DESIGN AND IMPLEMENTATION OF FLEXIBLE MANUFACTURING SYSTEMS -- SOME ANALYSIS CONCEPTS

by

Henry C. Co

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Industrial Engineering & Operations Research

APPROVED:

R. A. Wysk, Co-Chairman

W. J. Fabrycky
Co-Chairman

T. J. Greene

R. D. Foley

C. E. Nunnally

April, 1984
Blacksburg, Virginia

DESIGN AND IMPLEMENTATION OF FLEXIBLE MANUFACTURING SYSTEMS -- SOME ANALYSIS CONCEPTS

by
Henry C. Co

(ABSTRACT)

This study presents some analysis concepts and decision tools for the problems encountered in designing and implementing a flexible manufacturing system.

SIM-Q, an input-generator for simulation modelling developed in this study provides a powerful and expedient tool for resolving the material handling system selection, work scheduling, input control, and real time operation problems. The problem of input control is examined using SIM-Q and the viability of operating a flexible manufacturing system as a programmable transfer line is explored. SIM-Q is also used in this study to test the robustness of CAN-Q in modelling an existing FMS.

A linear zero-one linear programming model is formulated for the machine mapping and pooling problem. The system synthesis problem is solved by applying CAN-Q in an interactive computer program developed in this study. This model provides an integrated approach to the product selection and machine requirements planning problems. Finally, a dynamic decision approach to the justification of the FMS is developed and presented by imbedding queueing theory with simulation in a decision analysis framework.

ACKNOWLEDGMENTS

I am indebted to Profesor Richard A. Wysk -- for suggesting the topic area, for his constructive criticisms, perpetual encouragement, and for his continued willingness to read more drafts of this dissertation than I was ever willing to write.

A special note of gratitude is due , who helped me get started at Virginia Tech. I also wish to thank for his hospitality, to for collaborating with me on the contents of Chapter six, and to for her assistance in preparing this manuscript.

I would like to acknowledge the generous support of the Cunningham Scholarship Committee, and to thank the members of my dissertation committee, Professors W. J. Fabrycky, R. D. Foley, Timothy J. Greene, Charles E. Nunnally, and Richard A. Wysk.

TABLE OF CONTENTS

Chapter

page

<u>Appendix</u>

<u>page</u>

LIST OF TABLES

Table

page

LIST OF FIGURES

Chapter I

INTRODUCTION

The end of World War II saw the beginning of the emergence of the United States as the world's leader in science and technology. This leadership was challenged on October 4,1957 when the Soviet Union launched the world's first artificial satellite, Sputnik I. Shortly after, Sputnik II, carrying the dog Laika, was launched on November 3, 1957. The success of these two Russian satellites shook the western world as the United States failed in her attempt to launch Vanguard, an artificial satellite on December 6, 1957. The United States eventually reasserted her supremacy by landing the first man on the moon a decade later.

Another "Sputnik" challenge is imminent. While the United States has remained the leader in technological development, some industries in other countries, notably Japan and West Germany, are surpassing their American counterparts by taking the initiative into implementing innovative technological manufacturing programs. The computer applications to support and control manufacturing operations is perhaps the most obvious example of such programs.

In many conventional single shift batch-type operations, the average part spends as much as 95 percent of its flow time waiting or moving between machines (MERCM77). Moreover, it is not uncommon for the value of work-in-process inventory to equal the investment in the plant and equipment. The advent of programmable automation, as in the case of the Flexible Manufacturing System (FMS), combines the flexibility of the conventional job shop and the lower variable cost of transfer lines. Productivity is enhanced through improved machine utilization, reduced work-in-process inventory, and effective material usages.

1.1 THE SOCIO-ECONOMIC MOTIVATION FOR PROGRAMMABLE AUTOMATION

Since the Industrial Revolution (circa 1770), machines have been replacing human labor in the handling, machining, inspection, assembly, testing, packaging, and programming manufacturing operations. Labor shortages, together with increasingly stringent government regulation on industrial safety provided the social impetus for industrial automation, while lower manufacturing costs provide the economic motivation.

Traditionally, the choice of automated manufacturing systems has been limited to stand-alone numerical control machines versus the automated transfer line. Frequently, the decision is based upon the annual demand and the product life cycle. For example, the unit manufacturing cost of a

V-8 automobile engine is approximately $25.00 if produced in large quantity on a transfer line. The cost would be $2,500.00 or higher if produced in a conventional job shop employing skilled labor (COOKN77).

A transfer line is a special set of production machines usually designed to manufacture only one part type in the most efficient manner. The transfer line has a high economic break-even point and is not viable until high production, on the order of 10,000 parts per year is reached. The job shop is preferred for low production, notwithstanding the fact that the relative cost per part produced may run 100 to 1 in a job shop when compared to a transfer line (HUTCG73).

The impact of this manufacturing dilemma is significant. The manufacturing sector accounts for 30 percent of the Gross National Product (GNP) in the United States. Forty percent of the manpower employed in manufacturing is involved in batch-type metal working. Fifty to seventy percent of parts manufactured by the metal working industry are in lots sizes of 50 pieces or less (COOKN75). To compound the problem, the cost of labor in manufacturing is rapidly rising. The Bureau of Labor statistics reports 7.2 percent per annum increase for 1970-1974 in labor cost with only a corresponding 4.7 percent increase in productivity (MERCM77).

The phenomenal growth in digital computing technology, coupled with the rapid decline in computer hardware/software cost, has made automation technology for mid-volume, mid-variety manufacturing economically viable. Instead of using hardwired devices, the operation sequence in a programmable automated manufacturing system is controlled by a program of instructions. Consequently, programmable automation has the capability to change the operation sequence according to different product configurations. It is most suitable for low to medium volume production runs with large variety of part types.

1.2 TAXONOMY OF AUTOMATED MANUFACTURING

Manufacturing systems are frequently classified according to the characteristics of part volume versus part variety. Table 1 shows a taxonomy of manufacturing systems arranged according to decreasing volume and increasing variety.

1.2.1 Volume Versus Variety

Like manufacturing systems in general, automated systems may be classified according to the dimensions of volume versus variety. High volume - small variety implies a high production rate with a reasonably long duration. Low volume - large variety means shorter production duration, but not necessarily at low production rate.

TABLE 1

Classification by Volume-Variety

Type	Volume	Variety
Continuous Production	High volume: operating at 24 hours/day, 7 days/week.	Single product at a time.
Mass Production	High volume	Small variety.
Flow Production	High volume.	Small variety.
Batch Production	Mid-volume.	Large variety.
Job Production	Possibly one of a kind	

Table 2 identifies the relevant range of the different automated manufacturing systems for the manufacturing dimensions of volume versus product variety. Transfer lines are most efficient for high volume sequential operations dedicated to the production of a small part varieties. Standalone NC machines have much greater versatility for part changes and are most appropriate in low-volume, high-variety production. The flexible manufacturing system is most cost-effective in the mid-volume-variety range, where 50-75 percent of parts manufacturing expenditure lies.

1.2.2 Extent of Automation

Industrial automation is the application of machinery to perform and control automatically and continuously a wide range of operations. This may include transporting of parts to machining stations, loading of parts onto machine tools, selecting and inserting the proper tools for each operation, establishing and setting operating speed and other machining parameters, controlling machine motion, sequencing tools, conditions and motions until all operations are completed on a part, and unloading the part.

The extent of automation is basically decided by economic considerations. However, other factors such as limitations in the properties of materials and limitations in the avail-

TABLE 2

Automation Within the Dimensions of Volume Versus Variety

(SOURCE: MODMH82)

Type of system and the degree of flexibility	Number of parts in family	Average quantity per batch
Low		
Transfer Line	1- 2	7,000 and up
Medium		
Dedicated FMS	3-10	1,000-10,000
Sequential or random FMS	4-50	50- 2,000
Manufacturing cell	30-500	30- 500
High		
Stand-alone NC	200 and up	1- 50

ability of machines may preclude complete automation. Partial automation should be so designed that future embellishments towards more complete automation do not entail expensive replanning or lead to obsolescence.

1.2.3 Fixed Automation of Standard Machines

Fixed automation refers to the class of manufacturing systems where the operation sequence is fixed by the equipment configuration. Fixed automation is generally inflexible and though the basic operation sequence may be simple, the system is made complex by the integration and coordination of all the required operations into a single piece of equipment. Some examples of fixed automation are transfer lines, automatic assembly lines, oil refineries, and certain chemical process (GROOM80).

Automation of single, standard machine tools is accomplished by using semi-automatic loading, which automates clamping, machine cycle control, and unclamping of parts. The system may be equipped with automatic parts-feeding devices and simple interlocks applied to ensure a desired sequence of clamp-machine-unclamp. Also, depending on the product size and shape, such devices as vibrating hopper feeds, rotary hopper feeds, and magazine feeds of special design may be applied (EVANC59). Gages may be incorporated to ensure consistent quality. This type of automation is applicable to standard milling machines, multiple spindle drilling and tapping machines, lathes, broaches, boring machines, grinding machines, and honing machines.

For automating two or more standard machine tools, the machines are equipped with inter-machine material handling devices. In-process storage may be needed to absorb the impact of a machine breakdown and processing time variability. Automation is accomplished by an integrated system of automatic feed and intermachine storage systems, automatic load/unloading devices, in-process gaging equipment, and possibly feedback control systems for tool settings (EVANC59).

1.2.4 Fixed Automation of Special Machines

Automation of special machines is generally intended for high production rate - high volume demand production. Special machines are inflexible and changes in the process to accommodate product changeovers are difficult and very costly.

Combination-operation special machines permit more than one basic metal working operation (e.g., milling, drilling, tapping, spot facing, etc.) which does not require movement or turning of the parts during processing. Probably the simplest form of combination-operation special machines is the double-ended machines where two machine tools are combined to perform a production operation. Double-ended machines are used for milling, drilling, chamfering, facing, centering, or boring. Limited flexibility is achieved through interchangeable fixtures, movable or interchangeable heads, speed changing devices, and adjustable stops.

In a line-index special machine, parts loaded in fixtures are indexed in a straight line to a machining position between two or more single or opposed machining heads. After processing, the fixtures are indexed back to the unloading position. Index-type special machines include: the trunnion-type special machines, the dial-type or rotary-index special machines, the center-column special machines, and the transfer lines (EVANC59).

In a transfer line, the parts/fixtures are indexed in a straight line from station to station between horizontal and vertical machines. A maximum number of operations can be performed at maximum rate.

1.2.5 Programmable Automation

Instead of being hardwired, the operation sequence in a programmable automated manufacturing system is controlled by a program of instructions. Since this system has the capability to change the sequence of operations according to different product configurations, it is suitable for low-volume production runs of a large variety of part types.

Considered to be the foundation of FMS, the first numerically controlled machine tool (NC) was introduced in 1955 at the Massachusetts Institute of Technology. NC systems effect the automatic control of machine motion through information stored on punched tape/cards. They exist in various degrees of complexity: from two-axis point-to-point drilling machines to five-axis milling machines where three linear motions and two angular rotations are continuously and synchronously controlled to produce sculptured parts with complex contours.

Numerical Control machines were immediately followed by machining centers equipped with Automatic Tool Changing (ATC) systems. These machining centers have the capability of storing, selecting, and changing cutting tools, all under the control of punched tape/cards. A single machine can bank a magazine of up to 100 or more tools, making tool changes possible in a few seconds.

With the further advancement of computer technology, Computer Numerical Control systems (CNC) were developed. Punched tape/cards information storage was replaced by magnetic-disk storage or computer memory storage. Part programs in CNC systems are easier to edit and alter than in NC systems. Moreover, CNC systems can perform many auxiliary functions such as pre-selecting the next tool required and having it ready when needed.

The next generation of computer-aided-manufacturing is Direct Numerical Control (DNC), a system connecting a set of NC machines to a common memory of part-program or machine-program storage, with provisions for on-demand distribution of part-program data to the machines. DNC systems have provisions for collecting and displaying data, editing part programs, operator instructions, operation schedules, and other data related to the NC process- such as status of the operation for management information and control. With the DNC systems, machining accuracy and repeatability are further enhanced.

The state-of-the-art in programmable automation is the flexible manufacturing systems. A flexible manufacturing system is a computer integrated automatic manufacturing system where an automatic material handling system(s) is used to move the part from workstation to workstation. FMS permits random access of work stations and provides management control through the joint implementation of numerically controlled machines and an automated material handling system. This flexibility and increased control make flexible manufacturing systems best suited for mid-volume manufacture of a variety of high precision parts, and/or where there are regular changes in specifications and volume requirements (HUTCG73).

