:

For each family, from f = 1 to F...do the following:

1- Calculate for each job the total processing times on all machines.

$$P_{fj} = \sum_{m=1}^{M} P_{fjm}$$

2- The jobs within the family are sequenced in a descending order of P_{fj} , and the same steps of sequencing families are followed to sequence jobs.

Let K be an index for the ordered list.

3- Pick the first and second jobs in the list, find the best sequence for these two jobs by calculating the makespan for the two possible sequences. Let K = 3.

4- If $K > J_f$, stop, else go to (5).

5- Pick the job in the Kth position in the list, find the best sequence by placing it at all possible K positions in the partial sequence found in the previous step (without changing the relative positions of the already assigned jobs).

Select the sequence that yields the shortest makespan,

6- Set K = K+1, go to (4).

The best sequence will be obtained after completing sequencing the jobs within all the families.

4.2.4 The First Modified Heuristic: (MOD1) :

This heuristic, an original contribution of this study, is a modified version of NEH. A general rule for heuristic procedures is to decompose the original problem of (n) jobs and (m) machines into some kind of n-job, two-machine problems. In Johnsons algorithm for the $n/2/F/F_{max}$ problem, it is argued that the jobs which are placed near the beginning of the schedule should have short processing times on m1 to minimize idle time on m2. Similarly jobs which are placed near the end of the schedule should have short processing times on m2 to minimize idle time on m1.

The first modified heuristic (MOD1) carries over these ideas to scheduling jobs in a GT flow line. The procedure starts with ranking jobs in a preliminary list based on the assumption that a job with a larger difference between its processing time on the last machine and its processing time on the first machine should be given a higher priority than a job with a smaller difference, to reduce the jobs waiting time on both machines. This can lead to a reduction in the total waiting time, and an improvement in the makespan. So, the larger the difference between the last and the first machine processing time for a job or a family, the higher the priority to be first in the list. The rest of the procedures are the same as NEH's, as follows.

<u>Creation of family sequence :</u>

1- For each family, compute the total required time on the first machine, and the total required time on the last machine.

$$P_{f1} = \sum_{j=1}^{J_f} P_{fj1} + S_{f1}$$

$$P_{fM} = \sum_{j=1}^{J_f} P_{fjM} + S_{fM}$$

2- For each family compute D_f, which is the difference between the total time on the last machine and the total time on the first machine.

$$D_f = [P_{fM} - P_{f1}]$$

3- Arrange the families in a descending order of D_f. Let K be an index for the ordered list.

4- Pick the first and second families in the list, find the best sequence for these two families by calculating the makespan for the two possible sequences. Let K = 3.

5- If K > F, stop, else go to (6).

6- Pick the family in the Kth position in the list, find the best sequence by placing it at all possible K positions in the partial sequence found in the previous step (without changing the relative positions of the already assigned families).

Select the sequence that yields the minimum makespan.

7- Set K = K+1, go to (5).

<u>Creation of job sequence:</u>

The same steps followed to sequence families are used in sequencing the jobs within families.

For each family, from f = 1 to F...do the following:

1- Calculate for each job the difference between processing time on the last machine, and processing time on the first machine.

$$D_{fj} = [\ P_{fjM} - P_{fj1}\ ]$$

2- Arrange all jobs in the family in a descending order of D_{fj}. Let K be an index for the ordered list.

3- Pick the first and second jobs in the list, find the best sequence for these two jobs by calculating the makespan for the two possible sequences. Let K = 3.

4- If K > J_f, stop, else go to (5).

5- Pick the job in the Kth position in the list, find the best sequence by placing it at all possible K positions in the partial sequence found in the previous step (without changing the relative positions of the already assigned jobs).

Select the sequence that yields the minimum makespan.

6- Set K = K+1, go to (4).

The best sequence will be obtained after completing sequencing the jobs within all the families.

4.2.5 The Second Modified Heuristic (MOD2) :

This original heuristic is a similar modified version of NEH. It follows the same steps as in MOD1 except for the first step that develops a preliminary ordered list to choose from in the process of creating successively larger sequences. MOD2 is based on the assumption that a job with shorter processing time on m1 should be given a higher priority than a job with longer processing time on the first machine. By assigning these priorities, the waiting time of each job for processing on the first machine is reduced, which leads to a reduction in the total waiting time, and hence, the makespan (total flow time) improves.
Then, the procedure goes on with the same steps as in MOD1, as follows.

It should be mentioned here that the adjustments of these heuristics to be applicable for GT flow line are based on the same approach followed by Ham, and Hitomi to adjust Petrov's algorithm (63).

Creation of family sequence:

1- For each family compute the total required time on machine one (P_{f1}).

$$P_{f1} = [\sum_{j=1}^{J_f} P_{fj1} + S_{f1}]$$

2- Arrange the families in an ascending order of P_{f1}. Let K be an index for the ordered list.

3- Pick the first and second families in the list, find the best sequence for these two families by calculating the makespan for the two possible sequences. Let K = 3.

4- If K > F, stop, else go to (5).

5- Pick the family in the Kth position in the list, find the best sequence by placing it at all possible K positions in the partial sequence found in the previous step (without changing the relative positions of the already assigned families).

Select the sequence that yields the minimum makespan.

6- Set K = K+1, go to (4).

Creation of job sequence:

For each family, from f = 1 to F...do the following:

1- Arrange all the jobs in the family in an ascending order of their processing times on the first machine.

Let K be an index for the ordered list.

2- Pick the first and second jobs in the list, find the best sequence for these two jobs by calculating the makespan for the two possible sequences. Let K = 3.

3- If $K > J_f$, stop, else go to (4).

4- Pick the job in the Kth position in the list, find the best sequence by placing it at all possible K positions in the partial sequence found in the previous step (without changing the relative positions of the already assigned jobs).

Select the sequence that yields the minimum makespan.

5- Set K = K+1, go to (3).

4.3 <u>Performance Criterion (Makespan) Computation</u>:

The makespan (MS), as defined before, is the total production time, or the maximum flow time. It is defined as follows:

$$MS = C_{[F][J]M}$$

Where $C_{[F][J]M}$ is the completion time of the last job in the sequence [J], in the last family in the sequence [F], on the last machine M. So, the subscripts in the [ ] indicate the order in the sequence.

If we define $C_{[f][j]m}$ as the completion time of the jth job in the sequence, in the fth family in the sequence, on any machine m, the makespan can be calculated iteratively, starting with the completion time of the first job of a family as follows :

For j = 1 (the special case that considers the family setup time)

$$C_{[f][1]m} = \max (C_{[f][1]m-1} , C_{[f-1][J]m} + S_{[f]m}) + P_{[f][1]m}$$

For $j \neq 1$ (the general case of completion time of any other job)

$$C_{[f][j]m} = \max (C_{[f][j]m-1} , C_{[f][j-1]m}) + P_{[f][j]m}$$

Considering that

$$C_{(f)(j)(0)} = C_{(0)(j)(m)} = C_{(f)(0)(m)} = 0$$

4.4 <u>Evaluation Measures for Heuristics Solutions</u> :

A number of measures can be used to evaluate the "goodness" of a solution in terms a solution standard. This solution standard may be the optimal solution, or the best available solution. (27)

4.4.1 <u>The Achievement Measures</u> :

The achievement measures record the percentage of time a heuristic provides solutions that achieve some standard. The normal situation is to use the optimal solution as a standard. A measure of this type reflects the proportion of times that the heuristic solution is optimal. Since the optimal solution is not obtainable in this study, this standard will not be used.

When the optimal solution cannot be obtained, a measure can record the proportion of times a specific heuristic provides the best solution among other heuristics. This measure is called (BEST).

Another ratio or measure that can be grouped with the achievement measures is the proportion of times a specific heuristic provides the worst solution among other heuristics. This measure is called (WORST).

In general, these measures should not be used independently, but should be used with other measures such as any of the comparative measures. There are two major reasons for not depending on the achievement measures only.

First, a heuristic that achieves the highest percentage in providing the best solution can not be considered the best heuristic or the most efficient one in general. Consistency of performance is a very important factor here. Although a specific heuristic may be better in providing the best solution most of the times, the same heuristic may provide the worst solutions in the few times it does not provide the best solution. This means that when it misses, it misses by a wide margin. On the other hand, we may have a heuristic which provides solutions that usually come close to the best, but rarely reach it. This means that the later heuristic is a more consistent performer than the first one, which may be preferrable in most of the cases.

What has been discussed about the "BEST" ratio and the consistency of a heuristic is applicable to other achievement measures, including "OPTIMAL", and "WORST". This implies that a heuristic that yields the highest "WORST" ratio is not necessarily the worst heuristic or the least efficient one in general. But this depends on how far its "Worst" solution is from the best one. The difference between the values of the best and the worst solutions may be very small, and this heuristic at the same time performs more consistently than the other heuristics, as mentioned before.

The second reason for not depending only on the achievement ratios is that, this type of measures is sensitive to the individual characteristics of the problem. For example, we cannot generalize the results of small problems on large ones. One of the characteristics of scheduling problems (or combinatorial problems) is that the number of alternative optimal solutions for small combinatorial problems is quite high. So, it is quite possible for any heuristic to find one of these alternative optimal solutions. This means that, it is misleading to use the achievement ratios of a specific heuristic on small problems to predict its performance on large problems.

Generally speaking, the achievement measures are useful in explaining the performance of a particular heuristic, or as a major contribution in the search for better heuristic procedures. This can be achieved by pointing out the types of problems on which a heuristic performs very well, or those on which its performance is not that good. This may lead to some modifications to improve the overall performance of that heuristic. (77)

In summary, achievement ratios that will be used in this study to evaluate performance of heuristics are:

(1) "BEST" :

It is the percentage of solutions equalling the best heuristic solution =

100 (# of times being the best / total # of problems)

(2) "WORST" :

It is the percentage of solutions equalling the worst heuristic solution =

100 (# of times being the worst / total # of problems)

4.4.2 The Comparative Measures:

As mentioned before evaluatory measures are those that evaluate the goodness of a solution in terms of a standard that can be the optimal solution or the best solution if the optimal is not obtainable.

The Absolute Error :

One of these measures is the absolute error for a solution which is the difference between the solution value and the standard (optimal or the best available one),i.e., it is the deviation from the optimal or the best.

The absolute error is not a valuable measure, since it gives only the distance from the standard, or the size of deviation, and not the relative size of deviation. A deviation value of (2) from a solution value of (10) is completely different from a deviation of (2) from a solution value of (1000). This means that the error ratio should be relative and not absolute.

Let the following terms be defined as :

(S) : the solution value of a heuristic, and

(BS) : the standard, that is the optimal solution value or value of the best solution.

(WS) : the value of the worst solution.

In our case of minimizing an objective function ;

Error = (the solution value) - (the standard)

Error = (S) - (BS)

<u>The Efficiency (E)</u>:

Efficiency (E) is defined as the ratio of the optimal solution or the best solution to the solution value of a heuristic. It should be mentioned here that "Efficiency" as a measure of a heuristic performance is defined differently by many other authors or researchers. They define the efficiency of a heuristic as the length of time required to solve specific problem using this heuristic. The more efficient heuristic is the one that gives the same solution in shorter time.

<u>The Relative Error (R)</u> :

This measure avoids the drawback of the absolute error by comparing the size of the error against the standard.

$$\text{Relative error (R)} = 100 \left(\left| 1 - (BS) / (S) \right| \right)$$

The Error potential ratio (EPR) :

This ratio is one of the comparative measures that can substitute the relative error ratio. It is the ratio of the actual absolute error to the maximum potential absolute error. The maximum potential absolute error is the difference between the worst solution and the standard (the optimal solution or the best one)

$$EPR = 100 \left| (S) - (BS) \right| / \left| (WS) - (BS) \right|$$

Table 1 shows all the measures used to evaluate performance of different heuristics.

4.5 Hypotheses of the Study:

As mentioned previously in the scope of the study, the performance of five scheduling heuristics is examined under different circumstances, including different problem sizes, and different ranges of group setup time.

The impact of the following factors on the measures used to evaluate the solutions are investigated.

1- Impact of using different scheduling heuristics (the 5 mentioned heuristics)

2- Impact of having different ranges for the family or group setup time.

3- Impact of the problem size in general, and in particular

Table 1

The Evaluation Measures
For Heuristics Solutions

Measure	Meaning	Formula
BEST	Percentage of solutions equalling the best heuristic solution	$100 \left(\frac{\#\text{ of times being the best}}{\text{total }\#\text{ of problems}} \right)$
WORST	Percentage of solutions equalling the worst heuristic solution	$100 \left(\frac{\#\text{ of times being the worst}}{\text{total }\#\text{ of problems}} \right)$
E	Efficiency of the solution compared to the best solution	$100 \left(\frac{(BS)}{(S)} \right)$
R	Relative error ratio: size of error compared to the best solution	$100 \left(1 - \frac{(BS)}{(S)} \right)$
EPR	Error potential ratio : ratio of actual absolute error to the maximum potential absolute error.	$100 \left(\frac{(S) - (BS)}{(WS) - (BS)} \right)$

the impact of the number of families, number of machines, and number of jobs).

4- Impact of the interaction effect between the heuristic procedure, group setup time, and problem size.

Based on these investigated factors, the hypotheses of the study are stated as:

H_{01} : The five different heuristics produce solutions whose relative error ratios are not statistically different (regardless of the problem size).

H_{02} : The different categories of problem size produce solutions whose relative error ratios are not statistically different (regardless of the heuristic procedure used).

H_{03} : There is no interaction effect between the problem size and the heuristic procedure used. This means that the problem size does not affect the heuristic performance.

H_{04} : Number of families in the problem has no impact on the solutions relative error (regardless of the heuristic procedure used).