1.3 THE FLEXIBLE MANUFACTURING SYSTEM

A flexible manufacturing system (FMS) is "a group of NC machine tools that can randomly process a group of parts having different process sequences and process cycles using automated material handling and central computer control to dynamically balance resource utilization so that the system can adapt automatically to changes in part mixes and levels of output" (KLAHT83). Other names for flexible manufacturing systems include: computer integrated manufacturing system (CIMS), computer managed parts manufacturing (CMPM), computerized manufacturing system (CMS), multi-station digitally controlled manufacturing system, variable mission systems, automated job shop, etc. (CHENP80).

Flexible manufacturing systems incorporate many individual automation concepts and technologies into a single system. These include automatic material handling system(s) between machines, NC machine tool(s) and CNC, computer control over the material handling system(s) and machine tool(s) (DNC), and group technology (GROOM79).

The machining stations in a flexible manufacturing system are typically versatile DNC machines. As such, each machining station can process several distinct operations and a typical part may visit each machining station more than once and may have numerous alternative routes. Three basic configurations of flexible manufacturing systems offer various degrees of flexibility: manufacturing cells, random or sequential systems, and dedicated systems. The manufacturing cell typically consists of NC machines clustered around a robot. In more advance systems, two or more manufacturing cells are integrated into a multi-clustered system by conveyors and storage towers. The sequential system produces parts of a single type at a time in essentially a sequential flow pattern. Its quick tool-change capability makes it possible to process a family of parts on the same system. In a random system, a variety of part types are launched simultaneously into the system. Lastly, the dedicated system consists of specialized metal-working machines generally designed to process a narrow range of heavy, bulky parts of up to 10 or 20 tons.

1.4 PURPOSE AND OUTLINE OF THE STUDY

The first flexible manufacturing system in the United States began commercial operation in 1973 at Roanoke, Virginia. Because of its brief history, literature on design and implementation of FMS has been scarce. The purpose of this research is to present some analysis concepts and design tools for designing and implementating a flexible manufacturing system.

Towards this seemingly broad objective, this study is organized as follows: The following chapter presents an overview of the problems encountered in designing and implementing FMS. The chapter outlines the problems investigated, and describes the scope and limitations of this study. Chapter three reviews some analytical and simulation models of flexible manufacturing systems, highlighting a simulation model generator developed in this study. Chapters four and five examine some problems on system design and implementation. In Chapter six, a dynamic decision approach to justifying FMS is developed and presented. Finally, Chapter seven summarizes the conclusions of the study and includes suggestions for future research.

Chapter II

DESIGNING AND IMPLEMENTING A FLEXIBLE MANUFACTURING SYSTEM

The major considerations in the design, operation, and control of the flexible manufacturing system fall under two major problem categories defined by the C. S. Draper Laboratory as the *Work Selection Problems*, and the *Hardware Selection Problems* (DRAPL81). These problems, together with the *economic justification problem*, provide the structural framework for the organization of this study.

Figure 1 shows a taxonomy of the problems under the three major categories of Economic Justification, Work Selection, and Hardware Selection. Under Work Selection are the problems of part pre-selection, part selection, batching, and work scheduling. Under Hardware Selection, the problems of machine selection/requirements planning, material handling system configuration, machine mapping/pooling, and real time operation must be resolved.

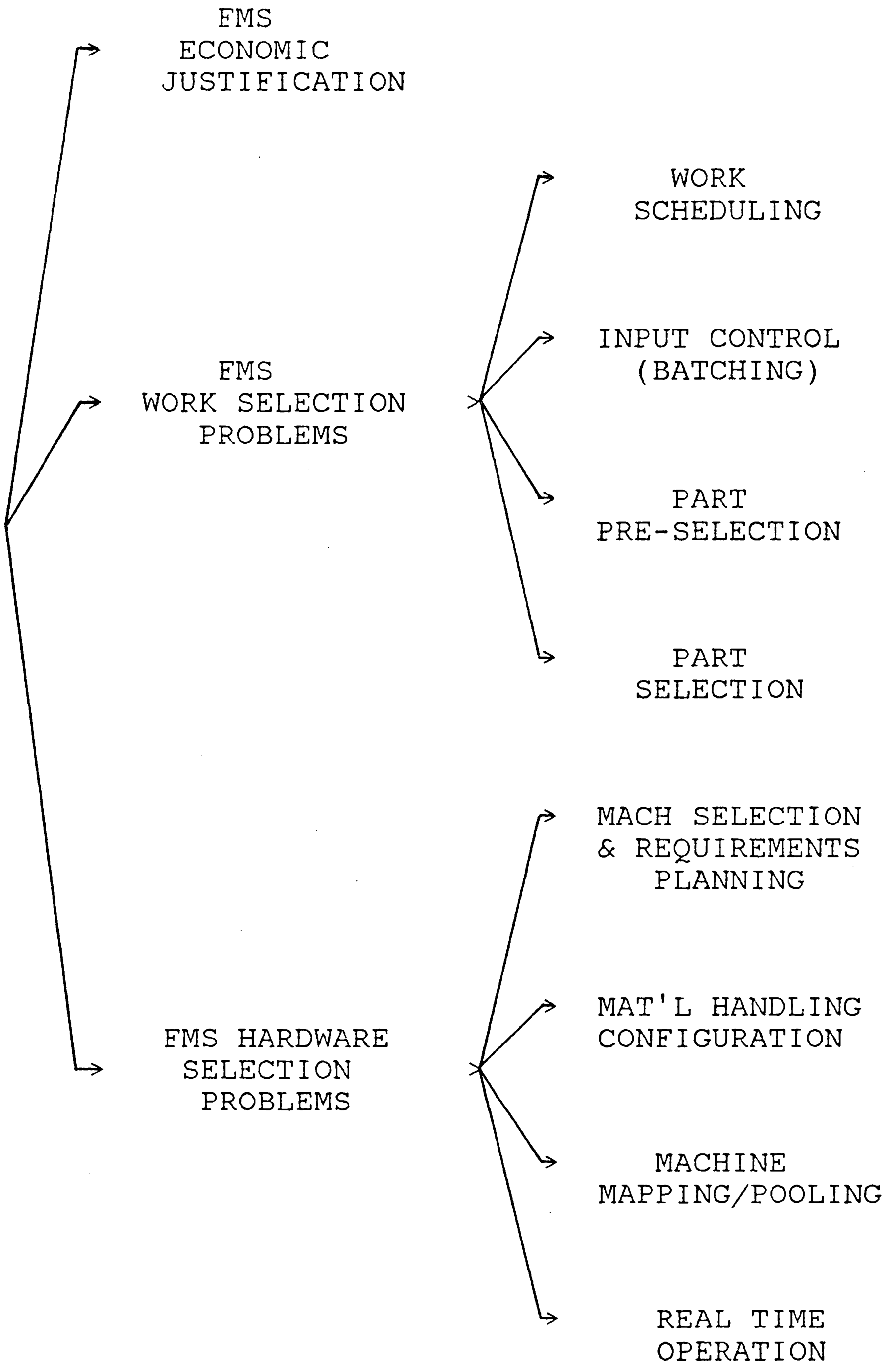


Figure 1: Design and Implementation Problems

2.1 SYSTEM SYNTHESIS

Two problems under Work Selection -- part pre-selection and part selection, and two problems under Hardware Selection -- machine selection/requirements planning and material handling configuration, constitute the system synthesis problem.

2.1.1 Part Pre-selection

The part pre-selection problem must be resolved at the design stage. Based on the geometry, weight, material, etc., of the parts, unsuitable (technologically infeasible or impractical) parts from a list of candidates for manufacturing by a generic flexible manufacturing system are screened out. The part pre-selection problem is not addressed in this study.

2.1.2 Part Selection

The problem of part selection is primarily resolved based on economic considerations. From a feasible set of parts for manufacturing by a generic flexible manufacturing system, those parts which are economically unattractive are deleted. An interactive computer model developed in this study provides an integrated approach for resolving this problem, together with the machine requirements planning problem described in the following subsection.

2.1.3 Machine Selection/Requirements Planning

The machine selection problem is resolved during the design stage. In the machine selection process, the following factors are taken into consideration: workpiece characteristics (including material, size, accuracy, and processing requirement), budget constraint, future changes in part variety/volume and required operating conditions, expected return on investment, and technological obsolescence, etc. The machine requirements planning problem involves an economic viability study (machine tool justification), and determining a complement most suitable for the intended workload. This study applies a queueing-network model called CAN-Q for resolving this problem simultaneously with the part selection problem.

2.1.4 Material Handling Systems and Auxiliary Equipment Selection

The machine tools versatility must be supported by corresponding versatility in the system for handling and routing materials. To be specified, subject to budget constraint, is a material handling system, fixturing, and auxiliary equipment that most efficiently and economically support a given production workload. Issues that must be resolved include the physical layout of the facilities (to minimize travel time, for ease in maintenance/cutting tool changes, spacial limitations, chip removal consideration, and access for maintenance and power supply), the track layout for the carriers, the type and capacity of shuttles, if any, and the capacity and location of buffer storages. This problem is not addressed directly in this study. However, a simulation model-generator developed in this research does provide a powerful design tool for configuring the material handling system and auxiliary equipment.

2.2 MACHINE MAPPING AND POOLING

Machine mapping is the problem of configuring tools and operations into machine group(s). Machine pooling or grouping is the problem of partitioning machines such that each group is able to perform a common set of operations, which may or may not overlap. An operation may be assigned to more than one machine, and some machines may have common sets of tools assigned to them. Stecke and Solberg (STECK77, STECK81, and STECK83) presented solution techniques for resolving the problem of machine mapping/pooling. The solution techniques consist of both heuristic approaches and mathematical programming formulations.

A set of non-linear integer programming models were presented by Stecke (STECK81 and STECK83). Five linearization techniques were introduced to linearize the quadratic terms in the models. However, the inherent difficulties of solving a non-linear integer model can be avoided if careful consideration is given to the proper definition of the decision variables. By defining a different set of variables, and using implied constraints, the machine mapping and pooling problem can be formulated as a zero-one linear programming problem. Large scale zero-one linear programming problems have been solved efficiently using implicit enumeration and the cutting plane techniques (see reference PADBM83).

2.3 BATCHING

The FMS is a unit processing system where parts are processed individually, not by batch. The FMS batching problem refers to the problem of partitioning the set of parts with planned requirements, into batches for simultaneous processing; subject to tool slot availability, material handling, and other constraints.

Batching limits the variety of parts flowing in the flexible manufacturing system. This is desirable for the following reasons:

1. The versatility of the individual machine tool is enhanced by pooling. Fewer classes of parts flowing in the system means less non-identical tools are required by the system at a time. Pooling of machines is maximized.

2. Less variety of parts means less diverse routing and therefore, less incidence of blocking/locking. Batching alters the flow pattern of parts in the system. At the extreme, if all the batches are homogeneous (one part type per batch), FMS is operated as a single-class job shop or as a programmable transfer line (non-backtracking). The control problem is therefore simplified.

This study explores the viability of operating FMS as a single class job shop (SCJS), or a programmable transfer line (PTL). The FMS is operated as a single-class job shop if each batch consists of parts of the same type (same routing). If the routings do not result in the backtracking of workflow, the single-class job shop FMS is a programmable transfer line.

Two features of FMS often cited in the literature -- the ability to random access any workstation and the reduction in tool change time -- are largely responsible for reduced machine idle time. The SCJS/PTL alternative reduces the flexibility available in the system. However, operating the system as a SCJS/PTL eliminates much of the complexity of sequencing and scheduling. The production idle time in an SCJS is attributable largely to line imbalance. Machine utilization, and therefore production capacity, is maximized by minimizing the balance delay of the production line. Therefore, a trade-off exists between better control in an SCJS/PTL, and the flexibility of a random FMS.

2.4 WORK SCHEDULING AND REAL TIME OPERATION

Scheduling involves the sequencing of jobs on each machine tool; subject to precedence constraints, deadlines, work-in-process storage restrictions, etc. The objective is to minimize the mean flow time, the average inventory level, the mean waiting time, the maximum waiting time, the mean lateness, the weighted mean flow time, the maximum job tardiness,or the maximum job lateness, etc.; or to maximize the minimum job lateness, the minimum job tardiness,etc. A comprehensive study of FMS scheduling may be found in references by Stecke (STECK77).

The complex control problems in flexible manufacturing systems arise from the multiplicity of control variables and the information requirement in real time operations of FMS. Real time operation is the problem of on-line control, and the problem of responding to machine outages for minimizing operations disruption.

Considerable research has been done in the areas of work scheduling and real time operation. Since no single close-form solution exists, most analyses were confined to simulation of specific systems. Although the work scheduling problem and the problem of real time operation are not addressed directly in this study, a simulation model-generator developed in this research provides a powerful design and analysis tool for resolving these problems.

2.5 ECONOMIC JUSTIFICATION

It is a concensus that the justification of flexible manufacturing systems is a complex, difficult, and often paradoxical process. This is true because the decision to install FMS is based on long range business strategy and it is not uncommon for FMS to be designed for uncertain objectives. In many instances, the parts that the system is supposed to make may not be conceived or designed yet. A dynamic decision approach to the justification of FMS is developed and presented in this study by imbedding queueing theory with simulation in a decision analysis framework.

2.6 DESIGN/ANALYSIS TOOLS

The methodology available for designing and analysis of manufacturing systems may be classified into the major categories of direct experimentation, simulation modelling, analytical solution, and numerical method.

A model is an abstract representation of a system, it is normally used to capture the essence, but not all the details of the system. Depending on the requirements of the analysis, the user must incorporate as much or as little detail as demanded. An advantage of simulation is the flexibility it provides the experimenter. Unlike analytical procedures, simulation modelling is tailor-made to the needs of the problem on hand, rather than restructuring the problem to fit available models. As the detail of the model increases, analytic solution procedures tend to become intractable.

Direct experimentation is costly, if feasible at all. Simulation conducts experiments on a model of a system, in lieu of direct experimentation. Days, months, and even years can be compressed in a matter of a few seconds of computer time. Moreover, system variables can be easily studied under more controlled conditions. The freedom in simulation model building also permits the users to obtain results in the form that is desired by the decision makers.

An obvious drawback of simulation modelling is that simulation, like direct experimentation, is experimental in nature. Statistical inferences based on observations are subject to statistical errors. Recent research in the area of numerical analysis suggest a compromise. Numerical procedures are approximations of analytical solutions. As such, much more detail can be accomodated than otherwise is possible with analytical solution methods. Approximation errors are controllable, while statistical errors are not. A trade-off exists between the greater flexibility of simulation modelling, and the control of approximation errors in numerical methods.

The chart in Figure 2 outlines the design/analysis tools developed in this study for resolving the problems encountered in designing and implementing FMS. These design/analysis tools encompass both analytical models (mathematical programming and queueing theory) and simulation.

2.6.1 Mathematical Programming

The machine mapping and pooling problem involves allocating operations and the required tools to machines, subject to technological and tool capacity constraints. This problem was examined by Stecke and Solberg in (STECK81) and (STECK83). A set of non-linear integer programming models were formulated and solved. This study presents an alternate model, formulating the problem as a zero-one linear programming problem.

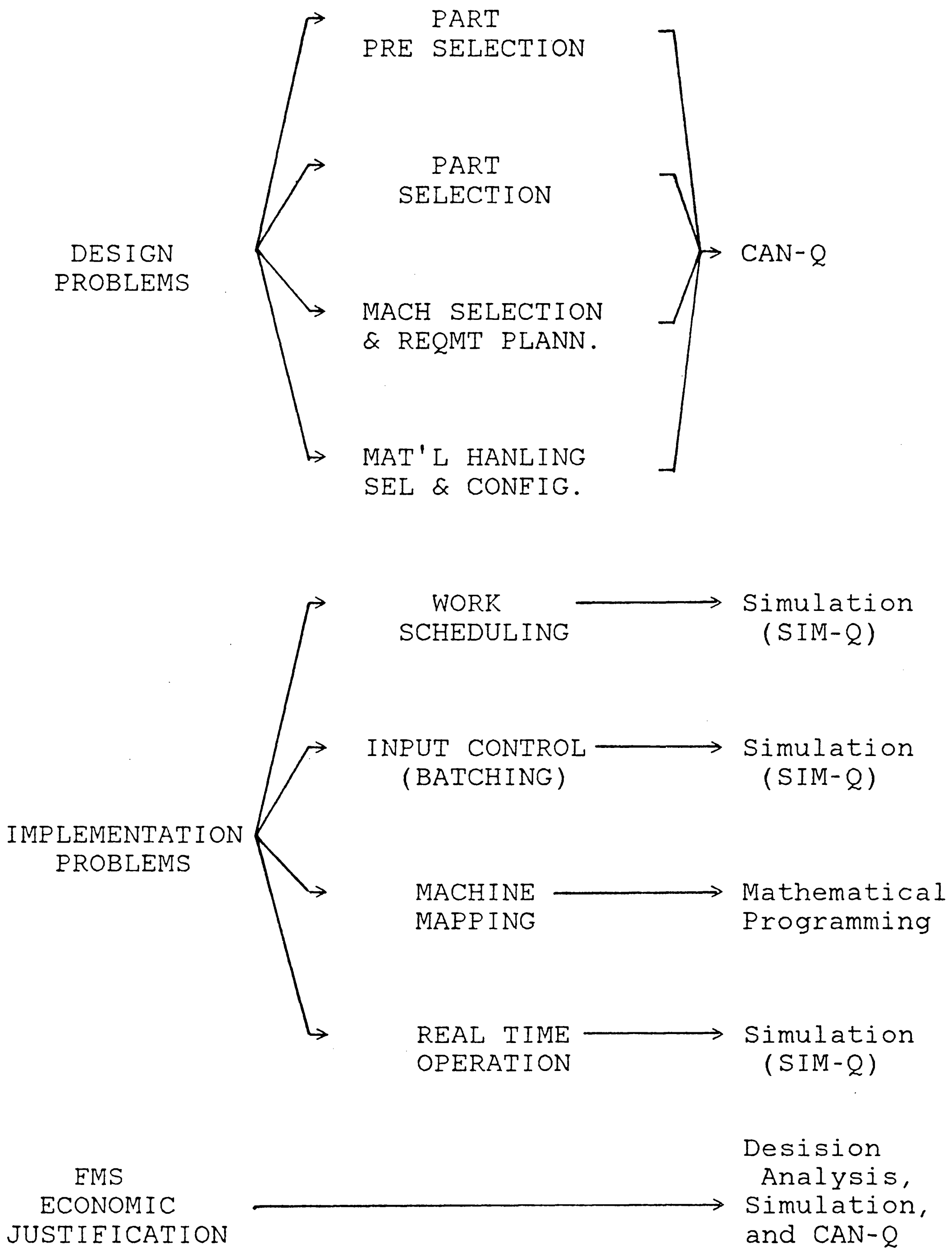


Figure 2: Design/Analysis Tools

2.6.2 Solberg's CAN-Q

CAN-Q is a performance evaluation package developed by Solberg for analyzing the workflow in production systems. The model is based on Jackson's queueing network (JACKJ57 and JACKJ63), which was extended by Gordon and Newell (GORDW67). The computational refinements for the model were introduced by Buzen (BUZEJ73). CAN-Q is described in (SOLBJ76 and SOLBJ77) and a user's manual is provided in (SOLBJ80). In this study, CAN-Q is imbedded in an interactive computer model for resolving the problems of product selection and machine requirements planning. CAN-Q is also applied in the economic justification model developed in this study by imbedding CAN-Q in a simulation model under a decision analysis framework.

CAN-Q is perhaps the most widely known analytical model for the planning of a flexible manufacturing system. It has been applied in analyzing different aspects of the design, implementation, and control of various types of system (STECK81). See for instance, references (LEIMF81), (STECK81), (SOLBJ80), (SECCO78), (KIMEJ78), and (WARDJ78). Inspite of many apparently unrealistic assumptions, CAN-Q has been found by many to be accurate in predicting the steady-state behavior of flexible manufacturing systems. Studies which attempt to validate the use of queueing network models through simulation and empirical studies include: (SOLBJ77), (HOREY78), (HUGHP73), (BUZEJ75), (GIAMT76), (LIPSL77) and (ROSEC78).

A statistical analysis of the robustness of CAN-Q in modelling an existing system is presented in Section 3.5.

2.6.3 SIM-Q

SIM-Q is the acronym for SLAM Integrated Model for Queueing-Networks. SIM-Q is developed in this study as an analysis tool for solving the material handling system selection, work scheduling, input control, and real time operation problems. SIM-Q exploits the structural simplicity of CAN-Q, while providing the user with the flexibility of modelling to fit specific system design and analysis requirements. The input data requirement of SIM-Q is patterned after CAN-Q, with some additional data necessary for modelling specific control variables and configurations.

SIM-Q creates simulation models that can be run on SLAM. The output report from SLAM consists of a general section followed by a section for the statistical results. The general section identifies the project title, the user's name, the month/day/year, the length of the simulation run, and the time when the statistical arrays were cleared. The statistical results include a summary of the flow time statistics, the waiting line (the number of parts waiting for each resource) statistics, and the resource utilization statistics.

2.6.4 Simulation Analysis

This study applies simulation extensively, mostly to illustrate the design tools/analysis concepts presented. With the exception of a few hypothetically contrieved examples, the setting of the simulation analyses centers primarily on the flexible manufacturing system at the Caterpillar Tractor Company in Peoria, Illinois. Henceforth, the system will simply be refered to as the Caterpillar system.

2.6.4.1 The Caterpillar System

The Caterpillar system consists of four OM3 omnimills (five axis machining centers), three OD3 omnidrills (four-axis drilling centers), two G/L vertical turret lathes, a DEA-coordinated inspection station, and 16 dedicated and undedicated (work-in-process storage) loading/unloading stations. The material handling system consists of two carts which run on a common rail located between two rows of workstations. The routing for the three part types processed in the system are shown in Tables 3, 4, and 5. All data used in this study regarding the Caterpillar system were taken from references (MAYER76), (LENZJ77) and (RUNNJ78).

2.6.4.2 The Performance Measure

The performance measure of interest in this study is the mean flow time. In a closed system, the mean flow time is inversely proportional to the output rate which is perhaps the single most important measure of the effectiveness of a flexible manufacturing system. This is very fortunate because for a sufficiently large sample size, the Central Limit Theorem assures that the sample mean is approximately Normally distributed.

2.6.4.3 The Method of the Batch Means

This research uses the batch means method for analyzing simulation results. A problem normally encountered in using the method of the batch means is that of autocovariance between the values at the end of one subinterval, and those at the beginning of the next interval. This variance can cause a positive covariance between batch means. Increasing the batch size decreases the covariance. However, the number of degrees of freedom also decreases so that the width of the corresponding confidence interval tends to increase.

TABLE 3

Job Routing for Part Type 1

Operation	Station	Time	Pallet	Frequency
1	19	1.00	1	1.00
2	8	18.50	1	1.00
3	10	4.00	1	0.20
4	19	10.00	1	1.00
5	17	10.00	2	1.00
6	4	23.30	2	1.00
7	7	14.50	2	1.00
8	10	3.50	2	0.05
9	17	10.00	2	1.00

TABLE 4

Job Routing for Part Type 2

Operation	Station	Time	Pallet	Frequency
1	16	10.00	3	1.00
2	3	37.90	3	1.00
3	6	26.60	3	1.00
4	10	6.60	3	0.33
5	16	10.00	3	1.00
6	15	10.00	4	1.00
7	9	36.80	4	1.00
8	5	38.40	4	1.00
9	7	6.30	4	1.00
10	2	38.40	4	1.00
11	10	4.00	4	0.20
12	18	10.00	4	1.00

TABLE 5

Job Routing for Part Type 3

Operation	Station	Time	Pallet	Frequency
1	18	10.00	4	1.00
2	4	26.30	4	1.00
3	1	34.80	4	1.00
4	10	32.00	4	1.00
5	7	15.10	4	1.00
6	6	7.10	4	1.00
7	10	2.50	4	0.05
8	15	10.00	4	1.00
9	14	10.00	5	1.00
10	8	20.60	5	1.00
11	7	3.40	5	1.00
12	10	6.00	5	0.20
13	14	10.00	5	1.00
14	13	10.00	6	1.00
15	9	12.20	6	1.00
16	6	5.80	6	1.00
17	10	4.80	6	0.20
18	13	10.00	6	1.00

This study follows Fishman's procedure for determining the batch size (PRITA79). Fishman's procedure is iterative. It starts with a batch size b = 1. The batch size is doubled repeatedly until the null hypothesis that the $X_i(b)$, for $i=1,2,...,I_b$ are IID is accepted. The test statistic is defined as:

$$Cb = 1 - \frac{\Sigma_{i=1}^{Ib-1} (X_{ib} - X_{i+1,b})^2}{2 \Sigma_{i=1}^{Ib} (X_{ib} - X_{Ib})}$$

Where I_b is the number of batches when the batch size is b. For large b, C_b is approximately the estimated autocorrelation coefficient between consecutive batches. If the $X_i(b)$'s are independent and Normally distributed, C_b has a mean of zero, a variance of $(I_b - 2)/(I_b - 1)$ and a distribution that is close to Normal for I_b as small as 8 (FISHG78). Therefore, a standard test using Normal tables can be applied. The covariance is assumed to be a monotonically decreasing function so that a one-tail test of size β can be applied. In particular, if $C_b \leq Z(2\beta)*(I_b-2)/(I_b -1)$, the null hypothesis is accepted.