H_{05} : Number of machines in the problem has no impact on the solutions relative error (regardless of the heuristic procedure used).

H_{06} : Number of jobs in the problem has no impact on the solutions relative error (regardless of the heuristic procedure used).

H_{07} : For different ranges of group/ family setup time, each heuristic produces solutions whose relative error ratios are not statistically different.

H_{08} : There is no interaction effect between the problem size and the heuristic procedure used.

4.6 Experiment Design :

The five heuristics were tested extensively over a total of 480 problems of different sizes. A program written in Turbo Pascal (version 3.0) was used to solve the problems on an IBM PS2 (Personal System 2, Model 50).

The problem sizes include small-, medium-, and large-size problems. The sizes range from problems having three families, each family has three jobs, (a total of nine jobs) to be processed on three machines, to problems having 15 families, 20 jobs each, (a total of 300 jobs) to be processed on 20 machines.

The experiment also includes problems at both extremes. Problems with a large number of families and jobs relative to a small number of machines (15 families, 20 jobs each, to be processed on 3 machines), and problems with a large number of machines processing a small number of families and jobs (20 machines, 3 families, each family has 3 jobs).

The job processing times and the family/ group setup times are integers randomly generated from the Uniform distribution. The random number generator that is built in Turbo Pascal has been used to generate the processing and setup times. The processing times were generated over the interval (1,50). The family/ group setup times were generated twice, once over the interval (1,50), we call this range of family setup time S1 since the ratio between the maximum family setup time to the maximum job processing time "S/P" is 50/50 = 1. The family setup time was generated another time over a wider interval (51,250). We call this range S5 because the ratio "S/P" of the maximum setup time to the maximum job processing time is 250/50= 5.

Each problem consists of one of three different numbers of families (3, 10, 15), and each family has one of three different numbers of jobs (3, 10, 20) to be processed on one of three numbers of machines (3, 10, 20).

Different combinations of these numbers were formed to represent three sizes of families, three sizes of machines, and three sizes of jobs, which yields 3x3x3 = 27 different categories of problem size. But the results of the very small problems including only three jobs with three machines were excluded regardless of number of families. It was found for these small problems that all of the five heuristics produce the same solution most of the time, so there is no difference among their performance. Excluding these three categories (3F,3M,3J), (10F,3M,3J), and (15F,3M,3J) from the total of 27 categories leaves only 24 different categories of problem size. Having ten different problems at each level, the total number is 240 problems. Each of these problems was solved once with the first range of family setup time (S1/ 1-50), and solved again with the second range of family setup time (S5/ 51-250), which makes the total number of problems 480. Then each of these 480 problems was solved using each of the five heuristics. Table 2 shows the distribution of problems (sizes, Numbers, and ranges of family setup time.

The procedures used to evaluate the impact of the investigated factors were as follows:

The solution values (total flow time or makespan) were used to calculate the five different ratios used as evaluatory measures for performance.

These five ratios were calculated for each problem (solved by a heuristic under a specific family setup time). Each ratio was then averaged for each set of ten problems of the same size, for each heuristic, and for each range of family setup time. The number of these averages is 24.

Then these ratios were aggregated according to :

1- the number of families in the problem (3, 10, 15) regardless of the number of machines or the number of jobs.

2- the number of machines in the problem (3, 10, 20) regardless of the number of families or the number of jobs.

3- the number of jobs within each family (3, 10, 20) regardless of the number of families or the number of machines.

Finally, one overall average was computed to represent the general performance of a heuristic over the range of the problems tested.

The last step in the analysis was using all these ratios and averages in statistical testing and nonstatistical analysis to answer the research questions or hypotheses.

The hypotheses of the study were tested using parametric statistical tests such as ANOVA (factorial experiments design, and randomized block or matched samples design), also using other nonparametric tests such as the Kruskal-Wallis, and Tukey's tests. The conceptual basics of these tests and the reasons for using them are discussed next.

4.7 Statistical Tests Used:

This section presents the conceptual bases of the statistical tests used and the reasons for using these tests in evaluating the hypotheses.

4.7.1 Parametric Statistical Tests: (3, 78, 113)

The main test used in the analysis is the ANOVA (analysis of variance).(87) ANOVA deals with inferences concerning more than two means. It has been developed to

Table 2

Distribution of the Problems
(Sizes, Numbers, and Ranges of Group Setup Time)

Problem Size			S1	S5
F	M	J	(1-50)	(51-250)
3	3	3	-	-
3	3	10	10	10
3	3	20	10	10
3	10	3	10	10
3	10	10	10	10
3	10	20	10	10
3	20	3	10	10
3	20	10	10	10
3	20	20	10	10
10	3	3	-	-
10	3	10	10	10
10	3	20	10	10
10	10	3	10	10
10	10	10	10	10
10	10	20	10	10
10	20	3	10	10
10	20	10	10	10
10	20	20	10	10
15	3	3	-	-
15	3	10	10	10
15	3	20	10	10
15	10	3	10	10
15	10	10	10	10
15	10	20	10	10
15	20	3	10	10
15	20	10	10	10
15	20	20	10	10
Total			240	240

analyze the experimental data to determine the effects of different treatments.

Although ANOVA assumes that samples are coming from normally distributed populations, and the variances of populations are equal, in general ANOVA is not overly sensitive to these assumptions. (78)

There are several experimental designs that can be employed with ANOVA to lead to conclusions about the difference among the means of several populations or treatments. The most comprehensive of them (for this research's purpose) to examine the effects of two attributes on means are the factorial experiments , and the matched samples designs.

4.7.1.1 ANOVA (Factorial Experiments Design) :(3)

The term "factorial" is used because the experimental conditions include all possible combinations of the factors involved. For example, in this study we have for factor A (problem size) 24 treatments, and for factor B (heuristic procedure used) five treatments. This yields a total of 24X5 = 120 treatment combinations. Since we have 10 replications for each combination, the total number of data items available for analysis using this experimental design should be 120X10 = 1200 items.

The Factorial Experiments design is used in this research to test the following factors:

- Heuristic, problem size, and their interaction.

- Family/ group setup time, problem size, and their interaction (for each heuristic).

This test uses all of the 240 items of solutions relative error for all the problems.

4.7.1.2 ANOVA (The Matched Samples or the Randomized Block Design): (3, 78)

This design aims at controlling some of the extraneous sources of variation. It calls for a single sample to be used in the different treatments. Since both treatments are tested under similar conditions, this design often leads to a smaller error than the independent sample design.

The matched samples (or the randomized block design) is used in this study to test the impact of the following factors:

1- Number of families (in the problem), and the heuristic procedure used.

2- Number of machines (in the problem), and the heuristic procedure used.

3- Number of jobs within each family (in the problem), and the heuristic procedure used.

The test is performed using the averages of the relative error ratio.

4.7.2 Nonparametric Statistical Tests:

The two nonparametric tests used in this study are the Kruskal-Wallis test, and Tukey's test.

4.7.2.1 The Kruskal-Wallis Test:(3, 113)

Although ANOVA (as mentioned before) is not that sensitive to the assumptions it requires, another nonparametric test is used to confirm the results obtained using ANOVA. The Kruskal-Wallis test also tests for the equality of more than two population means. The assumptions needed for the Kruskal-Wallis test are less restrictive than those for ANOVA. Kruskal-Wallis only requires all population distributions to have the same shape. This test is based on ranking the pooled observations, and it follows approximately a chi-square distribution.

In this study the Kruskal-Wallis test is used to investigate the following factors:

1- the differences among the heuristics performance measured by the relative error ratio R.

2- the impact of the number of families, number of machines, and number of jobs within each family on the solutions relative error.

4.7.2.2 Tukey's Honest Significant Difference Test:

Whenever the ANOVA or the Kruskal-Wallis test rejects the null hypothesis of equal means, we may want to consider testing for a difference between the individual treatment means.

For this purpose, a "t" test can be used, but it cannot be used in cases of multiple pair-wise comparisons. So another nonparametric test (Tukey's) is used to trace sources of differences among pairs of treatment means.

Tukey's test is used to compare between all possible pairs of population for equal means. It identifies the pairs of means that are significantly different from the others.

Summary:

This chapter, in summary, covered the definition of the problem and its assumptions, the procedures of the evaluated heuristics and the evaluation measures used. We also discussed the hypotheses of the study, the experiment design, and the statistical tests used in testing the hypotheses.

CHAPTER FIVE

RESULTS OF THE EVALUATION TESTS

This chapter reports the results of the statistical tests and the nonstatistical analysis. The conclusions of this analysis will answer the research questions that were stated as hypotheses.

The results of the evaluation are presented in two parts. The first part covers the hypotheses testing results, and the second part covers the nonstatistical analysis for the other evaluation measures that were not used in the statistical testing.

5.1 Statistical Analysis Results: (*)

The following are the results of different statistical tests used to evaluate the effects of the previously mentioned factors. The results are grouped according to the hypotheses, and supporting tables of the test details are in Appendix (B).

5.1.1 Effect of the Heuristic Procedure Used: (Hyp. #1)

Table 3 shows a summary of the results of statistical tests used to evaluate the first hypothesis (differences among the performance of the heuristics selected).

(*) All the statistical tests use the relative error ratio.

Table 3

Statistical Analysis Results for Hypothesis 1
(Differences Among Heuristics)

Statistical test	Design	S	Result
ANOVA	Factorial Exp.	S1	Reject **
ANOVA	Factorial Exp.	S5	Reject **
Kruskal-Wallis	-	S1	Reject **
Kruskal-Wallis	-	S5	Reject **

** : 1% significance level

The relative error ratios of S1, and S5 were used individually to test this hypothesis. The null hypothesis of equal means for all heuristics was rejected using the ANOVA test (factorial experiments design). This means that there were signigicant differences among the performance of heuristics. These results were confirmed at a 1% level of significance.

When the same hypothesis was tested again using the Kruskal-Wallis test, the same conclusions were reached. So whatever the statistical test used, the results were the same; there were statistically significant differences among performance of heuristics at both 5%, and 1% significance levels.

Since we rejected the null hypothesis of equal means for all heuristics, one further step was required to find out the sources of these differences among heuristics' performance.

Tukey's test was used, as a substitute for the "t" test, to test for equality of all possible pairs of five heuristics. Using the 24 averages of the relative error ratio R(S1), Tukey's test was performed for all ten pair-wise comparisons of the five heuristics, to identify the pairs of heuristic means that were significantly different.

As shown in table 4, the null hypothesis was rejected except for the comparisons between the first three heuristics, NEH, MOD1, and MOD2. This implies that there

Table 4

Pair-wise Comparisons
Between Heuristics
(Using Tukey's Test / S1 Results)

Comparison	Result
NEH & MOD1	Accept **
NEH & MOD2	Accept **
MOD1 & MOD2	Accept **
NEH & SOI	Reject **
MOD1 & SOI	Reject **
MOD2 & SOI	Reject **
NEH & CDS	Reject **
MOD1 & CDS	Reject **
MOD2 & CDS	Reject **
SOI & CDS	Reject **

** : 1% significance level

were no significant differences between these three heuristics at 1% significance level.

5.1.2 Effect of Problem Size: (Hyp. #2)

Problem size is defined to include all different categories of problem size, composed of different combinations of small-, medium-, and large-number of families, machines, and jobs within each family. There are 24 categories of problem size.

The relative error ratio was used in performing the ANOVA test (factorial experiments design). The null hypothesis was rejected for all the tests, and for both ranges of family/ group setup time, S1 and S5, as shown in table 5. This means that there were significant differences among the solutions relative error of the different problem size categories.

The same hypothesis was tested again using the Kruskal-Wallis test, and the results of the ANOVA test were confirmed by rejecting the null hypothesis. This implies significant effect for the problem size categories on the solutions relative error ratio R.

5.1.3 Interaction Effect of Heuristic and Problem Size: (Hyp. #3)

The ANOVA test (Factorial Experiments Design) was used to measure the interaction effect of the heuristic procedure

Table 5

Statistical Analysis Results for Hypothesis 2
(Impact of Problem Size)

Statistical test	Design	S	Result
ANOVA	Factorial Exp.	S1	Reject **
ANOVA	Factorial Exp.	S5	Reject **
Kruskal-Wallis	-	S1	Reject **
Kruskal-Wallis	-	S5	Reject **

** : 1% significance level

used and the problem size categories (combinations of small-, medium-, and large-number of families, machines, and jobs). As shown in table 6, the results of the ANOVA test reject the null hypothesis at 1% significance level. This implies that, there was a significant impact for the problem size on the relative performance of the heuristics. The same conclusion was achieved using the results of both S1 and S5.

Regarding the components of the problem size, represented by the number of families, machines, and jobs, the ANOVA test (Randomized Block) was performed to investigate the impact of each of them on the differences among heuristics performance. The three aggregated averages of the relative error ratio, grouped according to the number of families, machines, and jobs were used in the test to represent different problem sizes. As shown in table 7, the results of the ANOVA test show significant differences among the heuristics performance, at 1% level of significance.

Table 7 shows also that the results of applying the Kruskal-Wallis test confirm those of the ANOVA's, regarding the number of families, and number of jobs, but only at a 5% level of significance. The results for the number of machines are different, implying nonexistence of significant differences among the heuristics, when the relative error ratios were grouped based on the number of machines.