The size of the test β deserves attention. A large β means a high probability of rejecting the null hypothesis H_o when it is true. In this study, the critical value of Z at 2β= 0.05, i.e., Z(0.05)= 1.645 is used for all independence tests.

It is noteworthy that correlation tests for independence are not fool-proof. Independence implies zero correlation, but zero correlation does not necessarily imply independence.

2.7 SUMMARY

Buzacott and Shantikumar describe the hierarchy of the control problems in a flexible manufacturing system as consisting of three levels (BUZAJ82) : Pre-release planning, Release or Input Control, and Operational Control. Pre-release planning corresponds to the problems of system synthesis and machine pooling/mapping. Input control determines the sequence and timing of the release of jobs to the system, these aspects are covered in the batching problem. The objective of operational control is ensuring movement between machines and deciding which job is to be processed next by a machine. Resolving the work scheduling and real time operation problems accomplish this goal.

Chapter three reviews analytical and simulation models available in the literature. SIM-Q, the input generator for creating simulation models of automated manufacturing systems developed in this research is presented; a statistical test of the robustness of CAN-Q in modelling an existing FMS is described.

The FMS system synthesis model developed in Chapter four, is an application of CAN-Q. The model is embodied in an interactive computer software providing an integrated approach to the part selection and machine requirements planning decision process. The decision variables include the number of hours of operation per year, the number of units for each machine type, the product families to be included in the production plan, and the material handling capacity. Also, a zero-one linear programming model is formulated for the machine mapping and pooling problem addressed by Stecke and Solberg in (STECK80).

Chapter five presents an investigation of the impact of the batching decisions on the performance of a flexible manufacturing system. The FMS batching decision is of paramount importance because it has immediate impact on the scheduling, balancing, and real time operation of a flexible manufacturing system.

The economic justification problem of FMS is examined in Chapter six. The approach combines queueing theory with simulation in a decision analysis framework.

Chapter III

DESIGN/ANALYSIS METHODOLOGIES

This chapter reviews some models for the planning and analysis of automated manufacturing systems. A "user-friendly" approach to simulation modelling is developed and introduced.

3.1 CAN-Q: FMS AS A CLOSED QUEUEING NETWORK

CAN-Q is a performance evaluation package based on the theory of queueing network. The model is that of a closed system with exponential service times and random routing.

Intuitively, the flow of parts in a pure job shop exhibits characteristics of a random walk, and applications of CAN-Q to such systems have been remarkably successful. In a transfer line, however, all parts follow the same sequential routing, there is no buffer storage and the "state" of the system is deterministic. Each server has exactly one part which are either in-process, or blocked from going to the next server. There is no queue to speak of since parts are transferred synchronously from station to station.

Between the well-behaved, no buffer storage transfer lines and the complex pure job shops, implementing CAN-Q requires an in-depth understanding of the theory of queueing networks.

Solberg claims that:

> Because CAN-Q requires so little information about a system, one may be skeptical at first about its abilities to capture the true effects of system variations. For example, most people would assume that a transfer line is so different from a job shop that no single model would adequately represent both. CAN-Q does encompass these two forms of production systems, and many others as well. It is able to do so by simply neglecting those aspects of a system which have only a small effect upon long run average behavior ... This conclusion is not based on any apriori theory, but rather on empirical evidence from testing and experimentation (SOLBJ80).

The claim that CAN-Q encompasses both job shop and flow shop production systems may be valid. CAN-Q is robust because the performance measures used are based on expectations which are unaffected by the covariances and dependencies among random variables in the system. A simple analogy of the robustness of CAN-Q is that of measuring gas pressure in an enclosed container. Charles' Law and Boyle's Law give a good measure of the external gas pressure as would an elaborate analysis of the Brownian motion of some zillions of molecules in the container.

Expectations are based on the law of large numbers. Because of high variability, the steady state long run expected value may not exist. It has been shown that for one specific system, it took a simulation run equivalent to 1,250 eight-hours shifts for the results of a Q-GERT simulator to approximate the results from CAN-Q (RUNNJ78). The Q-GERT model was modelled for the Caterpillar FMS system and has the exact set of assumptions/constraints as CAN-Q. Apparently, flexible manufacturing systems take a very long time to settle down and reach steady state . If equipment breakdowns and other variables are introduced into the model, the system may not ever reach steady state condition. As such, the output values from CAN-Q may be considered as as limiting values. An analogy is that of a gambler who doubles his stake each time he loses. Theoretically, the gambler may keep playing until he becomes a billionaire and then quit. The problem is, he may run out of betting money soon enough, before he ever become a billionaire.

The robustness of CAN-Q means that it is not sensitive to many control variables which, in the real world, have major impact on the performance of the system.

3.2 OTHER ANALYTICAL MODELS

Buzacott and Shanthikumar (BUZAJ80) formulated some simple queueing models for analyzing the production capacity of FMS. These models demonstrated the optimality of a balanced work load, the advantage of having diversed routing, and the superiority of common storage over local storage. Analytical models often overlook details of the systems being modelled. Many control variables cannot be analyzed.

3.3 SIMULATION MODELS FOR ANALYZING FMS

There are many simulation models of the Caterpillar system available in the literature. Among these are the CATLINE system (MAYER76), the GCMS simulator (LENZJ77), and the CAMSAM (RUNNJ78) simulator.

3.3.1 The CATLINE Simulator

Mayer and Talavage's CATLINE (MAYER76) is specifically written for the Caterpillar system. CATLINE was written in GASP IV. Except for a few assumptions (for example, a loader is always available at each load/unload station), the model follows closely the actual system. Unfortunately, data input is cumbersome and the time required to modify CATLINE to model specific systems is significant.

3.3.2 The GCMS Simulator

Developed by Lenz and Talavage (LENZJ77), a more general model written in GASP IV is the General Computerized Manufacturing Systems Simulator (GCMS). Its efficient modular structure and user-written subroutines makes the GCMS simulator potentially a very powerful universal tool for studying flexible manufacturing systems.

Many alternative FMS software and hardware configurations can be analyzed and evaluated using the GCMS simulator. Unfortunately, the user's manual of GCMS is not easy to follow and it is plagued with several obscure errors (RUNNJ78).

3.3.3 The CAMSAM Simulator

Another simulator, the Computer-Aided Manufacturing Systems Analysis Model (CAMSAM), was written in Q-GERT (RUNNJ78). Q-GERT is a process approach to simulation developed by Pritsker (PRITA77). The structure of Q-GERT is relatively simpler than most simulation languages. There are only ten types of nodes which can be combined to model a system.

CAMSAM was used in (RUNNJ78) to test various control variables such as the number of loaders, transporter speed and type, in-process storage capacity, levels of stockpiled inventory, etc. It is a model of the Caterpillar FMS as it was modeled by Lenz (LENZJ77) and Mayer (MAYER76). The model was run under similar assumptions as the GCMS and CATLINE models and the results were compared.

Compared to CAN-Q, CAMSAM has several advantages. One is that the simulator allows the analyst to limit the queue capacity of each workstation. Blocking and balking of parts arriving to a full queue can be modelled easily. Processing times need not be exponentially distributed and the routing of parts can be modelled to closely follow that of the physical system.

3.4 SIM-Q

Unlike CAMSAM, CATLINE, GCMS and other simulators of flexible manufacturing systems, SIM-Q is not a canned simulation model but a tool to facilitate ease in simulation modelling. The methodology follows the conceptual framework of CAN-Q and provides the user the flexibility to interface the network model with FORTRAN function subprograms. A user's guide for SIM-Q can be found in Appendix A. The cost of running a simulation model generated by SIM-Q, though relatively higher than running CAN-Q, is modest.

3.4.1 Input Data Requirement

Those users familiar with CAN-Q will find the structure and data requirement of SIM-Q to closely parallel those of CAN-Q. The major difference is the routing data required in CAN-Q, data which is not part of the SIM-Q input. The routing data is needed only when running the SIM-Q generated SLAM model.

The first data card is the LABEL card where the user enters his/her name, the project title, and the date of run. The second data card is the OPTION card specifying the length of the simulation, the time to clear statistical arrays, and the option code for releasing workstations. An option code of "0" releases the workstations immediately after processing while an option code of "1" holds the workstation until the workpiece is ready to be transported by the material handling device to the next station in the routing (i.e., both resources must be available).

The third data card is the PROBLEM SIZE card which defines the number of machine tools and inspection stations, the number of material handling devices, the number of dedicated loading/unloading stations, the number of in-process stations, the number of types of pallets/fixtures, the number of parts circulating through the system (system loading), and the number of part types. SIM-Q automatically models the loading/unloading operations as multi-resource operations if the number of types of pallets/fixtures is not zero.

RESOURCE SPECIFICATION cards are next, defining the name of the resources, the number of parallel servers of each resource, and the dispatching rule associated with each resource. Finally, PART NAME cards input the name of each part type. Figure 3 shows a card image of the data input requirement for running SIM-Q.

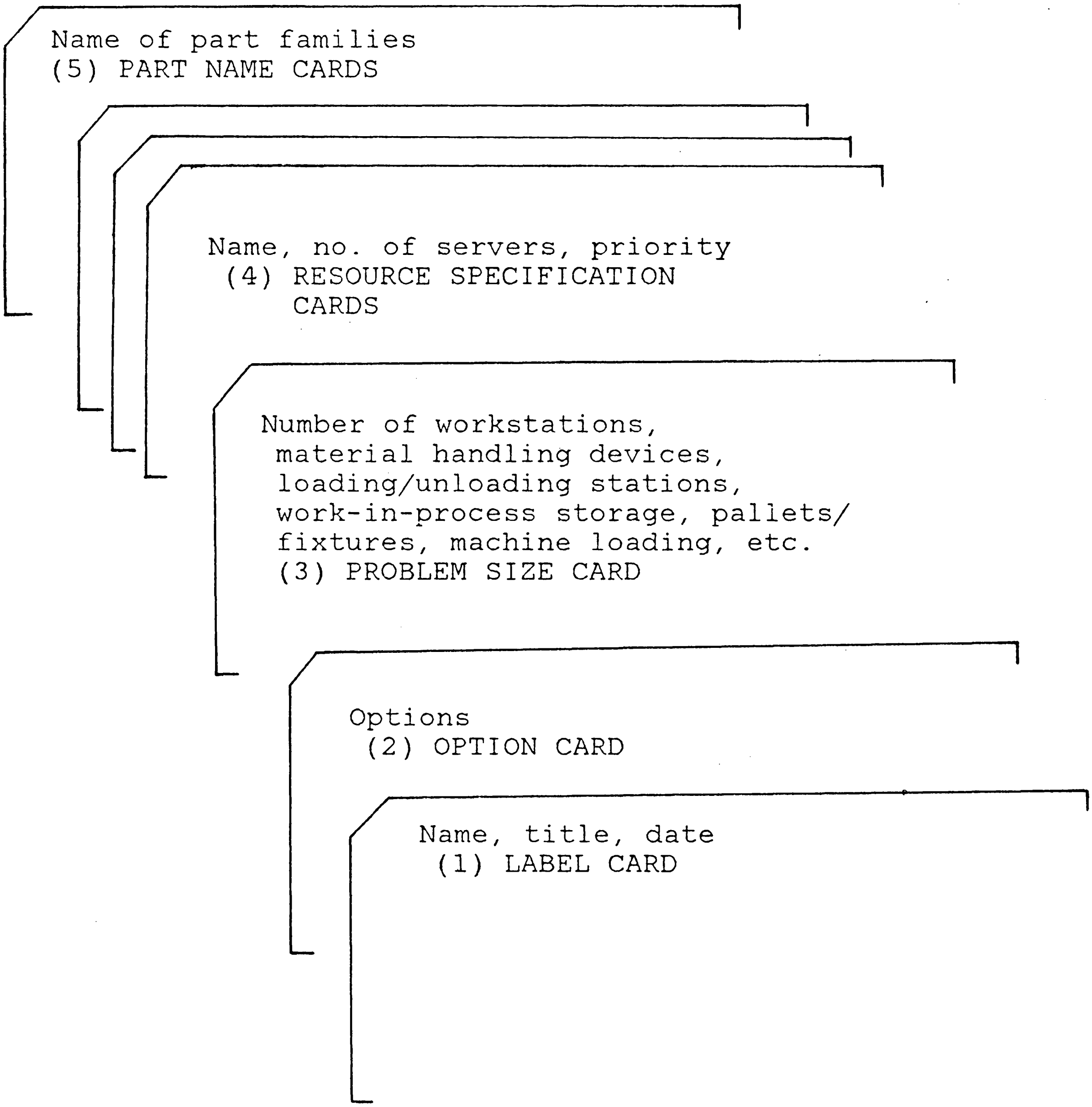


Figure 3: SIM-Q Input Data

3.4.2 Output

The output of SIM-Q is a self documented network "world-view" of a combined discrete-network SLAM simulation model. All the housekeeping specification statements needed to run the SLAM model are built-in to the network model.