Table 6

Statistical Analysis Results for Hypothesis 3

(Interaction Effect of the Heuristic and Problrm Size)

Statistical test	Design	S	Result
ANOVA	Factorial Exp.	S1	Reject **
ANOVA	Factorial Exp.	S5	Reject **

** : 1% significance level

Table 7

Differences Among Heuristics Based on

# of Families, Machines, and Jobs

Factor	Statistical test	Design	S	Result
# of Families	ANOVA	Randomized Block	S1	Reject **
	Kruskal-Wallis	-	S1	Reject *
# of Machines	ANOVA	Randomized Block	S1	Accept **
	Kruskal-Wallis	-	S1	Accept **
# of Jobs	ANOVA	Randomized Block	S1	Reject **
	Kruskal-Wallis	-	S1	Reject *

* : 5% significance level

** : 1% significance level

5.1.4 Effect of Number of Families: (Hyp. #4)

Two different tests were performed to investigate the effect of number of families on the solutions relative error. Both tests used the 24 averages of the relative error ratio of the five heuristics, grouped according to the number of families (regardless of the number of machines or jobs). The three groupings were chosen to represent different categories of problem size, 3 families as a small-family size, 10 as a medium-family size, and 15 as a large-family size.

The first test used was the ANOVA (randomized block design, or matched samples). As a result of this test, the null hypothesis could not be rejected, which means that there was no significant effect for the number of families on the solutions relative error (as a performance measure).

The second test used for this hypothesis was the Kruskal-Wallis test. The result was found to confirm the previous conclusion, no significant effect for the number of families at a 1% level of significance, as shown in table 8.

5.1.5 Effect of Number of Machines (Hyp. #5):

This hypothesis was also tested on two different tests, using the 24 averages of the relative error ratio, grouped into three levels according to the number of machines in the problem. These groupings represent three different categories of problem size (small, medium, and large).

Table 6

Statistical Analysis Results for Hypotheses 4,5,6

Effects of # of Families, Machines, and Jobs) (S1)

Hyp.#	Factor	Test	Design	Result
4	# of Families	ANOVA	Randomized Block	Accept **
		Kruskal-Wallis	-	Accept **
5	# of Machines	ANOVA	Randomized Block	Accept **
		Kruskal-Wallis	-	Reject *
6	# of Jobs	ANOVA	Randomized Block	Accept **
		Kruskal-Wallis	-	Accept **

** : 1% significance level

* : 5% significance level

Using the ANOVA test, the results could not reject the null hpothesis of having no significant effect for number of machines.

The second test for this hypothesis was the Kruskal-Wallis, which gave a different result from that of the ANOVA's. The null hypothesis was rejected at a 5% significance level, but not at a 1% level, which means that the number of machines in the problem has a significant effect on the solutions relative error. (*)

This apparent contradiction may due to using different statistical tests, each having different assumptions. It has been mentioned before that the Kruskal-Wallis test by its nature is less precise than the ANOVA test.

5.1.6 Effect of Number of Jobs (Hyp. #6):

As shown in table 8, the same result was reached using two different tests: the ANOVA test and the Kruskal-Wallis test. The results of both tests indicate that the null hypothesis of having equal population means could not be rejected at a 1% significance level.

----

* The same test confirms the ANOVA's results at 1% level of significance by accepting the same hypothesis. See table 8.

5.1.7 Effect of Group Setup Time (Hyp. #7) :

The ANOVA test (factorial experiments) was used to test the effect of group setup time on the relative performance of each heuristic under different ranges of family/ group setup time. The test was repeated five times, once for each of the five heuristics to test the impact of two different ranges of family/ group setup time, S1(1-50), and S5(51-250). The results of the test could not reject the null hypothesis for four of the heuristics (NEH, MOD1, MOD2, and CDS), and rejected it only for SOI. This means that changing family/ group setup time from S1 to S5 has a significant impact on SOI performance, but makes no significant difference for the other four heuristics (NEH, MOD1, MOD2, and CDS), at a 1% significance level. This is shown in table 9.

5.1.8 Interactior E'fect of Problem Size and Group Setup Time: (Hyp. #8)

This hypothesis was tested using the ANOVA test (factorial experiments design), for each of the five heuristics. A shown in table 10, the results could not reject the null hypothesis, which implies nonexistence of an interaction impact for problem size and group setup time.

Table 9

Statistical Analysis Results for Hypothesis 7
(Impact of Group Setup Time, S1&S5)
(Using the ANOVA Test/ Factorial Experiments)

Heuristic	Result
NEH	Accept **
MOD1	Accept **
MOD2	Accept **
SOI	Reject **
CDS	Accept **

** : 1% significance level

Table 10

Statistical Analysis Results for Hypothesis 8
(Interaction Effect of Problem Size and Group/ Family Setup Time)

Test	Design	S	Heuristic	Result
ANOVA	Factorial Experiments	S1/S5	NEH	Accept **
ANOVA	Factorial Experiments	S1/S5	MOD1	Accept **
ANOVA	Factorial Experiments	S1/S5	MOD2	Accept **
ANOVA	Factorial Experiments	S1/S5	SOI	Accept **
ANOVA	Factorial Experiments	S1/S5	CDS	Accept **

** : 1% significance level

5.2 Results for the Other Measures:

The following is a presentation of the results for the other performance evaluation ratios which were not used in the statistical testing.

5.2.1 General Performance of Heuristics in S1, and S5:

The figures used in this analysis are the overall averages of the 240 problems as shown in table 11. The results in the table show that NEH occupies the first place regarding all the ratios for S1, and S5, except for the percentage of worst solutions "WORST" (S1, and S5), and in the error potential ratio "EPR" (S5). For the percentage of worst solutions "WORST", both of MOD1, and MOD2 share the first place in S5, and S1 .

Comparing performance of each heuristic under the two different ranges of group setup time, all the ratios of NEH for S5 are worse than those for S1, although statistical tests using the relative error ratio showed that these differences were not significant. But this decline in performance can indicate the trend of NEH's performance. Perhaps if the range of family setup time increases, these differences may become more significant.

MOD1 performance improves for S5 compared to S1 in all of the ratios except for the percentage of worst solutions "WORST". MOD1's "WORST" ratio has a slight increase in S5, (only 0.4%) in spite of improving its rank order from being the third in S1 to be the first in S5.

Table 11

Overall Averages of Heuristics Performance

Measure	NEH		MOD1		MOD2		SOI		CDS	
	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5
BEST %	45.83	43.33	30.42	32.22	35.42	37.50	7.92	10.42	7.50	9.14
WORST %	4.17	7.50	4.58	5.00	2.50	5.83	47.50	41.67	42.50	41.67
Efficiency	.9938	.9930	.9885	.9900	.9906	.9910	.9690	.9750	.9720	.9750
Relative Error	.6245	.7320	1.158	.9530	.9430	.9320	3.145	2.491	2.790	2.4520
Error Potential	19.67	23.47	27.91	23.12	22.61	27.87	71.50	65.04	70.78	68.80

For MOD2, we can not generalize a trend for all the ratios. MOD2 has an improved ratio for the percentage of best solutions "BEST", and also for the relative error ratio "R". It has exactly the same efficiency ratio "E", and a worse ratio for the percentage of worst solutions "WORST", and for the error potential ratio "EPR".

SOI has the same situation as MOD1's. All SOI's ratios are greatly improved for S5 compared to S1. This improvement was statistically significant when the relative error ratio was used. CDS also has better results (regarding all the ratios) for S5 than for S1. But this improvement in performance of the heuristic was not statistically significant, when the relative error ratio was used.

In general the comparison between MOD1 and MOD2 indicates that MOD2 performs better than MOD1 for all of the ratios of S1, and S5 except for the percentage of worst solutions "WORST"/S5, and for the error potential ratio "EPR"/S5, where MOD1 is the best of all the five heuristics.

An interesting result of this study is the impact of group setup time on the performance of the heuristics. SOI was the only heuristic out of the five that improved its performance with increasing group setup time. This group setup time effect might be behind the improved results of SOI performance compared to CDS's in this study, which differ from the previous comparative studies results. Also, an important conclusion is that of the improvement on the performance of MOD1, SOI, and CDS in S5 than in S1 (whatever the ratio used to evaluate that performance).

5.2.2 Comparing Performance of Heuristics in S1:

The tables in Appendix (A) show the performance of all heuristics evaluated by the five evaluation ratios. Each figure of these averages in the tables represents an average of the results of ten replications for problems of the same size.

From the mentioned tables, we find that performance of the first three heuristics, NEH, MOD1, and MOD2 to be very close. The other two heuristics (SOI, and CDS) will be compared with each other, since there is a large difference between their performance on one side, and the first three heuristics performance on the other side.

Regarding the percentage of best solutions "BEST", table 1/ Appendix (A) shows that the number of times, MOD1 achieves a ratio that is at least as good as NEH's is 3 out 24 cases or 12.25%. MOD2 achieves performance as good or better than NEH in 33.33%.

But for the percentage of worst solutions "WORST", both MOD1, and MOD2 perform better than the other heuristics including NEH. The number of times that MOD1 has equivalent or better performance compared to NEH is 79.17%, and 87.5% for MOD2. MOD2 has an average ratio that is about 67% better than NEH's.

It has been found from using the efficiency ratio "E" that, this ratio does not magnify the differences among the huristics performance as the other ratios do. In table 3/ Appendix (A), each of the three heuristics NEH, MOD1, and MOD2 has about 99% efficiency ratio. The differences in performance are very small where they range only from (.3%) to (.5%) between NEH and the other two heuristics.

Table 4/ Appendix (A) shows the relative error ratio (R) which was used in the statistical tests. We see that MOD2 gives results that are equivalent to or better than NEH's in about 38%, and MOD1 achieves a ratio that is as good or better than NEH's in about 17% of the time. It should be mentioned here for SOI, that although it is the least efficient or the worst of the five heuristics, it achieves the best relative error ratio in one of the problem size categories (10F,3M,10J), and the second best in another category (15F,3M,20J).

Table 5/ Appendix (A) shows MOD1's error potential ratio "EPR" to be as good as or better than NEH's in 25% of the time. MOD2 achieves this level of performance in 42% of the time. The best performance by MOD2 is observed clustered in the category of 15-family problems. The absolute difference between NEH and MOD2 in the overall average of "EPR" is only 2.4%. The same high level of performance by SOI is observed in the category of (15F,3M,20J), where it achieves the second best, in spite of having the worst overall performance of all five heuristics.

An interesting conclusion from this study is the inconsistent performance of NEH. Although NEH occupies the first place for most of the ratios, it does not have the most efficient ratio for percentage of worst solutions "WORST". This can indicate that when NEH misses giving the best solution, it gives a poor one, which is often the worst. When this performance is compared with the performance of the other two heuristics MOD1, and MOD2, we find that they perform more consistently regarding the percentage of worst solutions "WORST".

5.2.3 Comparing Performance of Heuristics in S5 :

In the percentage of best solutions "BEST", (shown in table 6/ Appendix (A)), we find that the first place is occupied by NEH, then MOD2, with a difference of about 5.83%. MOD1's performance is equivalent to or better than NEH's in 50% of the 24 averages. SOI is the second worst heuristic after CDS, but at the same time it achieves some high-performance level (2nd place) in the following categories of problem size (3F,3M,20J), (10F,3M,20J), and in (15F,20M,20J). Achieving the second best in the category of the very large problems (15F,20M,20J) is a great achievement for SOI. In spite of that, it still has a low average performance, measured by percentage of best solutions.

The results of the overall average of the percentage of worst solutions "WORST" in table 7/ Appendix (A) show that MOD1 occupies the first place, followed by MOD2, then NEH in the third place. The difference between MOD1, and NEH is 2.5%, which represents 50% improvement for MOD1. The same for MOD2 and NEH, the difference is .83%, which represents 16.6% . In this ratio, both MOD1 and MOD2 perform as good as or better than NEH in about 83.33% of the 24 averages.

As mentioned before, regarding reliability on the efficiency ratio "E", all three heuristics NEH, MOD1, and MOD2 almost have the same efficiency ratio with very small differences that range from 0.2% to 0.3% between NEH's and both of MOD1's, and MOD2's. In spite of sharing the fourth and the last place with CDS, SOI has the highest efficiency ratio in (10F,3M,20J), and the second best in (15F,3M,20J). CDS also for the first time has the second best ratio in (10F,3M,10J).

Regarding the relative error ratio shown in table 9/ Appendix (A), the results indicate that the difference between NEH's and both MOD1's, and MOD2's is only 0.2. Both MOD1, and MOD2 achieve ratios that are as good as or better than NEH's in 33.33% of the 24 averages. Again SOI has the lowest relative error ratio in (10F,3M,20J), but still occupies the last place regarding the overall average of the relative error ratio.

Table 10/ Appendix (A) shows the averages of the error potential ratio "EPR". MOD1 has the best performance, and at the same time has a general performance that is equivalent to or better than NEH's in about 46% of the times.

5.2.4 Comparing Performance of MOD1, and MOD2:

In S1, generally speaking, MOD2 performs as good as or better than MOD1 whatever the performance measure used. But when the relative error ratio was used in the statistical tests, the results showed no significant difference between their performance.

For the percentage of best solutions "BEST", MOD2 has a better overall average (35.42%), compared to MOD1's (30.42%). MOD2 gives a better ratio in 54.2% of the 24 averages, while MOD1 outperforms it in only 29.2%.

Both heuristics achieve almost the same level of performance, when it is measured by the percentage of worst solutions "WORST". They achieve exactly the same ratio in about 54.17% of the 24 averages, whereas MOD2 still gives better results in 29.17% of the cases, compared to 16.67% for MOD1. In the overall average for this ratio, MOD2 has a better ratio (2.5%) compared to MOD1's (4.58%) with a difference of 2%, but this difference represents 83.2% improvement for MOD2.

MOD2 is also a better performer than MOD1, when the relative error ratio is used as an evaluatory measure. But when this ratio was used for testing the hypothesis, the statistical tests concluded that the difference between their performance was not significant.

From the results of the error potential ratio "EPR", we find that MOD2 is still doing better than MOD1 in the overall average, with a difference of 5.3% that represents about 28% improvement. MOD2 has a better performance in 66.67% of the cases, compared to 33.33% for MOD1. MOD2 does exceptionally well in the category of (large number of families) 15-family problems. The best performance of MOD1, is in the problems having small number of jobs, and those having medium number of families.