As in CAN-Q, SIM-Q models the FMS as a closed network of queues (resources). It is, however, very easy to convert the model into an open network model with arbitrarily defined job arrival distribution by replacing two lines in the network program.

The simulation model created is divided into three segments. The first part of the program defines the attributes of the system and describes the purpose of the user-supplied subprograms. The second segment is the body of the simulation model. The resources modelled may include machine tools, inspection station(s), material handling system (henceforth referred to as the transporters), load/unload stations, temporary storage stations, pallets/fixtures, and manual loaders. Parts enter the system through a load/unload station and go through a sequence of operations before leaving the system. For a closed network model, the number of entities closed network is fixed, thus it is assumed that a new part is launched whenever a completed part leaves the system. If a workstation (machine tool or load/unload station) is not available when requested by a part, the part may be routed to a temporary storage station. Blocking occurs if either the transporter is busy or the temporary storage stations are full. A conceptual representation of the model is illustrated in Figure 4.

Many alternative FMS software and hardware configurations can be analyzed and evaluated using the simulation model generated by SIM-Q. The simulator has the capability to incoporate the following:

1. Various types of workstations (machining stations, loading and unloading stations, inspection stations, work-in-process storage, etc.),
2. Various material handling system configurations (robots, conveyors, automatic guided vehicles, etc.),
3. Manual operators,
4. Pallets/fixtures and other auxiliary equipment,
5. Flow of parts, rework and bypassing of an operation,
6. Machine breakdowns and repair, and
7. Various control variables such as sequencing rule, part routing, input control, and part launching.

The last part of the SIM-Q output specifies the length of the simulation run and the time for clearing the file and array statistics. The purpose of clearing statistics is to weed out erratic transient responses. If clearing of sta-

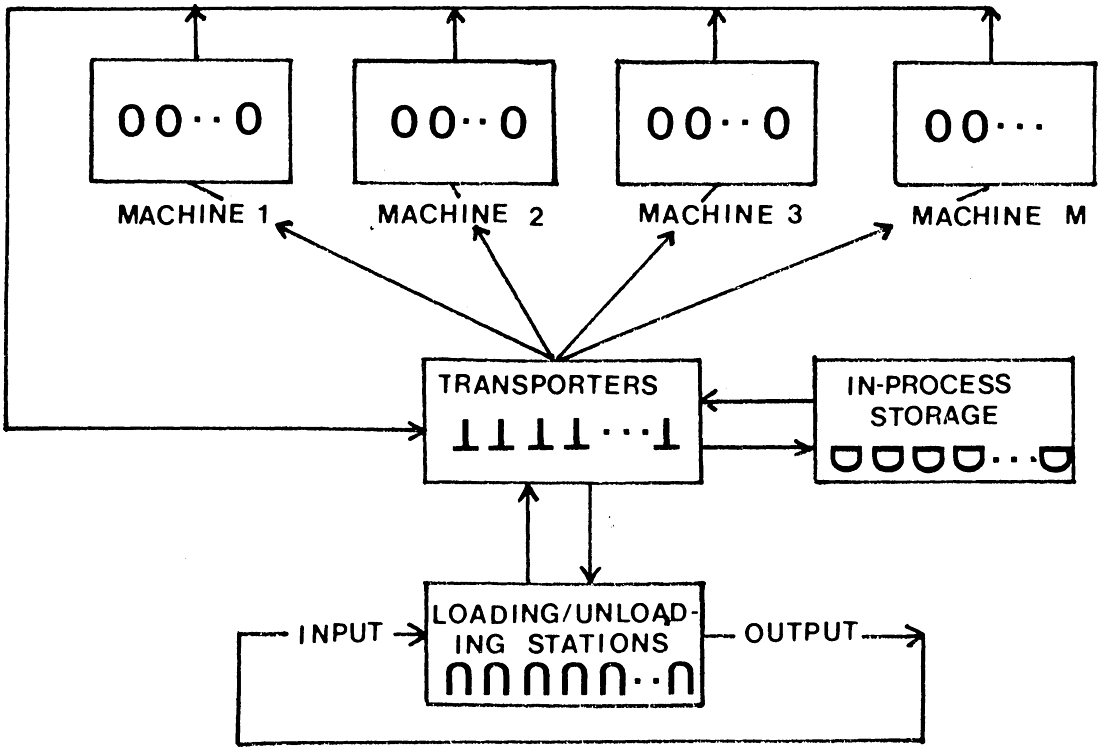


Figure 4: Conceptual Framework of SIM-Q

tistics is not desired, the user can specify the time of clearing to be zero. Figure 5 shows the logic flow of the network model.

3.4.3 Discrete World-View Interface

The SIM-Q generated network model may be interfaced with three user-supplied FORTRAN subprograms. The routing information, processing times, pallet/fixture requirement for each operation, probability of visits, location of resources, pool number (if some workstations are pooled), the probability of failure and time of repair (if reliability is to be incorporated into the model), the speed of the transporter(s), and other pertinent data are inputted through subroutine INTLC.

Subroutine EVENT is called to release a resource when it is no longer needed (service completion). Five events can be modelled. Events number 1, 2, 3, 4 and 5 release a pallet/fixture, a workstation, a transporter, a temporary storage station, or a manual loader, respectively.

Ten functions in function subprogram USERF provide the analyst the flexibility in modelling the network to fit specific simulation requirement. USERF(1) controls the part mix of the system by determining the type of an entering part. USERF(2) models the processing times, which may follow any user-specified probability density function.

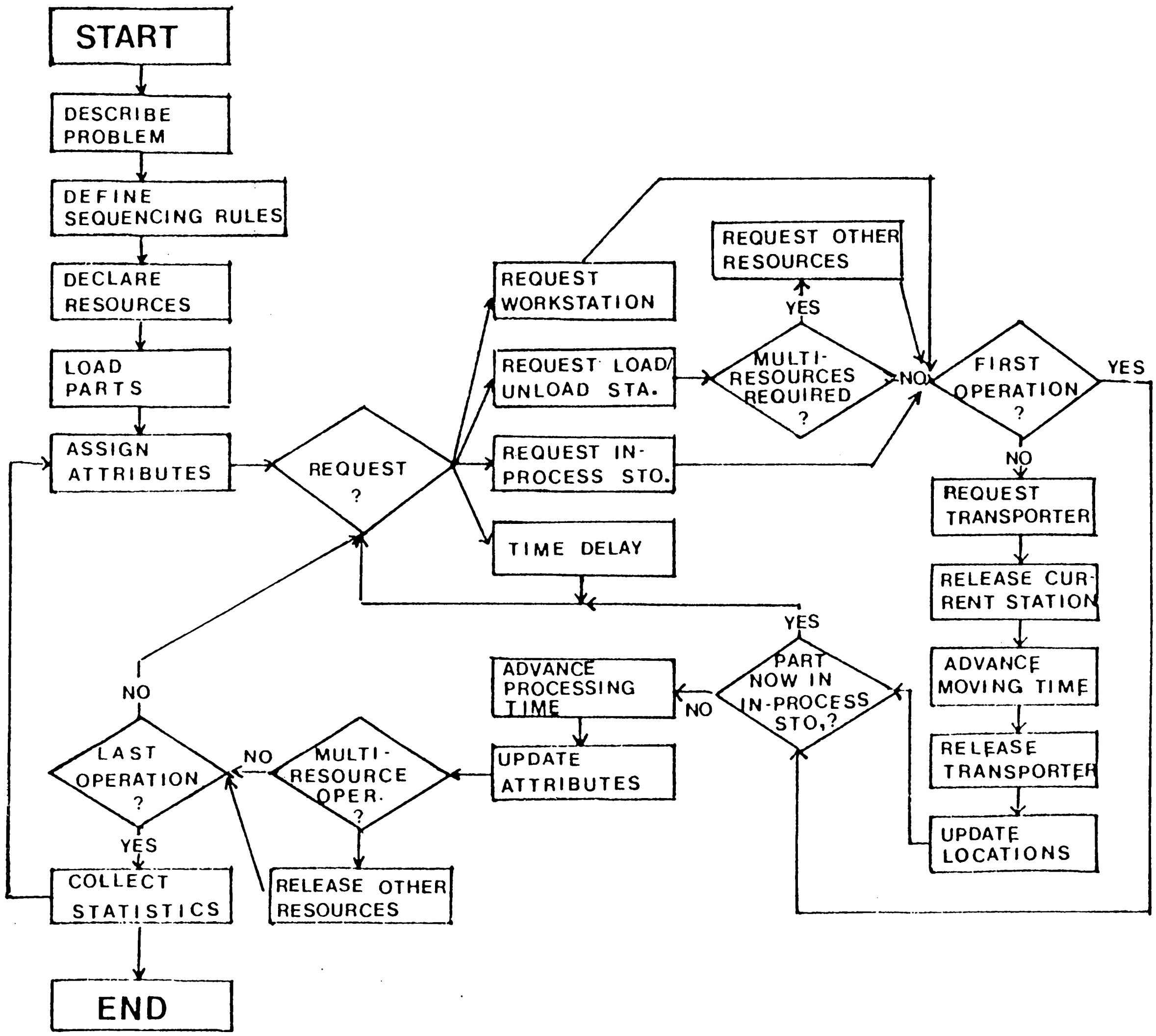


Figure 5: Logic Flow of the SIM-Q Network

USERF(3) returns the time it takes a material handling device to fetch and deliver a part from its current location to its next location. The simulation model generated by SIM-Q can keep track of the location of all resources (machine tools, inspection stations, material handling devices, loading/unloading stations, and temporary storage stations) and parts in the system. The layout of the system may be divided into zones and the location of each entity/resource designated by a zone number. The time required to transport (the transport time) a part between stations can be computed based on the current zone number of the part, its destination zone, the cart speed, and the availability of the cart. Modelling of traffic interference is also possible.

USERF(4) simulates whether a particular operation will be carried out or bypassed. Sometimes, a particular operation may be bypassed. An example of this is the in-process inspection operation where 100 percent inspection is not required. The user can input the probability of visit through Subroutine INTLC and USERF(4) returns a value "0" whenever an operation is bypassed.

USERF(5) returns the current location of a part while USERF(6) returns the priority code of the part whenever the part is requesting for a resource. The priority codes may be computed or inputted through Subprogram INTLC. USERF(7) returns the pallet type required for the current operation. If the current pallet type is different from the required pallet type, a new pallet type is requested. SIM-Q automatically models this as a multi-resource operation where a manual operator will also be requested.

USERF(8) identifies the workstation a particular part will be visiting next, and USERF(9) returns the location of this workstation. If the current operation is the last operation, USERF(8) returns a value of zero. This is accomplished by adding a dummy operation at the end of each routing. The function returns a value "0" if all required operations have been performed. The routing information is inputted through Subroutine INTLC. Routing may be deterministic or random. A part can be routed to an in-process storage if the requested workstation is not available. Finally, USERF(10) returns the resource number of the cart selected to carry the part. Decision criteria such as proximity (the distance between the transporter and the part), availability, and queue length (number of parts waiting to be moved) can be modelled easily.

A comprehensive listing of the user-supplied subroutine EVENT and function USERF can be found in the user's guide in Appendix A. Table 6 lists some USERF functions available in the user's guide.

TABLE 6

USERF functions available in the SIM-Q User's Guide

USERF(1): Launching of Parts	a. Equal proportion of parts. b. Unequal proportion of parts. c. Part type determined by the time of entry to the system.
USERF(2): Processing Times	a. Standard probability distribution functions. b. Machine breakdown modelled.
USERF(3): Material Handling Process	a. Standard probability distribution functions. b. Transport times determined by the distance travelled and the speed of the material handling device. c. Traffic interference modelled.
USERF(4): Visit Freq.	a. Bypassing of operation. b. Rework.
USERF(5) Current Location of Parts	
USERF(6): Dispatching Priority Codes	a. Seniority rule. b. SPT rule. c. Random sequencing. d. Shortest remaining processing time. e. Number of remaining operations. f. Cycle time of the next station. g. Composite rule.
USERF(7) Pallets/Fixtures Required	
USERF(8): Routing	a. Random routing. b. Deterministic routing. c. Multi-resource operation modelled. d. Pooled machines.
USERF(9) Next Location of Parts	
USERF(10): MHS Selection	a. Based on availability, proximity and queue length.