Regarding their performance in S5, the results indicate that the general performance of MOD2 is also better than MOD1's. For the percentage of best solutions "BEST", the difference between their overall averages is 5.28%, which represents 16.39% improvement for MOD1. In about 41.67% of the cases, both heuristics have exactly the same ratio, but MOD2 performs better than MOD1 in 37.5% of the cases compared to 20.83% for MOD1.

The same situation occurs for the percentage of worst solutions "WORST", with the same difference between their overall averages, about 16.6% improvement for MOD2. But the percentage of time both heuristics have the same ratio is higher, about 58.33%, with exactly the same percentage of getting a better performance than one another (20.83%), for each of them.

The difference between the relative error ratio for both heuristics is very small, about 0.021.

In the error potential ratio EPR, MOD1 has a better ratio than MOD2, with a difference of 4.75% in the overall average, which represents a 20.54% improvement for MOD1. In most of the cases (70.83%), MOD1 achieves better ratios than MOD2.

5.2.5 Effect of the Number of Families, Machines and Jobs on Heuristics Performance:

Generally speaking, little can be concluded about trends of the heuristics performance from only three points, but the following indications can be derived from the following tables (12-16).

Effect of the Number of Families: As shown in the tables, NEH performs better than the other heuristics for most of the performance measures. Although NEH has inconsistent performance in terms of percentage of best solutions "BEST", the other ratios including percentage of worst solutions "WORST", relative error ratio "R", and error potential ratio "EPR" show less response to the number of families compared to the response for other heuristics.

MOD2 is the next best performer for most of the ratios, and leads all heuristics in "WORST". Its performance tends to improve with an increasing number of families. This may indicate a better performance than NEH in the problems having a large number of families. Its performance measured by the percentage of worst solutions "WORST" is very consistent, where it has exactly the same ratio, regardless of the number of families. Its performance in the category of 15-family problems compared to NEH is better, which may indicate an expected high performance in problems with larger number of families.

SOI also has a tendency to improve its performance with increasing the number of families. It has a very low performance in the problems having a small number of families . In contrast to that, it has a better performance in the category of 15-family problems compared to CDS.

<u>Effect of the Number of Machines</u>: Regarding the effect of the number of machines, NEH's performance declines when the number of machines increases, but in a consistent pattern. It has the best ratios, whatever the number of machines is, except in the percentage of worst solutions "WORST", and the small-problem category of 3 machines .

MOD2 performs very well in the small-problem category of three machines, where it usually has a better or equivalent ratio compared to NEH. SOI also provides better results for the small-size category, but its performance worsens with an increasing number of machines. In contrast to SOI, CDS has an improving performance when the number of machines increases, because in general, CDS's performance depends on the number of machines. In its procedure it develops (M-1) subproblems , and offers (M-1) solutions to choose the best one from. So, the more subproblems it develops, the more solutions it provides to choose from, and the better the solution will be.

Effect of the Number of Jobs: From tables (12-16), we can see that NEH's performance improves when the number of jobs increases. It has almost the same level of performance for the medium- and large-size categories. For the small-size problems, all three heuristics NEH, MOD1, and MOD2 have exactly the same ratio. MOD2 is the best performer whatever the number of jobs is, for the percentage of worst solutions "WORST".

It is clear from the tables also that SOI has a very low performance for the small problems, but improves as the number of jobs increases.

Table 12

Effects of Number of Families, Number of Machines, and Number of Jobs on Heuristics Performance Measured by "BEST" Ratio (S1, S5)

PROB. SIZE	BEST (%)											
	NEH		MOD1		MOD2		SOI		CDS		AVG.	
	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5
FAMILIES												
3 F	58.75	47.50	30.00	41.25	36.25	48.75	6.25	11.25	11.25	10.00	28.50	34.25
10 F	41.25	45.00	33.75	28.75	28.75	28.75	8.75	8.75	10.00	6.25	24.50	24.25
15 F	37.50	37.50	27.50	35.00	41.25	35.00	8.75	11.25	1.25	11.25	23.25	26.00
MACHINES												
3 M	55.00	48.33	45.00	55.00	48.33	55.00	20.00	28.33	11.67	21.76	36.00	41.67
10 M	45.56	46.67	26.67	30.00	23.33	34.44	5.56	1.11	7.78	3.33	21.78	23.11
20 M	40.00	36.67	24.44	30.00	38.89	28.89	2.22	7.78	4.44	6.67	22.00	22.00
JOBS												
3 J	31.67	40.00	41.67	41.67	31.67	35.00	8.33	8.33	15.00	10.00	25.67	27.00
10 J	52.22	43.33	27.78	55.56	33.33	43.33	6.67	7.78	7.78	7.78	25.25	23.11
20 J	48.89	45.56	25.56	33.33	40.00	33.33	8.89	14.44	2.22	10.00	25.11	27.33
AVG.	45.83	43.33	30.42	32.22	35.42	37.50	7.92	10.42	7.50	9.14	25.42	27.33

Table 13

Effects of Number of Families, Number of Machines, and Number of Jobs on Heuristics Performance Measured by "WORST" Ratio (S1, S5)

PROB. SIZE	WORST (%)											
	NEH		MOD1		MOD2		SOI		CDS		AVG.	
	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5
FAMILIES												
3 F	2.50	10.00	2.50	5.00	2.50	5.00	66.25	47.50	27.50	37.50	20.25	21.00
10 F	5.00	5.00	5.00	6.25	2.50	7.50	38.75	42.50	47.50	40.00	19.75	20.25
15 F	5.00	7.50	6.25	3.75	2.50	5.00	37.50	35.00	52.50	47.50	20.75	19.75
MACHINES												
3 M	10.00	16.67	8.33	5.00	1.67	8.33	36.67	31.67	43.33	38.33	20.00	20.00
10 M	3.33	4.44	5.56	3.33	2.22	2.22	47.78	42.22	43.33	50.00	20.44	20.44
20 M	1.11	4.44	1.11	6.67	3.33	7.78	54.44	47.78	41.11	35.56	20.22	20.44
JOBS												
3 J	6.67	8.33	6.67	8.33	5.00	8.33	32.22	26.67	25.56	28.89	15.22	21.67
10 J	3.33	7.78	3.33	4.44	1.11	6.67	55.56	53.33	38.89	30.00	20.44	29.33
20 J	3.33	6.67	4.44	3.33	2.22	3.33	38.89	31.11	48.89	52.22	19.55	19.33
AVG.	4.17	7.50	4.58	5.00	2.50	5.83	47.50	41.67	42.50	41.67	20.25	

Table 14

Effects of Number of Families, Number of Machines, and Number of Jobs on Heuristics Performance Measured by "Efficiency" (S1, S5)

PROB. SIZE	EFFICIENCY									
	NEH		MOD1		MOD2		SOI		CDS	
	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5
FAMILIES										
3 F	0.9933	0.9920	0.9874	0.9910	0.9908	0.9940	0.9560	0.9690	0.9678	0.9710
10 F	0.9944	0.9940	0.9875	0.9900	0.9878	0.9890	0.9730	0.9770	0.9739	0.9770
15 F	0.9938	0.9930	0.9906	0.9910	0.9933	0.9890	0.9768	0.9800	0.9749	0.9790
MACHINES										
3 M	0.9962	0.9950	0.9955	0.9960	0.9967	0.9950	0.9902	0.9930	0.9908	0.9920
10 M	0.9928	0.9930	0.9844	0.9880	0.9874	0.9900	0.9616	0.9700	0.9642	0.9690
20 M	0.9932	0.9910	0.9879	0.9890	0.9897	0.9890	0.9612	0.9680	0.9677	0.9710
JOBS										
3 J	0.9882	0.9890	0.9863	0.9880	0.9860	0.9880	0.9663	0.9720	0.9692	0.9720
10 J	0.9948	0.9920	0.9868	0.9900	0.9896	0.9900	0.9642	0.9720	0.9711	0.9720
20 J	0.9966	0.9960	0.9917	0.9930	0.9947	0.9930	0.9744	0.9800	0.9752	0.9800
AVG.	0.9938	0.9930	0.9885	0.9900	0.9906	0 9910	0.9690	0.9750	0.9720	0.9750

Table 15

Effect of Number of Families, Number of Machines, and Number of Jobs on Heuristics Performance Measured by "Relative Error" Ratio

(S1, S5)

PROB. SIZE	RELATIVE ERROR											
	NEH		MOD1		MOD2		SOI		CDS		AVG.	
	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5
FAMILIES												
3 F	0.6666	0.8120	1.2650	0.9360	0.9200	0.6570	4.40[illegible]	3.1240	3.2180	2.8810	2.0950	1.6830
10 F	0.5643	0.6390	1.2480	1.0270	[illegible]	1.0650	2.6898	2.3490	2.6220	2.3360	1.6690	1.4830
15 F	0.6428	0.7450	0.9620	0.8960	0.6873	1.0750	2.3404	1.9990	2.5306	2.1400	1.4330	1.3710
MACHINES												
3 M	0.3637	0.5580	0.4607	0.3830	0.3335	0.4810	0.9818	0.6590	0.9335	0.8190	0.6146	0.5800
10 M	0.7293	0.7220	1.5570	1.1750	1.2602	1.0190	3.8460	2.9490	3.5930	3.0720	2.1971	1.7870
20 M	0.6936	0.8570	1.2239	1.1100	1.0310	1.1460	3.8870	3.2540	3.2250	2.9220	2.0121	1.8580
JOBS												
3 J	1.2120	1.0860	1.3740	1.2260	1.3898	1.2620	3.3695	2.8450	3.0947	2.6600	2.0880	1.8160
10 J	0.5294	0.7930	1.3210	1.0070	1.0524	0.9340	3.5732	2.7460	2.8977	2.3910	1.8747	1.5740
20 J	0.3279	0.4340	0.8512	0.7160	0.5350	0.7110	2.5681	2.0000	2.4801	2.3760	1.3525	1.2470
AVG.	0.6245	0.7320	1.1580	0.9530	0.9430	0.9320	3.1450	2.4910	2.7900	2.4520	1.7320	

Table 16

Effects of Number of Families, Number of Machines, and Number of Jobs on Heuristics Performance Measured by "Error Potential" Ratio (S1, S5)

PROB. SIZE	ERROR POTENTIAL RATIO											
	NEH		MOD1		MOD2		SOI		CDS		AVG.	
	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5	S1	S5
FAMILIES												
3 F	15.703	23.363	25.845	21.111	20.109	18.742	82.596	68.677	62.178	69.103	41.285	40.199
10 F	20.137	21.413	26.547	25.257	28.762	32.338	64.992	65.449	70.218	67.204	42.115	42.332
15 F	23.170	25.628	31.347	22.993	18.955	32.528	66.992	61.000	79.941	70.100	44.081	42.450
MACHINES												
3 M	24.993	31.553	26.495	19.056	20.413	22.192	52.440	43.639	65.553	57.285	37.979	34.745
10 M	18.630	20.568	29.536	23.617	23.652	27.555	76.921	71.434	72.158	76.112	44.179	43.857
20 M	17.161	20.978	27.235	25.332	23.029	31.969	60.649	72.918	72.464	69.170	40.192	44.073
JOBS												
3 J	29.908	26.546	31.132	22.785	30.553	33.335	72.802	66.902	67.363	64.794	46.352	42.872
10 J	16.914	24.120	27.583	23.875	22.217	26.134	75.983	70.385	65.693	64.392	41.678	41.781
20 J	15.600	20.764	26.970	22.589	17.704	25.961	66.154	58.458	78.143	75.884	40.740	40.731
	19.670	23.468	27.910	23.120	22.609	27.869	71.499	65.042	70.779	68.802	42.494	

Summary:

In chapter five, we presented the results of the evaluation, including the hypotheses testing results, and the results for the other performance measures that were not used in the statistical analysis. The most important finding is that MOD1 and MOD2 performance was as good as or better than NEH performance. This result was confirmed by the results of the statistical tests.

CHAPTER SIX

SUMMARY, CONCLUSIONS, AND SUGGESTIONS FOR FUTURE RESEARCH

6.1 Summary :

The group technology concept has been gaining interest as a potential method of increasing the effectiveness of batch production. The increasing use of small- and medium-batch production in the U.S.A. and around the world has brought new focus to the group technology concept.

Group technology (GT) is an efficient way of reforming job-shop manufacturing (used in small-batch production) and increasing its productivity. Economic advantages similar to those associated with continuous flow-line production can be achieved for small-batch production.

The GT methodology involves studying similarities among parts, grouping those parts having similar manufacturing sequences into families, determining the machines and tools required to process them and arranging these machines in cells to process these families of parts.

The basic idea of GT started with the aim of reducing setup time while sequencing parts on a single machine (GT center). But the concept has been developed to include more complicated forms of layout. The most rationalized form is the GT flow line. Although it offers the most advantages of the system, the GT flow line also requires a higher level of similarity among members of the family. The third form is the GT cell, which is applicable whenever the GT flow line is not possible because of the lower level of similarity among the parts. This form is most commonly used in flexible manufacturing systems.

The advantages of implementing the GT concept have a wide impact on the performance of the manufacturing and nonmanufacturing functions in the company. These advantages include increase in capacity due to reduced setup time, simplification of materials flow, reduction in throughput time, and in work-in-process (inventory), and improvement in production planning and control. Among the other indirect but significant advantages, are the effect GT has on the quality of production, and also on the quality of work environment, in addition to improving the other social and behavioral aspects of human resources management.

But the most promising potential of GT is the role it can play in developing the factory of the future, a concept that is based on integrating all the modern manufacturing- and information-related technology in different areas of design, processing, planning and scheduling production, and in materials handling. In particular, GT can provide the parts data base, that can integrate design and manufacturing information in applying Computer-Aided Design (CAD), and Computer-Aided Manufacturing (CAM) in Computer-Integrated

Manufacturing (CIM), which is the base for facory of the future.

One of the major impacts of GT is on production scheduling, because the scheduling problem is greatly simplified by using GT. The scope of the scheduling problem is reduced from that of scheduling for a large shop to scheduling for a small group of machines in a cell forming a GT flow line.

The obvious problems of the functional (job-shop) layout can be drastically reduced through the GT layout, which allows more control over the flow of parts through the shop (cell).

Proper scheduling is also an integral part of GT, because when better scheduling decisions are combined with reduced setup time, and reduced transportation, this will result in a significant cost reduction. Simplifying the production scheduling problem through GT implementation leads to a possible application of the available mathematical models that have been developed for the conventional flow shop.

The accurate schedule for GT applications can facilitate implementing the just-in-time concept, because it will avoid the need to maintain a large quantity of parts due to their uncertain demand. The accurate schedule can lead to producing the part just before it is needed.

Most of the flow shop problems including three machines or more, are said to be NP-complete, i.e., no simple rule has been offered for determining the optimal schedule. Although heuristics do not necessarily provide optimal solutions, they provide efficient and economical ways to get good solutions. The heuristic approach is used commonly in scheduling problems, because the other optimization techniques such as dynamic programming and branch and bound require prohibitive computation time and memory to keep track of the calculation even for small problems.

The literature review in the area of GT flow line scheduling shows that there is a need in this area for a comparative study to evaluate different solution techniques that can be used in scheduling the families and their jobs.

This study is a computer-based investigation and comparison of five different heuristics to solve the scheduling problem for a GT flow line.

The five evaluated heuristics included three heuristics that proved in different previous comparative studies that they perform better than the other heuristics. These heuristics are NEH, SOI, and CDS. They were developed for the conventional flow shop, but in this study they were adjusted for group production application. The performance of these three heuristics was compared with the performance of two new heuristics MOD1, and MOD2. These two heuristics are original contributions of this study, and they were developed by modifying NEH.

The five heuristics performance was compared in solving the scheduling problem for a static deterministic GT flow line. The objective function to be minimized in solving the scheduling problem is the makespan or the total flow time or the maximum completion time.

The study also investigated the effect of the following factors on the relative performance of the selected heuristics:

1- The effect of problem size.

2- The effect of the number of families, number of machines, and number of jobs.

3- The effect of group setup time.

All these factors were expressed in eight hypotheses that were tested statistically, and analyzed using different evaluation measures.

The five heuristics were tested over a total of 480 problems. A program written in Turbo Pascal was used to solve these problems on an IBM PS2 micro computer (Model 50).

Each of the 480 problems was solved using all five heuristics (NEH, MOD1, MOD2, SOI, and CDS). Two ranges of group setup time were used S1 (1-50), and S5 (51-250).

The problem sizes included different combinations, of small-, medium-, and large-number of families, machines, and jobs. The smallest problem has three families, each family has three jobs to be processed on three machines, and the largest problem has fifteen families, each family has twenty jobs to be processed on twenty machines.

The solution values of all the problems were used to calculate five different evaluation measures. These measures/ ratios were used in the analysis, and in tesing the hypotheses.

6.2 Conclusions :

Eight hypotheses were tested in the study to investigate different factors including the differences among heuristics performance (hyp.1), effect of problem size (hyp.2), interaction effect of heuristic and problem size (hyp.3), effect of number of families (hyp.4), number of machines (hyp.5), number of jobs (hyp.6), effect of group setup time (hyp.7), and interaction effect between setup time and problem size (hyp.8). The following are the most important results of the study.

From the results of the five heuristics, the statistical tests showed significant differences among their performance.

The pair-wise comparisons between heuristics, indicated that performance of the first three heuristics (NEH, MOD1, and MOD2) was almost equivalent. There were no significant differences between their performance when Tukey's test was used.

It was also found that the different categories of problem size had a statistically significant effect on the solutions relative error.

Another important conclusion was that the range of group setup used had no significant effect statistically on the relative performance of heuristics. It was also found that there was no significant interaction effect between the group setup time and the categories of problem size.

Regarding the effect of number of families, the statistical tests showed no significant impact. At the same time there were no effects for number of machines, and number of jobs on the solutions relative error.

When the relative error ratio was grouped according to the number of families and to the number of jobs, the heuristics proved to have significantly different performance. Similarly, there were no significant differences among the heuristics performance when the relative error ratio was grouped according to the number of machines.

Another conclusion from the statistical tests showed that there was a significant interaction effect for the problem size categories, and the selected heuristics, which means that the relative performance of a heuristic was affected by the problem size.

MOD1, and MOD2 showed a performance that was as good as or better than NEH's. MOD1, and MOD2 had the best and the second best ratio for the percentage of worst solutions "WORST", compared to NEH's, which might indicate a more consistent performance.

The same high performance was also shown by MOD1 and MOD2 for the error potential ratio "EPR". This relationship between "WORST" and "EPR" was expected, since the later ratio considers in its formula the worst solution value.

MOD2's performance showed an improvement with an increasing number of families. It provided the best performance in the fifteen-family problems. This might lead to expecting a better performance on larger problems compared to the other heuristics.

Comparing MOD1 and MOD2 results, MOD2 performed either better than or equivalent to MOD1 for all the ratios. Although at the same time, when the relative error ratio was used in the statistical analysis, the results showed no significant difference between their performance.

6.3 Suggestions for Future Research:

There are many topics that can be suggested for future research. These include some of the factors or the assumptions that have not been covered in this research.

One of these topics is extending the scope of the current study to compare the performance of the three heuristics that have proved in this study that they perform equivalently, and much better than the others. These heuristics are NEH, MOD1, and MOD2. Their performance should be tested on larger-size problems than those used in this study, with more concentration on the number of jobs within each family. This is because the main goal is to solve the scheduling problem for a GT flow line that completes processing a set of similar families. So, it is not expected to have a GT flow line that has a very large number of machines, or to find one GT flow line that processes a very large number of similar families.

However, little is really known about the true dimensions of such scheduling problems in practice, now or in the future. So, an empirical study can investigate the scheduling techniques used in the companies that have implemented the GT concept. Based on a survey, or various cases study, the data about their applications can be gathered and analyzed to extract their problems and come up with the proper problem dimensions to focus future research on testing them.

A theoretical computer-based study can compare the performance of the three heuristics NEH, MOD1, and MOD2 under different assumptions, such as having a dynamic stochastic GT flow line.

Another study can examine the case of switching the order of a heuristic procedure by starting with sequencing the jobs within the families first before sequencing the families themselves, and see whether this change affects the result (solution) or not. This modification can be improved by repeating the same process. We can start the procedure by sequencing the families first, then the jobs within the families. The last job sequence within the families can be used to sequence the families again, then the jobs within the families, then the families, ..and so on. We can switch back and forth until no further change occurs.

Another suggested topic can start with one of the best three heuristics (NEH, MOD1, or MOD2) as a base solution and apply a fine tuning routine to improve its performance. Any of the neighborhood search techniques can be used , similar to those used by Dannenbring.

Another suggestion is to use any of the three heuristics NEH, MOD1, or MOD2, as a base for applying the procedures of the branch and bound technique. Starting the B&B procedures with a-good-enough solution may save alot of the required time by the B&B to solve a flow-shop scheduling problem. This may lead to improving the applicability of B&B to real-life problems.

Another study may test the performance of the best three heuristics in scheduling for a cell in a flexible manufacturing system.

It was also found from the literature review that nothing has been mentioned in detail about GT applications in Japan, and the different planning and scheduling techniques used there. So, the same suggested empirical studies investigating the scheduling techniques used in the companies that have implemented the GT concept, can be applied to the Japanese companies.

APPENDICES

Table 1
Average "BEST" Ratio in S1

Exp. #	BEST (S1)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	5.000	4.000	5.000	0.000	4.000	3.60
3,3,20	8.000	4.000	5.000	3.000	0.000	4.00
3,10,3	5.000	5.000	3.000	1.000	3.000	3.40
3,10,10	8.000	0.000	2.000	0.000	0.000	2.00
3,10,20	5.000	3.000	2.000	0.000	0.000	2.00
3,20,3	4.000	6.000	6.000	1.000	2.000	3.80
3,20,10	6.000	2.000	2.000	0.000	0.000	2.00
3,20,20	6.000	0.000	4.000	0.000	0.000	2.00
10,3,10	6.000	5.000	2.000	4.000	1.000	3.60
10,3,20	6.000	6.000	5.000	1.000	1.000	3.80
10,10,3	1.000	5.000	3.000	0.000	2.000	2.20
10,10,10	5.000	3.000	1.000	0.000	1.000	2.00
10,10,20	4.000	1.000	3.000	1.000	1.000	2.00
10,20,3	3.000	2.000	3.000	0.000	2.000	2.00
10,20,10	4.000	3.000	3.000	0.000	0.000	2.00
10,20,20	4.000	2.000	3.000	1.000	0.000	2.00
15,3,10	4.000	5.000	7.000	2.000	1.000	3.80
15,3,20	4.000	3.000	5.000	2.000	0.000	2.80
15,10,3	3.000	4.000	0.000	3.000	0.000	2.00
15,10,10	5.000	1.000	4.000	0.000	0.000	2.00
15,10,20	5.000	2.000	3.000	0.000	0.000	2.00
15,20,3	3.000	3.000	4.000	0.000	0.000	2.00
15,20,10	4.000	2.000	4.000	0.000	0.000	2.00
15,20,20	2.000	2.000	6.000	0.000	0.000	2.00
TOTAL	110.00	73.00	85.000	19.000	18.000	61.000
%	45.83%	30.42%	35.42%	7.92%	7.50%	25.42%

Table 2

Average "WORST" Ratio in S1

Exp. #	WORST (S1)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	1.000	0.000	0.000	8.000	2.000	2.20
3,3,20	0.000	1.000	0.000	5.000	1.000	1.40
3,10,3	1.000	1.000	2.000	5.000	3.000	2.40
3,10,10	0.000	0.000	0.000	7.000	3.000	2.00
3,10,20	0.000	0.000	0.000	6.000	4.000	2.00
3,20,3	0.000	0.000	0.000	7.000	4.000	2.20
3,20,10	0.000	0.000	0.000	7.000	3.000	2.00
3,20,20	0.000	0.000	0.000	8.000	2.000	2.00
10,3,10	1.000	1.000	0.000	4.000	3.000	1.80
10,3,20	1.000	0.000	0.000	2.000	7.000	2.00
10,10,3	1.000	2.000	0.000	3.000	4.000	2.00
10,10,10	0.000	0.000	0.000	5.000	5.000	2.00
10,10,20	0.000	1.000	0.000	3.000	6.000	2.00
10,20,3	1.000	0.000	1.000	5.000	3.000	2.00
10,20,10	0.000	0.000	0.000	6.000	4.000	2.00
10,20,20	0.000	0.000	1.000	3.000	6.000	2.00
15,3,10	1.000	1.000	1.000	3.000	6.000	2.40
15,3,20	2.000	2.000	0.000	0.000	7.000	2.20
15,10,3	1.000	0.000	0.000	4.000	5.000	2.00
15,10,10	0.000	1.000	0.000	4.000	5.000	2.00
15,10,20	0.000	0.000	0.000	6.000	4.000	2.00
15,20,3	0.000	1.000	0.000	5.000	4.000	2.00
15,20,10	0.000	0.000	0.000	6.000	4.000	2.00
15,20,20	0.000	0.000	1.000	2.000	7.000	2.00
TOTAL	10.00	11.00	6.000	114.000	102.000	48.600
%	4.17%	4.58%	2.50%	47.50%	42.50%	20.25%

Table 3

Average "Efficiency" Ratio in S1

Exp. #	E (S1)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	0.991	0.991	0.994	0.969	0.982	0.99
3,3,20	0.999	0.996	0.997	0.988	0.995	1.00
3,10,3	0.985	0.986	0.982	0.964	0.969	0.98
3,10,10	0.997	0.982	0.991	0.932	0.960	0.97
3,10,20	0.996	0.983	0.993	0.944	0.946	0.97
3,20,3	0.985	0.993	0.991	0.975	0.980	0.98
3,20,10	0.997	0.985	0.987	0.940	0.961	0.97
3,20,20	0.996	0.983	0.991	0.936	0.949	0.97
10,3,10	0.995	0.993	0.994	0.994	0.993	0.99
10,3,20	0.997	0.999	0.997	0.997	0.991	1.00
10,10,3	0.986	0.987	0.992	0.964	0.972	0.98
10,10,10	0.996	0.968	0.962	0.950	0.948	0.96
10,10,20	0.996	0.991	0.995	0.981	0.976	0.99
10,20,3	0.994	0.980	0.981	0.960	0.971	0.98
10,20,10	0.994	0.990	0.991	0.965	0.968	0.98
10,20,20	0.997	0.992	0.990	0.973	0.972	0.98
15,3,10	0.998	0.997	0.999	0.995	0.991	1.00
15,3,20	0.997	0.997	0.999	0.998	0.993	1.00
15,10,3	0.989	0.987	0.983	0.974	0.959	0.98
15,10,10	0.991	0.983	0.994	0.967	0.964	0.98
15,10,20	0.999	0.993	0.995	0.978	0.994	0.99
15,20,3	0.990	0.985	0.987	0.961	0.964	0.98
15,20,10	0.994	0.992	0.994	0.966	0.973	0.98
15,20,20	0.992	0.991	0.995	0.975	0.971	0.98
AVG	0.994	0.988	0.991	0.969	0.972	0.983