3.5 THE ROBUSTNESS OF CAN-Q IN MODELLING THE CATERPILLAR SYSTEM

This section presents a statistical test of the robustness of CAN-Q in modelling the Caterpillar system. The validation process involves the test of a hypothesis comparing the mean flow times derived from CAN-Q, with the simulation output of SIM-Q. The equations for computing the confidence intervals, for hypothesis testing, for the test of independence of batch means, for stratified sampling, and other details of the statistical treatment of simulation were taken from references (PRITA79), (KLEIJ74), and (LAWAV82).

3.5.1 The Caterpillar System

A simulation model of the Caterpillar system was generated using SIM-Q. The program listing can be found in Section A.14 of Appendix A. Two simulation runs were executed. The first simulation run assumes zero transport times while the second simulation run assumes an average cart speed of 10.0 zones per minute. Each simulation was run for a length of 25,000 minutes, equivalent to approximately 52 eight-hour shifts. Following Solberg's recommendation (SOLBJ80), the number of parts circulating in the system was assumed to be 16, one less than the total number of machining stations, inspection station, and dedicated loading/unloading stations.

Plots of the flow times generated from the simulations were examined to determine the truncation point for the steady state analysis. For the two simulations, it was decided to delete as transient responses the observations generated during the first 600 minutes of simulation.

In the first simulation, the estimated mean flow time is 303.13 minutes; and the variance is 7.75, with 34 degrees of freedom. In the second simulation, the estimated mean flow is 342.48 minutes; and the variance is 21.79, with 28 degrees of freedom. The 90 percent confidence interval for the mean flow time is [298.55 ; 307.71] minutes for the first simulation, and [334.54 ; 350.42] for the second simulation.

Using the same routing data, CAN-Q was executed for various assumptions of the expected value of the time for transporting a part between stations (transport times). The number of parts circulating in the system was also assumed to be 16. The computed system performance measures are summarized in Table 7.

The expected value of the transport times is an input in CAN-Q and must be approximated. In the second simulation run, 1,157 parts corresponding to 15,039 operations were

TABLE 7

Performance As a Function of Average Transport Time

Average Transport Time	Mean Time In the System			
	Processing	Travelling	Waiting	Total
0.00	178.42	0.00	189.66	368.08
0.10	178.42	1.11	189.04	368.57
0.20	178.42	2.22	188.42	369.05
0.40	178.42	4.43	187.20	370.05
0.50	178.42	5.54	186.60	370.56
0.75	178.42	8.31	185.16	371.89
1.00	178.42	11.08	183.82	373.31
1.25	178.42	13.85	182.61	374.88
1.50	178.42	16.62	181.59	376.63
1.60	178.42	17.72	181.25	377.39
1.67	178.42	18.50	181.03	377.95
1.75	178.42	19.39	180.81	378.61
1.885	178.42	20.90	180.50	379.82
2.00	178.42	22.16	180.33	380.90
2.50	178.42	27.69	180.62	386.74
3.00	178.42	33.23	183.34	394.99
3.50	178.42	38.77	189.58	406.77
4.00	178.42	44.31	200.50	423.23

completed in 24,957.98 minutes of simulation. The average utilization of the two carts is 56.8 percent. Therefore, the average time per transfer between workstations is 2(0.568)(24957.98)/15039 = 1.885 minutes. The null hypothesis asserts that the mean flow times computed in CAN-Q are statistically equal to the mean flow times estimated using the SIM-Q model. The test statistic computed for the first simulation run (zero transport times) is equal to 23.33, with 34 degees of freedom. Therefore, the null hypothesis is rejected.

Similarly, assuming an average transport time of 1.885 minutes, the test statistic computed for the second simulation run is equal to 8.00 (28 degrees of freedom), greater than the critical value of 1.645. Therefore, the null hypothesis is also rejected. However, it is encouraging to note that the mean flow time computed from CAN-Q is within 10 percent of the point estimate for the mean flow times derived from simulation.

As mentioned earlier, the expected value of the transport times is an input in CAN-Q and must be approximated. It is noted from Table 7 that the computed mean flow times vary from 368.08 minutes for zero transport times to 423.23 minutes if the average transport time is 4.00 minutes. Clearly, choosing the appropriate average transport time is not a trivial problem.

3.5.2 Blocking and Locking

In the Caterpillar System, a part is routed to a temporary loading/unloading station whenever the next station in the routing is not available. The temporary loading/unloading stations have finite capacity. Blocking and locking are possible. CAN-Q assumes no blocking and may tend to overestimate the production capacity of the system. Moreover, CAN-Q does not recognize the difference in flow pattern. Whether a part goes through stations M1 - M2 - M3, or M3 - M2 - M1, or M2 - M1 - M3, etc., the system performance computed from CAN-Q is the same.

The same simulation model was executed, except for exponentially distributed processing times. The diversity of routing, the element of uncertainty introduced in the processing times, compounded by modelling the loading/unloading operations as multi-resource operations, caused the system to lock after the 1,136.19th minute.

3.5.3 Sequencing Rules

In the preceeding section, the sequencing rule was based on the expected cycle time of the next station in the routing. In this section, two simulation runs were executed on the same simulation model, except for the first-in-first-out (FIFO), and the "Seniority" (time in the system) sequencing rules. Both runs were executed for a length of 25,000 minutes. The confidence interval for the mean flow time is [335.69 ; 351.97] minutes, for the first-in-first-out rule; and [337.99 ; 348.53] minutes, for the "Seniority" rule. The confidence intervals are very close to that estimated for the "cycle time" sequencing rule. The test statistics computed is 7.72 (28 degrees of freedom) for the FIFO rule; and 11.78 (29 degrees of freedom) for the "Seniority" rule. Although the corresponding null hypothesis can not be accepted, the mean flow times computed from CAN-Q are within 10 percent of the point estimates derived from simulation.

3.5.4 Layout and System Configuration

In CAN-Q, no transport time is involved in moving a part to, or from the in-process storage. A part is routed to a local in-process storage station if the part's next station is busy, and when the station becomes idle, the waiting part can be processed immediately. In the Caterpillar system, the part is either routed to a central storage station, or remains in the current station of the part when the central storage stations are full (blocking). A station is always idle while a part is in the process of being transported to the station.

To evaluate the effect of this assumption, another simulation model of the Caterpillar system was generated using SIM-Q, with all the assumptions of CAN-Q except for centrally located in-process storage. The model can be found in Section A.12 of Appendix A. The transition probabilities, average processing times, and the average number of operations needed to complete an item were taken from the "Input Summary Report" of CAN-Q and used as input for the simulation model. The average transport time was arbitrarily assumed to be 1.67 minutes. The sequencing rule is first-in-first-out.

This model was executed for a length of 25,000 minutes, equivalent to approximately 52 eight-hour shifts. Plots of the flow times simulated were examined to determine the truncation point for the steady-state analysis. Statistics generated during the first 600 minutes of simulation were discarded.

The 90 percent confidence limits for the mean flow time is [386.45 ; 404.71] minutes. The computed t-statistic is

3.18, with 61 degrees of freedom. Therefore the null hypothesis is rejected. The mean flow time computed from CAN-Q is statistically different from the results of the simulation.

It is noted that the difference between the point estimate of the mean flow time from the simulation and the mean flow time computed in CAN-Q is only 17.63 minutes, a difference of less than 5 percent. The average number of operations needed to complete a part is 11 and the mean transport time is 1.67 minutes. If an adjustment of 18.37 minutes were added to the mean flow time computed from CAN-Q, the computed t-statistic would have been equal to 0.149 and the null hypothesis would be accepted.

3.5.5 Deterministic Routing

Now consider a model of the Caterpillar system with all the assumptions of CAN-Q, except for deterministic routing. The model can be found in Section A.13 of Appendix A. The loading/unloading operations are modelled as single-resource operations (thus eliminating the complications of pallet/ manpower availability) and temporary local storages are assumed to be adequate that blocking and locking will not occur.

The model was executed for a length of 25,000 minutes. Observations during the first 1000 minutes of simulation were discarded. The 90 percent confidence interval is (361.10 ; 378.46) and the computed t-statistic is 1.548. Therefore the null hypothesis is accepted.

The result of this section is very encouraging. It suggests that at least for a three-part-types system, the routing need not be "random" for the results of CAN-Q to be valid.

3.5.6 Probability Distribution of Processing Times

The exponentially-distributed-processing-times assumption is generally believed to be the most restrictive among the assumptions in CAN-Q. This subsection tests the performance of CAN-Q assuming various probability density functions for the processing times.

Eight runs were executed on the same simulation model as Section 3.5.5. The first simulation assumes constant processing and transport times while the second simulation assumes exponentially distributed processing and transport times. The third, fourth, and fifth runs assume uniformly distributed processing times and exponentially distributed transport times. The three runs differ in the extend of variability in processing times. The third run assumes the variance to be equal to the mean. The fourth run doubles the variance while the fifth run assumes the variance to be three times the mean. The sixth, seventh, and eighth runs are similar to the third, fourth, and fifth simulation runs except for normally distributed processing times.

In each case, the simulation length is 25,000 minutes. The sequencing rule is first-in-first-out. The expected value of transport times is assumed to be 1.67 minutes. Table 8 summarizes the statistical analysis of the eight simulation runs. CAN-Q was found to perform poorly in all cases other than that of exponentially distributed processing times. Table 9 shows the computed mean flow time from CAN-Q to be consistently higher than the point estimates of the mean flow time derived from simulation. The discrepancy tends to increase with decreasing variability in processing times.

3.5.7 Pure Flow Shop

Intuitively, the greater the variety of parts in the system, the more diverse is the routing and the more likely it is that the flow of parts exhibits characteristics of a random walk. Section 3.5.5 demonstrated the robustness of CAN-Q in modelling a three-part-types random FMS. It has been shown for exponentially distributed processing times

TABLE 8

Robustness of CAN-Q In Modelling Non-Exponential Processing Times

Run	Density Function of Processing Times	Computed t-statistic	Null Hypothesis
1	Constant	92.16	Reject
2	Exponential	2.77 0.64	Reject Accept*
3	Uniform I	48.34	Reject
4	Uniform II	61.18	Reject
5	Uniform III	38.88	Reject
6	Normal I	51.01	Reject
7	Normal II	33.51	Reject
8	Normal III	47.25	Reject

* adjusted CAN-Q mean flow time used.

TABLE 9

Point Estimates of the Mean Flow Time (Various Processing Time Distributions)

Run	Density Function of Processing Times	Point Estimate	Difference vs CAN-Q
1	Constant	287.18	- 24.0%
2	Exponential	392.30	- 3.8 - 1.0*
3	Uniform I	296.34	- 21.6
4	Uniform II	298.42	- 21.0
5	Uniform III	306.37	- 18.9
6	Normal I	295.23	- 21.9
7	Normal II	303.18	- 19.8
8	Normal III	305.05	- 19.3

* adjusted CAN-Q mean flow time used.

that the routing need not be random for the results of CAN-Q to be valid.

This subsection investigates the robustness of CAN-Q in modelling a sequential flow system. In particular, each of the three part types is processed as if the Caterpillar system is operated as a programmable transfer line. Moreover, the original routings are alterered to eliminate backtracking of workflows.

The line configuration for Part Type-1 has some degree of backtracking. If the two inspection operations are combined, a uni-directional flow pattern would result. Similarly, combining the inspection operations for Part Type-2 eliminates the backtracking workflows while combining the inspection operations and merging two drilling operations in the routing for Part Type-3 result in a uni-directional flow pattern. The embellished routings are shown in Tables 10, 11, and 12. The processing times are essentially the same as the original data, the only difference being that backtracking workflows have been eliminated.