Table 4

Average "Relative Error" Ratio in S1

Exp. #	R (S1)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	0.872	0.914	0.624	3.104	1.811	1.46
3,3,20	0.057	0.377	0.278	1.244	0.506	0.49
3,10,3	1.503	1.352	1.810	3.600	3.103	2.27
3,10,10	0.292	1.817	0.877	6.754	3.998	2.75
3,10,20	0.382	1.732	0.696	5.612	5.381	2.76
3,20,3	1.515	0.747	0.860	2.522	2.029	1.53
3,20,10	0.334	1.504	1.301	5.967	3.862	2.59
3,20,20	0.377	1.673	0.916	6.445	5.057	2.89
10,3,10	0.522	0.685	0.601	0.552	0.747	0.62
10,3,20	0.267	0.142	0.267	0.299	0.926	0.38
10,10,3	1.411	1.278	0.779	3.583	2.838	1.98
10,10,10	0.410	3.176	3.807	5.033	5.229	3.53
10,10,20	0.355	0.905	0.504	1.862	2.442	1.21
10,20,3	0.648	2.008	1.884	3.983	2.859	2.28
10,20,10	0.608	0.970	0.921	3.450	3.171	1.82
10,20,20	0.293	0.816	1.003	2.756	2.764	1.53
15,3,10	0.200	0.311	0.121	0.520	0.929	0.42
15,3,20	0.264	0.335	0.113	0.172	0.682	0.31
15,10,3	1.148	1.316	1.740	2.592	4.117	2.18
15,10,10	0.914	1.694	0.613	3.334	3.615	2.03
15,10,20	0.149	0.743	0.516	2.247	1.617	1.05
15,20,3	1.047	1.541	1.266	3.937	3.622	2.28
15,20,10	0.613	0.818	0.607	3.445	2.717	1.64
15,20,20	0.807	0.938	0.522	2.476	2.946	1.54
AVG	0.624	1.158	0.943	3.145	2.790	1.732

Table 5

Average "Error Potential Ratio" in S1

Exp. #	EPR (S1)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	27.631	28.678	17.947	87.717	45.104	41.42
3,3,20	5.098	23.705	16.686	57.944	46.937	30.07
3,10,3	30.997	27.500	40.944	74.861	51.433	45.15
3,10,10	5.942	24.862	11.762	95.424	59.306	39.46
3,10,20	5.371	28.582	11.836	86.303	84.683	43.35
3,20,3	40.160	25.925	27.820	74.167	70.362	47.69
3,20,10	4.576	23.276	21.780	89.049	62.843	40.30
3,20,20	5.851	24.228	12.093	95.301	76.758	42.85
10,3,10	22.871	27.953	39.613	46.486	48.799	37.14
10,3,20	30.292	12.981	16.631	43.353	87.704	38.19
10,10,3	37.056	33.096	19.398	80.865	62.066	46.30
10,10,10	17.650	23.419	44.712	79.169	69.986	46.99
10,10,20	11.957	31.993	16.556	60.971	90.970	40.49
10,20,3	18.487	37.333	38.684	67.826	52.703	43.01
10,20,10	13.108	23.957	20.137	71.344	75.468	40.80
10,20,20	9.673	21.642	35.367	69.266	84.047	44.00
15,3,10	24.302	29.384	16.856	56.318	79.565	41.28
15,3,20	39.765	36.270	14.742	22.822	85.207	39.76
15,10,3	29.573	25.309	35.083	56.084	86.990	46.71
15,10,10	21.764	42.065	13.452	73.290	82.858	46.69
15,10,20	7.356	28.495	20.125	85.319	71.133	42.49
15,20,3	23.175	37.131	22.289	83.007	80.626	49.27
15,20,10	14.386	24.650	13.690	84.987	67.305	41.00
15,20,20	25.036	26.974	15.303	74.111	85.844	45.45
AVG	19.670	27.913	22.609	71.499	70.779	42.494

Table 6

Average "BEST" Ratio in S5

Exp. #	BEST (S5)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	4.000	8.000	8.000	2.000	1.000	4.60
3,3,20	6.000	4.000	4.000	4.000	3.000	4.20
3,10,3	6.000	7.000	8.000	1.000	2.000	4.80
3,10,10	4.000	2.000	4.000	0.000	0.000	2.00
3,10,20	7.000	1.000	2.000	0.000	0.000	2.00
3,20,3	4.000	5.000	5.000	1.000	2.000	3.40
3,20,10	2.000	4.000	4.000	1.000	0.000	2.20
3,20,20	5.000	2.000	4.000	0.000	0.000	2.20
10,3,10	6.000	5.000	4.000	2.000	2.000	3.80
10,3,20	4.000	4.000	5.000	4.000	1.000	3.60
10,10,3	6.000	2.000	2.000	0.000	0.000	2.00
10,10,10	5.000	1.000	3.000	0.000	1.000	2.00
10,10,20	2.000	6.000	2.000	0.000	0.000	2.00
10,20,3	5.000	2.000	2.000	1.000	0.000	2.00
10,20,10	5.000	3.000	2.000	0.000	0.000	2.00
10,20,20	3.000	3.000	3.000	0.000	1.000	2.00
15,3,10	5.000	6.000	6.000	2.000	2.000	4.20
15,3,20	4.000	6.000	6.000	3.000	4.000	4.60
15,10,3	2.000	4.000	4.000	0.000	0.000	2.00
15,10,10	5.000	2.000	3.000	0.000	0.000	2.00
15,10,20	5.000	2.000	3.000	0.000	0.000	2.00
15,20,3	1.000	5.000	0.000	2.000	2.000	2.00
15,20,10	3.000	1.000	5.000	0.000	1.000	2.00
15,20,20	5.000	2.000	1.000	2.000	0.000	2.00
TOTAL	104.000	87.000	90.000	25.000	22.000	65.600
%	43.33%	36.25%	37.50%	10.42%	9.17%	27.33%

Table 7

Average "WORST" Ratio in S5

Exp. #	WORST (S5)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	4.000	0.000	2.000	4.000	2.000	2.40
3,3,20	0.000	1.000	0.000	5.000	3.000	1.80
3,10,3	2.000	1.000	1.000	4.000	4.000	2.40
3,10,10	1.000	1.000	0.000	5.000	3.000	2.00
3,10,20	0.000	0.000	0.000	4.000	6.000	2.00
3,20,3	1.000	1.000	1.000	3.000	6.000	2.40
3,20,10	0.000	0.000	0.000	8.000	1.000	1.80
3,20,20	0.000	0.000	0.000	5.000	5.000	2.00
10,3,10	1.000	1.000	1.000	6.000	3.000	2.40
10,3,20	1.000	1.000	1.000	0.000	6.000	1.80
10,10,3	0.000	1.000	0.000	6.000	3.000	2.00
10,10,10	0.000	0.000	0.000	7.000	3.000	2.00
10,10,20	1.000	0.000	1.000	1.000	7.000	2.00
10,20,3	0.000	0.000	2.000	3.000	5.000	2.00
10,20,10	0.000	2.000	1.000	5.000	2.000	2.00
10,20,20	1.000	0.000	0.000	6.000	3.000	2.00
15,3,10	1.000	0.000	1.000	2.000	5.000	1.80
15,3,20	3.000	0.000	0.000	2.000	4.000	1.90
15,10,3	0.000	0.000	0.000	4.000	6.000	2.00
15,10,10	0.000	0.000	0.000	4.000	6.000	2.00
15,10,20	0.000	0.000	0.000	3.000	7.000	2.00
15,20,3	2.000	2.000	1.000	4.000	2.000	2.20
15,20,10	0.000	0.000	1.000	7.000	2.000	2.00
15,20,20	0.000	1.000	1.000	2.000	6.000	2.00
TOTAL	18.000	12.000	14.000	100.000	100.000	48.800
%	7.50%	5.00%	5.83%	41.67%	41.67%	20.33%

Table 8

Average "Efficiency" Ratio in S5

E.p. #	E (S5)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3.3.10	0.983	0.993	0.997	0.990	0.990	0.99
3.3.20	0.996	0.994	0.995	0.988	0.991	0.99
3.10.3	0.991	0.995	0.996	0.977	0.981	0.99
3.10.10	0.991	0.986	0.990	0.962	0.958	0.98
3.10.20	0.997	0.985	0.990	0.965	0.964	0.98
3.20.3	0.993	0.995	0.995	0.973	0.980	0.99
3.20.10	0.991	0.988	0.989	0.953	0.962	0.98
3.20.20	0.993	0.990	0.996	0.941	0.944	0.97
10.3.10	0.997	0.997	0.994	0.991	0.994	0.99
10.3.20	0.998	0.997	0.996	0.998	0.990	1.00
10.10.3	0.992	0.983	0.986	0.966	0.974	0.98
10.10.10	0.994	0.987	0.989	0.955	0.968	0.98
10.10.20	0.994	0.996	0.990	0.984	0.972	0.99
10.20.3	0.990	0.981	0.975	0.972	0.965	0.98
10.20.10	0.992	0.984	0.992	0.970	0.973	0.98
10.20.20	0.993	0.993	0.993	0.976	0.978	0.99
15.3.10	0.995	0.996	0.999	0.994	0.991	0.99
15.3.20	0.998	0.999	1.000	0.999	0.995	1.00
15.10.3	0.988	0.996	0.991	0.970	0.965	0.98
15.10.10	0.994	0.988	0.987	0.967	0.968	0.98
15.10.20	0.995	0.989	0.990	0.987	0.975	0.99
15.20.3	0.980	0.986	0.983	0.972	0.976	0.98
15.20.10	0.993	0.990	0.989	0.969	0.981	0.98
15.20.20	0.998	0.993	0.985	0.982	0.978	0.99
AVG	0.993	0.990	0.991	0.975	0.975	0.985

Table 9
Average "Relative Error" Ratio in S5

Exp. #	R (S5)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3.3.10	1.720	0.690	0.271	0.974	0.965	0.92
3.3.20	0.374	0.642	0.481	1.210	0.911	0.72
3.10.3	0.832	0.474	0.439	2.349	1.929	1.21
3.10.10	0.931	1.446	1.036	3.678	4.190	2.26
3.10.20	0.339	1.518	0.986	3.470	3.650	1.99
3.20.3	0.672	0.486	0.528	2.716	2.000	1.28
3.20.10	0.867	1.198	1.093	4.689	3.798	2.33
3.20.20	0.711	1.031	0.422	5.908	5.606	2.74
10.3.10	0.339	0.264	0.613	0.917	0.627	0.55
10.3.20	0.230	0.258	0.382	0.150	0.988	0.40
10.10.3	0.793	1.703	1.426	3.368	2.576	1.97
10.10.10	0.623	1.332	1.054	4.473	3.235	2.14
10.10.20	0.644	0.407	0.987	1.618	2.823	1.30
10.20.3	1.012	1.909	2.532	2.821	3.530	2.36
10.20.10	0.768	1.625	0.847	3.002	2.733	1.80
10.20.20	0.704	0.714	0.676	2.445	2.177	1.34
15.3.10	0.520	0.357	1.099	0.566	0.876	0.68
15.3.20	0.167	0.084	0.039	0.135	0.546	0.19
15.10.3	1.154	1.418	0.947	2.992	3.549	2.01
15.10.10	0.642	1.189	1.324	3.285	3.238	1.94
15.10.20	0.494	1.090	0.966	1.309	2.456	1.26
15.20.3	2.005	1.264	1.700	2.823	2.377	2.05
15.20.10	0.733	0.962	1.060	3.128	1.855	1.55
15.20.20	0.244	0.702	1.457	1.753	2.225	1.28
AVG	0.732	0.953	0.922	2.491	2.452	1.512

Table 10
Average "Error Potential" Ratio in S5

Exp. #	EFR (S5)					
	NEH	MOD(1)	MOD(2)	SOI	CDS	AVG
3,3,10	55.198	16.654	20.000	46.496	49.139	37.50
3,3,20	14.090	30.115	18.810	52.029	48.153	32.64
3,10,3	25.571	11.143	18.033	68.491	54.703	35.59
3,10,10	32.995	28.489	20.759	69.687	80.275	46.44
3,10,20	9.670	35.799	22.468	80.204	86.958	47.02
3,20,3	22.725	16.556	24.652	57.965	72.911	39.00
3,20,10	16.696	18.440	17.682	87.041	76.891	43.35
3,20,20	9.961	11.629	7.335	87.501	83.792	40.06
10,3,10	22.977	20.917	37.355	69.394	54.058	40.94
10,3,20	30.397	24.501	27.974	18.301	80.385	36.31
10,10,3	14.636	32.596	30.140	82.240	52.959	42.51
10,10,10	13.786	17.447	21.662	85.588	60.909	39.88
10,10,20	24.591	6.857	35.456	55.096	86.654	41.73
10,20,3	22.986	29.314	54.023	56.843	73.624	47.36
10,20,10	18.429	43.333	22.147	67.078	62.385	42.67
10,20,20	23.500	27.087	29.948	89.048	66.659	47.25
15,3,10	25.326	20.275	25.327	42.412	54.322	33.53
15,3,20	41.329	1.873	3.688	33.203	57.652	27.55
15,10,3	28.121	24.319	27.530	70.190	84.408	46.91
15,10,10	13.618	20.287	34.191	76.754	83.308	45.63
15,10,20	22.123	35.620	27.757	54.552	94.834	49.00
15,20,3	45.234	22.791	45.434	65.685	50.161	45.86
15,20,10	18.051	29.031	36.085	89.014	58.238	46.08
15,20,20	11.219	29.756	50.212	56.089	77.873	45.03
AVG	23.468	23.120	27.869	65.042	68.802	41.660

Table 11

ANOVA Table (Factorial Experiments Design)

Effects of Problem Size, Heuristic, and Their Interaction (S1)

Source of variation	Sum of squares	D.F.	Mean squares	F	F/table	Result
Problem size	887.279	23	38.577	8.6612	1.81 **	Reject
Heuristic	1279.688	4	319.922	71.828	3.34 **	Reject
Interaction	675.067	92	7.338	1.6475	1.44 **	Reject
Error/ Chance	4810.557	1080	4.454			
Total	7652.59					

** ; 1% significance level

Table 12

ANOVA Table (Factorial Experiments Design)

Effects of Problem Size, Heuristic, and Their Interaction (S5)

Source of variation	Sum of squares	D.F.	Mean squares	F	F/table	Result
Problem size	548.347	23	23.841	12.915	1.81 **	Reject
Heuristic	744.013	4	186.003	100.760	3.34 **	Reject
Interaction	487.202	92	5.296	2.869	1.44 **	Reject
Error/ Chance	1993.321	1080	1.846			
Total	3772.883					

** : 1% significance level

Table 13

ANOVA Table (Factorial Experiments Design)
Effects of Problem Size, Family Setup Time, and
Their Interaction (S1)

Heur.	Source	S.S.	D.F.	M.S.	F	F/table	Result
NEH	P	62.384	23	2.712	2.699	1.84 **	Reject
	S	1.386	1	1.386	1.379	6.70 **	Accept
	P&S	24.373	23	1.060	1.055	1.84 **	Accept
	E	434.27	432	1.005			
	T	522.409					
MOD1	P	139.324	23	6.058	2.057	1.84 **	Reject
	S	5.054	1	5.054	1.716	6.70 **	Accept
	P&S	28.310	23	1.231	.418	1.84 **	Accept
	E	1272.2	432	2.945			
	T	1444.89					
MOD2	P	131.263	23	5.707	2.021	1.84 **	Reject
	S	.011	1	.011	.0039	6.70 **	Accept
	P&S	74.508	23	3.239	1.147	1.84 **	Accept
	E	1220.13	432	2.824			
	T	1425.912					
SOI	P	1221.777	23	53.121	11.805	1.84 **	Reject
	S	51.344	1	51.325	11.406	6.70 **	Reject
	P&S	84.826	23	3.689	.820	1.84 **	Accept
	E	1943.788	432	4.500			
	T	3301.735					
CDS	P	773.141	23	33.615	7.510	1.84 **	Reject
	S	15.104	1	15.104	3.374	6.70 **	Accept
	P&S	58.074	23	2.525	.564	1.84 **	Accept
	E	1933.491	432	4.476			
	T	2779.81					

P : problem size
S : setup time
E : error
T : total
** : 1% significance level

Table 14

ANOVA Table (Randomized Block Design)
Effects of the No. of Families, Machines,
and Jobs, With Heuristic.(S1)

Source	S.S.	D.F.	M.S.	F	F/table	Result
# of Families	1.127	2	.5635	2.4989	8.65 **	Accept
Heuristic	15.887	4	3.9718	17.613	7.01 **	Reject
Error/ Chance	1.804	8	.2255			
Total	18.818					
# of Machines	7.482	2	3.741	.4659	8.65 **	Accept
Heuristic	13.428	4	3.357	.4181	7.01 **	Accept
Error/ Chance	64.236	8	8.0295			
Total	85.146					
# of Jobs	1.43	2	.715	1.8124	8.65 **	Accept
Heuristic	15.568	4	3.892	9.8657	7.01 **	Reject
Error/ Chance	3.156	8	.3945			
Total	17.294					

** : 1% significance level

BIBLIOGRAPHY

1 Aggarwal, Sumer C., and E. Stafford. "A Heuristic Algorithm for the Flow Shop with a Common Job Sequence on All Machines." Decision Science 6 (April 1975): 237-251.

2 -----------------. "MRP, JIT, OPT, FMS?." Harvard Business Review, Special Report (Spt./Oct. 1985): 8-10, 12-16.

3 Anderson, David R., Dennis J. Sweeney, and Thomas A. Williams. Statistics for Business and Economics. 3rd ed. N.Y.: West Publishing, 1987.

4 Aneke, N. A. G., and A. S. Carrie. "A Comparative Flow Line Classification Scheme." International Journal of Production Research (GB). 22 (March/April 1984): 281-287.

5 Arn, E. A. GT - An Integrated Planning and Implementation Concept for Small- and Medium-Batch Production. Berlin: Springer-Verlag, 1975.

6 Asfahl, C. Ray. Robots and Manufacturing Automation. N.Y.: John Wiley, 1985.

7 Ashour, Said. "A Branch and Bound Algorithm for Flow-Shop Scheduling Problems." AIIE Transactions 2 (1970): 172-176.

8 ------------. "An Experimental Investigation and Comparative Evaluation of Flow Shop Scheduling Techniques." Operations Research 18 (May/June 1970): 541-549.

9 Ashton, James E., and Frank X. Cook Jr. "Time to Reform Job Shop Manufacturing." Harvard Business Review (March/April 1989): 106-111.

10 Baker, K. Introduction to Sequencing and Scheduling. N.Y.: John Wiley, 1974.

11 Bignell, V., M. Dooner, J. Hughes, C. Pym, and S. Stone, eds. Manufacturing Systems: Context, Applications and Techniques. London: Basil Blackwell, 1985.

12 Black, J. T. "Cellular Manufacturing Systems Reduce Setup Time, Make Small-Lot Production Economical." Industrial Engineering (Nov. 1983): 36-48.

13 Burbidge, John L. "Production Flow Analysis." The Production Engineer 42 (Dec. 1963): 742.

14 ----------------. "Production Flow Analysis - A Technique for Finding the Families and Groups for GT." First International Seminar on GT, The International Center for Advanced Technology and Vocational Training, Turin, Italy, Spt. 1969.

15 ----------------. The Introduction of Group Technology. N.Y.: John Wiley, 1975.

16 ----------------. "Whatever Happened to GT." Management Today (Spt. 1978): 78-93.

17 ----------------. Group Technology in the Engineering Industry. London: Mechanical Engineering Publications, 1979.

18 ----------------, and B. G. Dale. "Planning Introduction and Predicting Benefits of Group Technology." Engineering Costs and Production Economics 8 (July 1984): 117-128.

19 ----------------. "Operation Scheduling with GT and PBC (Period Batch Control)." International Journal of Production Research (U.K.) 26 (March 1988): 429-442.

20 Campbell, H. G., R. A. Dudek, and M. L. Smith. "A Heuristic Algorithm for the n-Job, m-Machine Sequencing Problem." Management Science 16 (1970): B630-B637.

21 Chakravarty, A. K., and A. Shtub. "The Use of Group Technology Concepts in Scheduling Flexible Manufacturing Systems." Proceedings of the 16th Annual Meeting of the American Institute for Decision Sciences, Toronto, Canada, Nov. 5-7, 1984, 493-495.

22 Cho, K., E. E. Enscore, and I. Ham. "A Heuristic Algorithm to Minimize Total Tardiness." Proceedings of the 10th American Manufacturing Research Conference, Society of Manufacturing Engineers, 1982, 514-517.

23 Coffman, E. G. Computer and Job-Shop Scheduling Theory. N.Y.: John Wiley, 1976.

24 Conway, Richard W., W. Maxwell, and W. L. Miller. Theory of Scheduling. M.A.: Addison Wesley Publishing, 1967.

24a Cook, Stephen A. "The Complexity of Theorem-Proving Procedures." Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, Shaker Heights, Ohio, May 3-5, 1971, 151-158.

25 Dale, Barrie G., and Phillip C. T. Willey. "The Need to Predict the Potential for GT in Manufacture." Machinery and Production Engineering 130 (Jan. 5, 1977): 9-15.

26 --------------. "GT: Predicting Its Benefits." Work Study (Feb. 1980).

27 Dannenbring, David G. "The Evaluation of Heuristic Solution Procedures for Large Combinatorial Problems." Ph.D. Diss., Colombia University, 1973.

28 Dannenbring, David G. "An Evaluation of Flow Shop Sequencing Heuristics." Management Science 23 (July 1977): 1174-1182.

29 Dempster, M. A. H., J. K. Lenstra, and A. A. Kan. "Deterministic and Stochastic Scheduling." Proceedings of the NATO Advanced Study and Research Institution on Theoretical Approaches to Scheduling Problems, Durham, England, July 6-17, 1981.

30 Devries, M. et. al. GT: An Overview and Bibliography. Cincinnati, Ohio: Machinability Data Center, 1976.

31 Dumolien, William J., and William P. Santen. "Cellular Manufacturing Becomes Philosophy of Management at Components Facility." Industrial Engineering (Nov. 1983): 72-76.

32 Dutkosky, R., K. Hitomi, and I. Ham. "Production Scheduling in GT Application: Yesterday's Concept in Tomorrow's World." ASME, Technical Paper (1976): MS 76-275.

33 Elgomayel, J., and V. Nader. "Optimization of Machine Setup and Tooling Using the Principles of GT." Computers and Industrial Engineering 7,3 (1983): 187-198.

34 Emmons, H. "One-Machine Sequencing to Minimize Certain Functions of Job Tardiness." Operations Research 17,3 (1969): 701-715.

35 Fazakerley, G. Margaret. "GT- Social Benefits and Social Problems." The Production Engineer 53 (Oct. 1974): 383-386.

36 Flynn, B. B. "Repetitive Lots: The Use of Sequence-Dependent Setup Time Scheduling Procedure in GT and Traditional Shops." Journal of Operations Management 7,1/2 (Oct. 1987): 203-216.

37 Flynn, B. B., and F. Robert Jacobs. "An Experimental Comparison of Cellular Layout with Process Layout." Decision Sciences 18 (Fall 1987): 562-581.

38 Foo, F. C., and J. G. Wager. "Setup Times in Cyclic and Acyclic GT Scheduling Systems." International Journal of Production Research 21 (Jan./Feb. 1983): 63-64.

39 Ford, F. Nelson, William N. Ledbetter, and Brian S. Gaber. "The Evolving Factory of the Future: Integrating Manufacturing and Information Systems." Information and Management (Netherlands) 8 (Feb. 1985): 75-80.

40 French, Simon. Sequencing and Scheduling: An Introduction to the Mathematics of the Shop. Chichester: Ellis Howard, 1982.

41 Gallagher, C. C., and W. A. Knight. Group Technology. London: Butter Worths, 1973.

42 ----------------------------------. Group Technology: Production Methods in Manufacture. Chichester: Halstead Press, 1986.

43 Gold, Bela. "CAM Sets New Rules for Production." Harvard Business Review (Nov./Dec. 1982): 88-94.

44 Grayson, T. J., and G. A. Oddie. A Bibliography of Papers, Books, and Journals Dealing with GT and Its Related Subjects. U.K.: University of Birmingham, 1965.

45 --------------. Group Technology, A Comprehensive Bibliography. U.K.: GT Center, Atomic Energy Authority, 1972.

46 --------------. "Some Research Findings on Evaluating Group Technology." International Journal of Production Research 16,2 (1978): 89-103.

47 Groover, Mikell P., and Emory W. Zimmers Jr. CAD/CAM: Comuter-Aided Design and Manufacturing. Englewood Cliffs, N.J.: Prentice-Hall, 1984.

48 -----------------. Automation, Production Systems and Computer-Integrated Manufacturing. Englewood Cliffs, N.J.: Prentice-Hall, 1987.

49 Guerrero, H. H. "GT I: The Essential Concept." Production and Inventory Managemnt 28 (First Quarter 1987): 62-70.

50 --------------. "GT II: The Implementation Process." Production and Inventory Management 28 (Second Quarter 1987): 1-9.

51 Gunn, Thomas G. Computer Applications in Manufacturing. N.Y.: Industrial Press, 1981.

52 --------------. "The Mechanization of Design and Manufacturing." Scientific American 47 (Spt. 1982): 115-130.

53 Gupta, J. N. D. "A Functional Heuristic Algorithm for the Flow-Shop Scheduling Problem." Operational Research Quarterly 22,1 (1971): 39-48.

54 --------------, and R. A. Dudek. "Optimality Criteria for Flow-Shop Schedules." AIIE Transactions 3 (1971): 199-206.

55 -----------------. "Optimal Schedules for Special-Structure Flow shop." Naval Research Logistics Quarterly 22,2 (1975): 255-269.

56 Ham, I. "Selected References on GT in English." Society of Manufacturing Engineering, Technical Paper No. MR 71-284, 1971.

57 -------, and W. Reed. "First Group Technology Survey Reveals New Manufacturing Game Plan." Machine and Tool Blue Book 72,5 (1977): 100-108.

58 Ham, I., K. Hitomi, N. Nakamura, and T. Yoshida. "optimal Group Scheduling and Machining-Speed Decision Under Due-Date Constraints." ASME Transactions, Journal of Engineering for Industry 101 (May 1979): 128-134.

59 -------. "Current Trend and Future Prospects of GT Applications Related to Integrated Computer-Aided Manufcturing." Proceedings of Conference on International Manufacturing Engineering, Australia, The Institucion of Engineers, Aug. 1980.

60 -------, K. Hitomi, and T, Yoshida. GT Applications to Production Management. International Series in Management Science/ Operations Research. Boston: Klumer-Nijhoff Publishing, 1985.

61 Harvey, Robert E. "How the Machining Cell Makes Flexibility." Iron Age 57 (Feb. 25, 1983): 57-60.

62 Hax, Arnoldo C., and Dan Candea. Production and Inventory Management. N.J.: Prentice-Hall, 1984.

63 Hitomi, K., and I.Ham. "Operations Scheduling for GT Applications." CIRP Annals 25,1 (1976): 419-422.

64 ---------------------. "Group Scheduling Technique for Multi-Product, Multi-Stage Manufacturing Systems." ASME, Journal of Engineering for Industry 99B (1977): 759-765.