Table 13 shows the mean flow time computed from CAN-Q. Assuming deterministic processing times, three simulation runs were executed, one run for each part type. The number of parts in the system was assumed to be 22, equal to the number of stations for each line(4 Omni-mills, 3 Omni-

TABLE 10

Embellished Job Routing for Part Type 1

WORK STATION	SERVICE TIME	NUMBER OF STATIONS ASSIGNED
MANUAL	1.000	3
LATHE	18.500	2
MANUAL	10.000	3
MANUAL	10.000	3
MILL	23.300	4
DRILL	14.500	3
INSPEC	0.975	1
MANUAL	10.000	3

TABLE 11

Embellished Job Routing for Part Type 2

WORK STATION	SERVICE TIME	NUMBER OF STATIONS ASSIGNED
MANUAL	10.000	3
MILL	37.900	2
DRILL	26.600	1
MANUAL	10.000	1
MANUAL	10.000	3
LATHE	36.800	2
DRILL	38.400	1
DRILL	6.300	1
MILL	38.400	2
INSPEC	3.000	1
MANUAL	10.000	3

TABLE 12

Embellished Job Routing for Part Type 3

WORK STATION	SERVICE TIME	NUMBER OF STATIONS ASSIGNED
MANUAL	10.000	2
MILL	26.300	2
MILL	34.800	2
DRILL	15.100	1
DRILL	7.100	1
MANUAL	10.000	2
MANUAL	10.000	2
LATHE	20.600	1
MANUAL	10.000	2
MANUAL	10.000	2
LATHE	12.200	1
DRILL	9.200	1
INSPEC	34.285	1
MANUAL	10.000	2

drills, 2 vertical turret lathes, 1 inspection station, and 12 manual loading/unloading stations).

The same simulation model as in Section 3.5.6 was executed. The mean flow times estimated from simulation appear to be very close to the production rates computed in CAN-Q, even without adjustment. The adjusted CAN-Q mean flow time is 218.85, 624.75, and 778.40 minutes, for Line-1, Line-2 and Line-3, respectively. After deleting the statistics during the transient period, the mean flow times estimated from simulation are practically constant. The percent difference between the simulated mean flow times and the adjusted mean flow times computed from CAN-Q for the three lines is 1.35 percent, 0.47 percent, and 1.58 percent, respectively.

In spite of the assumptions on processing times and the routing mechanism, the results of the simulation are surprisingly close to the results obtain in CAN-Q. However, it has been shown earlier that the performance measures obtained by assuming deterministic processing times are significantly different from the results obtain from assuming exponentially distributed processing times. The implication is that if the results of CAN-Q holds for the deterministic case, it may not hold for the case of exponentially distributed processing times.

TABLE 13

Flow Times of the Caterpillar System As a Programmable Transfer Line (CAN-Q)

Average Transport Time	Mean Time In the System Processing	Travelling	Waiting	Total
Line 1 (Part Type-1)				
0.00	88.27	0.00	115.46	203.73
1.67	88.27	13.36	103.86	205.49 218.85*
Line 2 (Part Type-2)				
0.00	227.40	0.00	378.11	605.51
1.67	227.40	16.70	363.95	608.05 624.75*
Line 3 (Part Type-3)				
0.00	219.48	0.00	535.24	754.72
1.67	219.48	23.38	512.16	755.02 778.40*

* adjusted for centrally located in-process stations.

3.6 SUMMARY

This chapter presented an introduction to SIM-Q. Modelling a flexible manufacturing system using SIM-Q is not much more difficult than modelling in CAN-Q. The user's guide provides a comprehensive listing of FORTRAN subprograms to enhance the capability of the network model. The cost of running a simulation model generated by SIM-Q is modest. Creating a 300-line SLAM network program on an IBM 370 model 158 requires only a fraction of a second of CPU while executing a SLAM model for a typical simulation experiment (the Caterpillar system analyzed in this study) requires approximately three minutes of CPU.

A statistical evaluation of the robustness of CAN-Q for a variety of system configuration (based on the Caterpillar system) was presented. The random-routing assumption is generally suspected to be a major factor in the validity of CAN-Q. The case studies in this chapter showed that the results of CAN-Q hold for a three-part-types system with fixed routing and that CAN-Q is surprisingly accurate in modelling a pure flow shop with deterministic processing times.

By assuming exponentially distributed processing times, CAN-Q may underestimate the output capacity of the system. As it turned out in a case study, the error may be compensated by the assumption of infinite-capacity local in-process storage. The results tend to suggest that in aggregate planning where exact figures are not required, CAN-Q does provide an expedient tool for system design. However, since the investigator has no control over the compensatory effects of the errors, the application of CAN-Q requires prudent judgement.

As an analytical tool, CAN-Q is insensitive to many control variables which have major impact on the performance of the system. This precludes its application in analyzing various control strategies.

Chapter IV

SYSTEM SYNTHESIS, MACHINE POOLING AND MAPPING

In this chapter, Solberg's CAN-Q is imbedded in an interactive computer model for resolving the machine requirements and part selection problems. Also, an alternative formulation of the machine mapping and pooling problems outlined in reference (STECK80) is presented.

4.1 THE SYSTEM SYNTHESIS MODEL

The FMS system synthesis model developed in this study is embodied in an interactive computer software model providing an integrated approach to the part selection and machine requirements planning decision process. The model assumes the following:

1. The part pre-selection problem has been solved. A list of FMS-compatible part families has been identified.
2. The machine selection problem and the material handling system selection problem have been solved. The corresponding process plan, machine types, machinability data, and cycle times are known.
3. The production requirements have been estimated.

4. The investment costs of the machines of each type are known.

4.1.1 The System Synthesis Model

A logic flow chart of the model is shown in Figure 6. An executive routine that calls CAN-Q prompts the decision maker for the input data which include the names of the part families and machine types, the routing and processing times, the production requirement, the acquisition costs of the machines, and the "lower bound" of the machines required. The CAN-Q - based subroutine then proceeds to preprocess the routing and production requirement data, computes the relative workload of each machine type, and identifies the bottleneck operation. The production capacity is computed and the decision maker is then asked:

1. whether machines are to be added to the bottleneck operation, or
2. one or more product families are to be dropped/added to the production plan.

At each iteration, the production plan, the investment cost, the number of machines for each machine type, and the production capacity are updated. The program stops when the system has reached the desired production capacity, or when the investment requirement exceeds the available budget.

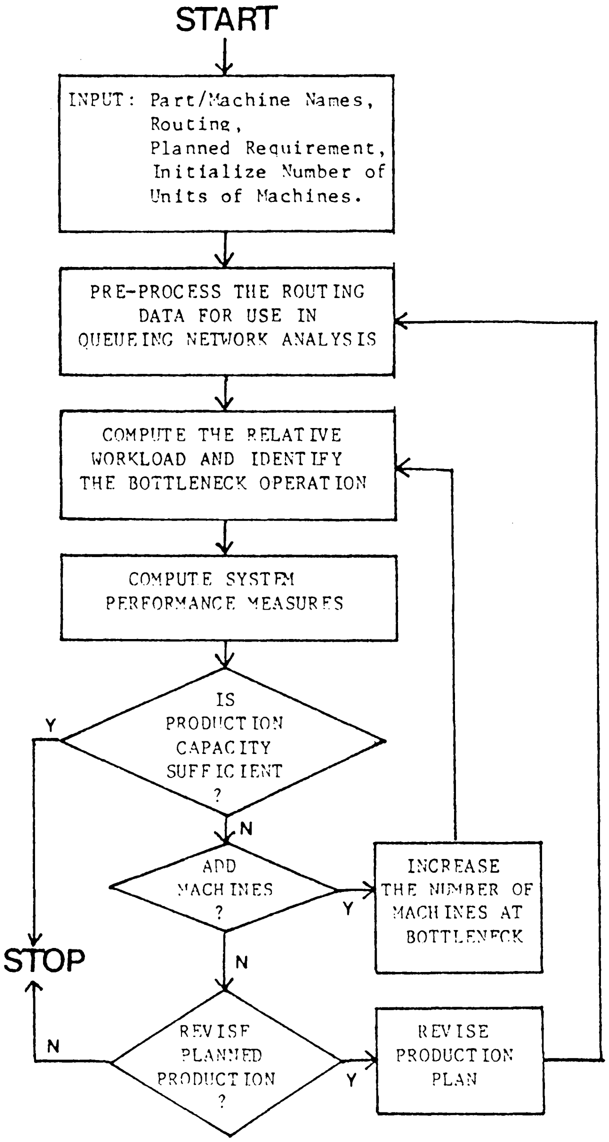


Figure 6: The Logic Flow Chart for the System Synthesis Model

4.1.2 System Configuration

The output of the system synthesis model is a set of system configurations specifying the minimum number of units for each machine type, the investment requirement, the capacity of the material handling system, the set of product families included in the production plan, and the output capacity of the system. If desired, the computed machine requirements may be modified to account for the shop/system efficiency, the limited tool storage capacities, and the desired machine redundancy.

The system synthesis model is used as the "determinstic phase" of the FMS justification model, to be described in Chapter six. An illustration of the model is provided in Section 6.5. A program listing of the system synthesis model can be found in Appendix B.

4.2 THE MACHINE MAPPING AND POOLING PROBLEM

The machine mapping and pooling problem is the problem of configuring tools and operations into a machine group(s). This problem has been addressed by Stecke and Solberg (STECK77, STECK81, and STECK83). Their objective functions were the following:

1. To maximize flexibility by maximizing the number of machines that can perform each operation.

2. To pre-balance the total processing times by minimizing the deviation of the total processing times assigned to each machine.

3. To minimize the distance travelled by maximizing the number of consecutive operations to be performed in the same machine.

The constraints include:

1. Each operation must be assigned to some machine.

2. Certain operations may require a specific machine.

3. Each tool must be allocated some slots in the tool magazine and the number of tool slots available for each machine is limited (this constraint may not hold in the near future, due to technological advancement).

4. All tools needed in a single operation must be assigned to the same machine.

5. It is desirable to assign operations which require common sets of tools to the same machine.

By defining a different set of variables, and using implied constraints, this section presents a zero-one linear programming model for solving the machine mapping and pooling problem. Large scale zero-one linear programming problems with similar structure have been solved efficiently using implicit enumeration and the cutting plane techniques. Specifically, Crowder, Johnson, and Padberg reported on the optimality of 10 large-scale zero-one linear programming problems of real-world situations. The problems are characterized by sparse constraint matrices with rational data, very much similar to the model presented in this Chapter. The authors designed an experimental computer system called PIPX which took a few minutes of CPU to solve Zero-One linear programming problems with up to 2,750 variables on the IBM MPSX/370 system (see reference PADBM83).

4.2.1 The Decision Variables

There are two sets of decision variables. The first set of decision variables, the $X_{ik's}$, maps the machines to the operations; the second set of decision variables, the $Y_{kj's}$, assigns tools to the machines.

Typically, the machining stations in a flexible manufacturing system are versatile NC machines. The machining stations are generally equipped to process several operations. A typical part may visit each station more than once, and may have numerous alternative routes. Each machining station has a tool magazine where tools are inserted. A machining station must be equipped with all the tools needed for an operation set.

Let n equal the total number of operations, l equal the total number of tool types, and m equal the total number of machines. The maximum number of decision variables is m x (n + l). Since many of the $X_{ik's}$ and $Y_{kj's}$ are known to be zero, the actual number of variables is much less.

By definition,

X_{ik} = 1 if machine k is equipped to process operation i.
= 0, otherwise.

Y_{kj} = 1 if machine k is equipped with tool type j.
= 0, otherwise.

where $i= 1,2,\ldots,n$
$j= 1,2,\ldots,l$
$k= 1,2,\ldots,m$

4.2.2 The Parameters

The parameters of the model are shown in Table 14. Note that machines of the same type may be "forced pooled" by treating them as one machine and defining the parameter S_i as the number of parallel servers(machines) in the machine group. P_{ik} then should be replaced by (P_{ik}/S_i).

4.2.3 The Constraints

Five linear constraints must be satisfied. These are:

1. Each operation must be assigned to some machine. Therefore:

$$1 \le \Sigma_k \; X_{ik} \le g_i, \text{ for every } i.$$

Note that if $\Sigma_k X_{ik} = 1$ for all i, a fixed route is imposed. Otherwise, some operations may have alternative routing.

TABLE 14

Parameters of the Machine Mapping and Pooling Model

T_{ij} = 1 if tool j is needed in operation i.

= 0, otherwise.

P_{ik} = the processing time for operation i on machine k.

= Pi if processing time is machine-independent.

a_k = maximum number of tool slots in the tool magazine of machine k.

b_i = number of tools needed for operation i.

c_j = number of slots occupied by tool j.

d_k = maximum workload for machine k.

e_k = minimum workload for machine k. (determined by the production ratio of each part type and the routing of the part types).

g_i = number of duplicate assignments allowed for operation i (minimum of 1).

2. Certain operation may require a specific machine. Thus: $X_{ik} = 0$ for every operation i that cannot be processed by machine k. (i.e., the variable can be deleted)
3. Each tool must be allocated some slots in the tool magazine and the number of tool slots available for each machine is limited.

$$\Sigma_j \ c_j Y_{kj} \leq a_k, \text{ for every machine } k.$$

4. All tools needed in a single operation must be assigned in the same machine. Specifically, machine k will not be able to process operation i if for some $T_{ij}=1$, $Y_{kj}=0$.