65 -----------, N. Nakamura, T. Yoshida, and K. Okuda. "An Experimental Investigation of Group Production Scheduling." Proceedings of the 4th International Conference on Production Research, Tokyo, Japan, 1977, 607.

66 Houtzeel, Alexander. "Integrating CAD/CAM Through GT." Paper Presented at The American Production and Inventory Control Society, Operations Management Workshop, Michigan State University, July 26-28, 1982.

67 Huber, R. F. "Transfer Line Hosts Family of Parts." Production 75 (April 1975): 70-72.

68 Hyde, William F. Improving Productivity by Classification, Coding, and Data Base Standardization- The Key to Maximizing CAD/CAM and GT. N.Y.: Marcel Dekker, 1981.

69 Hyer, Nancy Lea. "Group Technology: Development, Diffusion, and Applications in the U.S. and International Context." Ph.D. diss., School of Business, Indiana University, 1982.

70 ----------------. "The Potential of Group Technology for U.S. Manufacturing." Journal of Operations Management 4 (May 1984): 183-202.

71 ----------------, and U. Wemmerlov. "Group Technology and Its Role in Modern Manufacturing Systems." Proceedings of the 16th Annual Meeting of the American Institute for Decision Sciences, Toronto, Canada, Nov. 5-7, 1984, 387.

72 ---------------------------------. "Group Technology Oriented Coding Systems: Structures, Applications, and Implementation." Production and Inventory Management (Second Quarter 1989): 55-78.

73 ---------------------------------. "Group Technology in the U.S. Manufacturing Industry: A Survey of Current Practices." Proceedings of the 1986 Annual Meeting of the Decision Sciences Institute, Hawaii, Nov. 23-25, 1986, 295-297.

74 Ignall, E., and L. Schrage. "Application of the Branch and Bound Techniques to Some Flow-Shop Scheduling Problems." Operations Research 13 (1965): 400-412.

75 Jackson, David. Cell System of Production- An Effective Organization Structure. London: Business Book, 1978.

76 Jacobs, F. R., and D. J. Bragg. "Repetitive Lots: Flow-Time Reductions Through Scheduling and Dynamic Batch Sizing." Decision Sciences 19,2 (1988): 281-294.

77 Johnson, S. N. "Optimal Two-and Three-Stage Production Schedules with Setup Times Included." Naval Research Logistics Quarterly 1 (1954): 61-68.

77a Karp, Richard M. "Combinatorics, Complexity, and Randomness." Communications of the ACM 29 (Feb. 1986): 98-111.

78 Kilpatrick, Michale, Business Statistics Using Lotus 1-2-3. N.Y.: John Wiley, 1987.

79 King, J. R. "Scheduling and the Problem of Computational Complexity." Omega 7,3 (1979): 233-240.

80 Knox, Charlie. "CAD/CAM and GT: The Answer to Systems Integration?." Industrial Engineering (Nov. 1980): 66-73.

81 Koenig, D., T. Gongaware, and I. Ham. "Applications of GT Concept for Plant Layout and Management of a Miscellaneous Parts Shop." Proceedings of the 9th North American Manufacturing Research Conference, The Penn. State University, By SME Transactions, May 19-22, 1981, 497-502.

82 Kusiak, Andrew. "Designing Expert Systems for Scheduling of Automated Manufacturing." Industrial Engineering 19 (July 1987): 42-46.

83 --------------. "An Expert System for Group Technology." Industrial Engineering 19 (Oct. 1987): 56-61.

84 Langston, Michael A. "Remarks on the Makespan Minimization Problem." Computers and Industrial Engineering 8,3/4 (1984): 193-196.

85 Lee, L. C. "Group Technology." Management Decision 25,4 (1987): 33-37.

86 Leonard, R. "Group Technology: The State of the Art." Machinery and Production Engineering 137 (Oct. 29, 1980): 31-32.

87 Lindman, H. R. Analysis of Variance in Complex Experimental Designs. San Francisco: W. H. Freeman, 1974.

88 MacConnell, W. R. "Why Companies Make the Decisions to Implement GT." Proceedings of the 2nd Annual Confernce of GT Economic Aspects, Institution of Production Engineers, Banbury: Whately Hall, Nov. 28-29, 1972.

89 Marklew, J. J. "An Application of the Cell System in a Small Machine Shop." Machinery and Production Engineering 121 (Aug. 23, 1972): 259-263.

90 Masaru, Nakajima, and Hayash Kenichi. "Automated Parts Layout Planning in Garment Industry by Using GT." Proceedings of the 10th North American Manufacturing Research Conference, Society of Manufacturing Engineering, 1982.

91 McLauchlan, G. "Grouping Families of Tools avoids Duplication-Cuts Design Time." Machine and Tool Blue Book 67 (March 1972): 64-69.

92 Millar, G. W. "Think GT Before System Integration." Manufacturing Engineering 6 (Nov. 1985): 95-98.

93 Mitrofanov, S. P. Scientific Principles of GT. Boston: National Lending Library for Science and Technology, 1966.

94 Mosier, C. T., D. A. Elvers, and D. Kelly. "Analysis of GT Scheduling Heuristics." International Journal of Production Research 22 (Spt./Oct. 1984): 857-875.

95 Mosier. C. T., and D. A. Elvers. "Heuristic Implementations of Group Technology Scheduling." AIDS Conference Proceedings, Toronto, Canada, Nov. 1984.

96 Nakamura, N., T. Yoshida, and K. Hitomi. "Group Production Scheduling for Minimum Total Tardiness." AIIE Transactions 10 (June 1978): 157-162.

97 Nawaz, M., E. E. Enscore, and I. Ham. "A Heuristic Algorithm for the m-Machine, n-Job Flow Shop Sequencing Problem." Omega 41,1 (1983): 91-95.

98 Ncllmann, David O., and Leighton F. Smith. "Just-In-Time vs. Just-In-Case Production Inventory Systems, Concepts Borrowed Back from Japan." Production and Inventory Management (Second Quarter 1982): 12-21.

99 Nichal, J. D. "Production Cost Savings." Proceedings of the 2nd Annual Conference on GT Economic Aspects, Institution of Production Engineers, Banbury: Whately Hall, Nov. 28-29, 1972.

100 Nisanci, Ibrahim. "Use of GT Concepts in Integrated Production Planning." Proceedings of the NATO Advanced Research Institute, Amsterdam, The Netherlands, Spt. 6-10, 1982, 149-152. In: Efficiency of Manufacturing Systems, N.Y.: Plenum Press, 1983.

101 Oh, Keytack H., and William Gravel. "GT Applications in Manufacturing Operations Through Computer System." Proceedings of the 8th Annual Conference on Computers and Industrial Engineering. In Computers and Industrial Engineering 11,1-4 (1986): 170-174.

102 Ozden, M., P. J. Egbelu, and A. V. Iyer. "Job Scheduling in a GT Environment for a Single Facility." Computers and Industrial Engineering 9,1 (1985): 67-72.

103 Page, E. S. "On the Scheduling of Jobs by Computer." The Computer Journal 5,3 (1962): 214-221.

104 Palmer, D. S. "Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time, A Quick Method of Obtaining a Near Optimum." Operational Research Quarterly 16,1 (1965): 101-107.

105 Park, Yang B., C. D. Pegden, and E. E. Enscore. "A survey and Evaluation of Static Flow Shop Problems." International Journal of Production Research 22,1 (1984): 127-141.

106 -------------. "An Evaluation of Static Flow Shop Scheduling Heuristics in Dynamic Flow Shop Models via Computer Simulation." Computers and Industrial Engineering (U.K.) 14,2 (1988): 103-112.

107 PERA (Production Engineering Research Association). Methods of Loading and Scheduling in GT Cell Systems. Report No. 309, PERA, 1976.

108 Petrov, V. A. Flow Line Group Production Planning. Yorkshire: Business Publications, 1968.

109 Pullen, R. D. "A Survey of Cellular Manufacturing Cells." The Production Engineer (Spt. 1976): 451-454.

110 Radharmanan, R. "A Heuristic Algorithm for Group Scheduling." Computers and Industrial Engineering 11,1-4 (1986): 204-208.

111 Ralstadas, A. "Scheduling Batch Production by Means of Mini-Computer." Proceedings of Conference on Computer-Aided Manufacture, Glascow: National Engineering Laboratory, 1978, 1-16.

112 Salmon, B. "GT: The Key to Integrated Manufacturing." Proceedings of CADCON West, Anaheim, C.A., Jan. 14-17, 1958, 12-20.

113 Sandy, Robert. Statistics for Business and Economics. N.Y.: McGraw-Hill Publishing, 1989.

114 Sanl, Greg. "Flexible Manufacturing System Implemented at the Floor Level." Industrial Engineering (June 1988): 34-39.

115 Schaffer, G. H. "GT Expands Capacity." American Machinist 124 (Jan. 1980): 130-134.

116 ---------------. "GT via Automated Process Planning." American Machinist 124 (May 1980): 119-122.

117 Schonberger, Richard J. "Some Observations on the Advantages and Implementation Issues of Just-In-Time Production Systems." Journal of Operations Management 3,4 (1982): 1-11.

118 ----------------------. "Plant Layout Becomes Product-Oriented with Cellular-Just-In-Time-Production Concept." Industrial Engineering (Nov. 1983): 66-71.

119 Setiaputra, W. "A Survey of Flow-Shop Permutation Scheduling Techniques and an Evaluation of Heuristics' Solution Methods." Master's Thesis, Penn. State University, 1980.

120 Shunkk, Dan K. "GT Provides Organized Approach to Realizing Benefits of CIMS." Industrial Engineering 17 (April 1985): 74-80.

121 Silver, E. A., R. V. Vidal, and D. De Werra. "A Tutorial on Heuristics Methods." European Journal of Operational Research 5 (1980): 153-162.

122 Simon, H. A., and A. Newell. "Heuristics Problem Solving: The Next Advance in Operations Research." Operations Research 6 (Jan./Feb. 1958): 1-10.

123 Smith, A. M. "Financial Justification." Proceedings of the 2nd Annual Conference on GT Economic Aspect, Institution of Production Engineers, Banbury: Whately-Hall, Nov. 28-29, 1972.

124 -------------, and E. Stafford. "A Multivariate Analysis of Heuristic Solutions to Large Scale Flow Shop Scheduling Problems." Working Paper Series- WP 79-33, Management Research Center, School of Management, Syracuse University. A Paper Presented at The Institute of Management Sciences, XXIV International Meeting, Hawaii, June 1979.

125 Spencer, M. S. "Scheduling Componenets for GT Lines: A New Application for MRP." Production and Inventory Management (Fourth Quarter 1980): 43-49.

126 Star, J. "Production Flow Analysis Complements GT." Metal Working Production 113 (Oct. 1969): 56-60.

127 Starr, Martin K. Managing Production and Operations. N.Y.: Prentice Hall, 1989.

128 Sundaram, R. M. "A Group Scheduling Algorithm for Jobs That Utilize GT Manufacturing Concept." Proceedings of the 10th North American Manufacturing Research Conference, Society of Manufacturing Engineers, 1982.

129 --------------. "Some Scheduling Algorithms for GT Manufacturing System." Proceedings of the 1st International Conference on Computer Applications in Production and Engineering, CAPE,83, Amsterdam, The Netherlands, April 25, 1983, 765-772.

130 Taylor, J., and I. Ham. "The Use of a Micro Computer for Group Scheduling." Proceedings of the 9th North American Manufacturing Research Conference, The Penn. Stste university, May 19-22, 1981, 483-491.

131 Tuttle, H. C. "Parts Grouping Pays Handsomely." Production 73 (May 1974): 99-105.

132 Vaithianathan, R., and K. L. McRoberts. "On Scheduling in GT Environment." Proceedings of the AUTOFACT III Conference, Society of Manufacturing Engineers, 1981.

133 Vakharia, A. J., and U. Wemmerlov. "Flow Shop Scheduling With Consideration to Part Families." Proceedings of the Annual Meeting of the American Institute for Decision Science, Las Vegas, Nov. 11-13, 1985, 790-792.

134 Waghodekar, A., and J. J. Emang. "Group Technology: A Research Bibliography." Opsearch (India) 20,4 (Dec. 1983): 225-249.

135 Welch, A., and J. J. Emang. "GT: the Heart of FMS." Proceedings of the First International Conference of FMS, Brighton, England, Oct. 20-22, 1982, 121-130.

136 Wemmerlov, Urban, and Nancy Hyer. "Cellular Manufacturing in the US Industry: A Survey of Users." Proceedings of the 1978 Annual Meeting of the Decision Science Institute, Boston, Nov. 23-25, 1987, 802-804.

137 ---------------, and A. Vakharia. "Job and Family Scheduling of a Flow-Line Manufacturing Cell: A Simulation Study." Working Paper (Received from the author by mail), 1988.

138 Yoshida, T., and K. Hitomi. "Optimal Two-Stage Production Scheduling with Setup Times Separated." AIIE Transactions 11,3 (Spt. 1979): 261-263.

139 "Eaton Corporations: GT at the Heart of CIM." Mechanical Engineering 107 (Nov. 1985): 66-68.

140 "GT: A Comparative Approach." Tooling and Production 47 (Feb. 1982): 96-98.

141 "GT: An Integrating Force." Industry Week 61 (May 26, 1986): 229.

142 "GT: Ready for a Big Step." Industry Week (March 23, 1987):151.

143 "GT: Its Time Now." Production 99 (Dec. 1987): 66-67.

144 "How to Get Mass Production Benefit from Small Lot Output." Management Review 67 (Oct. 1978): 55-56.