That is, $\Sigma_j \ T_{ij} Y_{kj} < b_i$ implies $X_{ik}=0$.

Therefore, the following constraint should hold:

$$X_{ik} \leq (1/b_i) \ (\Sigma_j \ T_{ij} Y_{kj})$$

Clearly, whenever

$$\Sigma_j \ T_{ij} Y_{kj} < b_i,$$

the right hand side of the constraint is less than 1, and therefore, $X_{ik}=0$.

5. One may implicitly solve the line balancing problem by imposing an upper limit and a lower limit to the workload for each machine. The following then should hold:

$$\Sigma_i \ P_{ik} X_{ik} \leq d_k, \text{ for all machine } k.$$

and

$$\Sigma_i P_{ik}X_{ik} \geq e_k, \text{ for all machine } k.$$

The constraints can be tightened to provide an almost equal production balance to the system.

4.2.4 The Objective Functions

The objective function may be one or more of the following:

1. To maximize flexibility by maximizing the number of machines that can perform each operation.

   $$\text{Maximize } z = \Sigma_k \Sigma_i X_{ik}.$$

2. Alternatively, one may wish to weight the operations by their processing times. Then

   $$\text{Maximize } z = \Sigma_k \Sigma_i P_{ik}X_{ik}.$$

   or

   $$\text{Maximize } z = \Sigma_i P_i \Sigma_k X_{ik}.$$

3. To balance the assigned workload by minimizing the maximum differences in machine workload. Notice that the expression

   $$\Sigma_i P_{ik}X_{ik}$$

   represents the maximum workload possible for machine k. The objective function then is:

   $$\text{Maximize } Z = z$$

   where the inequalities

   $$\Sigma_i P_{ik}X_{ik} - \Sigma_i P_{ik'}X_{ik'} - z \leq 0$$

   $$\Sigma_i P_{ik'}X_{ik'} - \Sigma_i P_{ik}X_{ik} - z \leq 0$$

for all $k \neq k'$

must be satisfied in addition to the constraints listed in the preceding section.

4. One may wish to minimize the sum of the absolute deviation in machine workload. i.e.,

$$\text{Min } Z = \Sigma_{k \neq k'} \left| \Sigma_i P_{ik'} X_{ik'} - \Sigma_i P_{ik} X_{ik} \right|$$

Notice that this objective function is non-linear. To transform the objective function to a linear function, let

$$r_{kk'}^{+} = \Sigma_i P_{ik} X_{ik} - \Sigma_i P_{ik'} X_{ik'},$$

$$\text{if } \Sigma_i P_{ik} X_{ik} > \Sigma_i P_{ik'} X_{ik'},$$

$$= 0, \text{ otherwise.}$$

and

$$r_{kk'}^{-} = \Sigma_i P_{ik'} X_{ik'} - \Sigma_i P_{ik} X_{ik},$$

$$\text{if } \Sigma_i P_{ik} X_{ik} < \Sigma_i P_{ik'} X_{ik'},$$

$$= 0, \text{ otherwise.}$$

Then $r_{kk'}^{+} - r_{kk'}^{-} =$

$$\Sigma_i P_{ik} X_{ik} - \Sigma_i P_{ik'} X_{ik'},$$

for all $k \neq k'$.

The new objective function is:

$$\text{Minimize } Z = \Sigma_k [r_{kk'}^{+} + r_{kk'}^{-}]$$

Subject to:

$$\Sigma_i P_{ik} X_{ik} - \Sigma_i P_{ik'} X_{ik'} - r_{kk'}^{+} - r_{kk'}^{-} = 0$$

for all $k \neq k'$. and all the constraints listed earlier.

Notice that m(m-1) auxiliary variables and m(m-1)/2 constraints have been added. The objective function may present some dimensional problems for a fairly large sized system. Fortunately, typical FMS's consists of fewer than a dozen workstations.

5. To minimize the distance travelled by maximizing the number of consecutive operations to be performed in the same machine.

   If operations i and i+1 are consecutive operations, then $X_{ik} - X_{i+1,k} = 0$ if the two operations are processed in the same machine k. Otherwise,

   $$X_{ik} - X_{i+1,k} = +1 \text{ or } -1.$$

   The objective is to :

   $$\text{Min } z = \Sigma_{\text{for } i=1 \text{ to } n-1} \Sigma_{k=1 \text{ to } m} |Xik - Xi+1,k|$$

   Again, the objective function can be transformed to a linear objective function as in the preceding section.

4.3 SUMMARY

This chapter examined three of the eight FMS design/implementation issues outlined in Figure 2. Two decision models were presented. The system synthesis model determines the initial FMS configuration to satisfy a given production requirement. It also provides valuable insights on the incremental productivity of additional investment by evaluating the sensitivity of deleting one or more product families from the production plan.

The second model presented is a zero-one linear programming model for solving the machine mapping and pooling problems outlined in (STECK80).

Chapter V

INPUT CONTROL IN FLEXIBLE MANUFACTURING SYSTEMS

Batching is the partitioning of parts selected for processing in a flexible manufacturing system, into batches to be launched to the system at a given time. The decision variables include the number of batches, the part mix, and the lot size of each part type. In this chapter, the methodology for evaluating the viability of operating the FMS as a single-class-job-shop or a programmable transfer line (SCJS/PTL) is presented. Insights to various performance dependencies are drawn from the case studies presented.

5.1 BALANCE DELAY AND THE VIABILITY OF THE SCJS/PTL

As a SCJS/PTL, the machining stations are more likely to be pooled without violating the technological limitations of the system (such as the number of tools a machining station can have). This is true because all facilities are dedicated to the processing of parts of a single type. Pooling the machines reduces the frequency of blocking and has been shown to increase the system performance. See for instance, references (STECK81) and (KLEIL76). However, to operate the FMS as a SCJS/PTL, a system configuration must be determined for each part type. A major problem is that of line balancing. For each line, the problem is to assign individual tasks to the workstations in such a manner that some predetermined measure of performance is optimized. If a line is balanced perfectly, all stations perform the same amount of work and smooth production with no forced delay is achieved.

To illustrate the importance of balance delay on the viability of the SCJS/PTL, consider a simple system of two stations and a robot transporter. Two parts of equal proportion are to be processed. The parts are launched from station M_1 for processing, then picked-up by the robot to station M_2 for further processing. It is assumed that a part is picked-up by the robot only when its destination is free. Otherwise, the part remains in its current location and blocking occurs. The processing times for the two part types are:

Part Type	Processing Time (minutes)
A	15.0 - (1.0) - 15.0
B	20.0 - (1.0) - 20.0

* transport time in parenthesis

If parts of one type are batched in lots and then reset for the other part type, the cycle time for the two part types is 16 minutes and 21 minutes, respectively. The output rate therefore is 3.75 parts per hour for A, and 2.857 parts per hour for B or an average of 3.3 parts per hour.

If equal proportions of each part are processed, the average system output rate is 3.243 parts per hour.

If both part types are processed simultaneously at equal proportion, a simple Gantt chart analysis (or simulation) would show the output rate to be about 2.9 parts per hour. Operating the FMS as a SCJS/PTL in this example resulted in 14 percent improvement in production capacity.

Now consider the same system with the following processing times:

Part Type	Processing Time (minutes)
A	10.0 - (1.0) - 20.0
B	30.0 - (1.0) - 10.0

* transport time in parenthesis

Following the analysis as above, the output for the SCJS/PTL alternative is 2.3 parts per hour; the output rate for the random FMS is approximately 2.9 parts per hour. Moreover, the output rate of the random FMS can be increased to 3.0 parts per hour by adding a temporary storage station, and loading the system with 3 parts at all times. This example illustrated the case where high balance delay in a SCJS/PTL resulted in lower output rate.

5.2 BLOCKING AND LOCKING

Let m equal the number of workstations, s equal the number of temporary storage stations, and n equal the number of parts loaded to the FMS. "n" is sometimes called the system loading $n \le m + s$. An FMS is locked when parts in the system are blocked indefinitely. For $s = 0$, locking may occur if the flow of the n parts in the system constitutes one or more closed path(s). This is illustrated in Figure 7. For $s > 0$, locking is possible if $n \ge m + s$. Figure 8 illustrates such a case. Clearly, locking could have been avoided if the workflows are non-backtracking.

The number of backtracking workflows generally increases with the variety of part types (hence a diversity of routing) in the system. By limiting the number of part types in the system, batching reduces the frequency of blocking and the probability of locking

Consider a system consisting of three machines and a robot transporter. Two part types of equal proportion are produced in this system as follows:

Part Type	Routing	Processing Times (minutes)
A	M_1-M_2-M_3	4.0 - (0.1) - 5.0 - (0.1) - 4.0
B	M_3-M_2-M_1	2.0 - (0.1) - 3.0 - (0.1) - 3.0

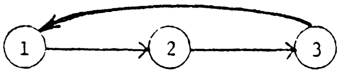
* transfer times in parenthesis

Part type A is launched through machine M_1 and exits the system from machine M_3; Part type B enters the system

(a)

A part completing operation in station 1 waits for station 2 to become available while the part in station 2 waits to be routed to station 1.

(b)



The part in station 2 blocks the part in station 1. The part in station 3 blocks the part in station 2 while being blocked by the part in station 1.

(c) Other possibilities

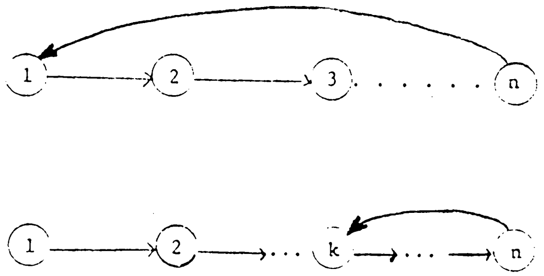
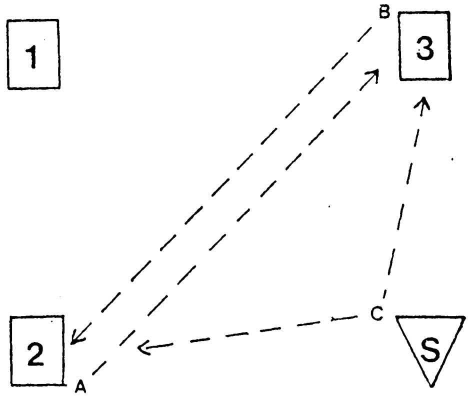


Figure 7: Locking Occurs When a Closed Path Is Formed



Part c just completed processing in station 1 and is to be routed to either station 3 or station 2. Since neither stations are available, the part is routed to the temporary storage station S. Meanwhile, part A is waiting in station 2 for station 1 to become available while part B is waiting in station 3 for station 2 to become available.

Figure 8: Locking Occurs Even With Temporary Storage Stations

through machine M_3 and departs from machine M_1. Thus the workflow of the two part types are directed opposite each other and the flow pattern of the two part types forms two closed paths. To avoid locking, two temporary storage stations are added to the system.

Assuming that parts of either type are launched alternately, the system was simulated for 2,400 minutes; for n=3. The simulated average output rate is about one part per 4.2 minutes. The results may be taken as deterministic since the simulation variance is very small.

Now, if parts of one part type are batched in lots and then reset for the other part type, the average output rate (if equal proportion of each part type are processed) is 14.63 parts per hour or one part every 4.1 minutes. Furthermore, there is no need for any temporary storage station and the material handling requirement is also reduced. Interesting enough, this difference in output rate represents the additional material handling times for transfering parts to and from the two temporary storage stations in the random flexible manufacturing system.

5.3 WORK-IN-PROCESS STORAGE

Consider the case where eight part types of equal proportion are processed in the same system as follows:

Part Type	Routing	Processing Times (minutes)
A	M_1-M_2-M_3	15.0 - (0.1) - 20.0 - (0.1) - 25.0
B	M_1	15.0
C	M_2	20.0
D	M_3	25.0
E	M_1-M_2	15.0 - (0.1) - 20.0
F	M_2-M_3	20.0 - (0.1) - 25.0
G	M_1 - M_3	15.0 - (0.1)- - 25.0
H	M_3-M_2-M_1	25.0 - (0.1) - 20.0 - (0.1) - 15.0

* transfer times in parenthesis

Suppose only one temporary storage station is available. A simulation of the system showed that when three parts are loaded simultaneously, locking occurred after completing only five parts in 137 minutes. Now another temporary storage is added so that $n \leq m + s - 1$. The system was again simulated for 7,500 minutes, and statistics were cleared before the 2,500th minute. The simulated output rate is approximately constant at 3.39 parts per hour.

Now consider batching. Suppose the parts are partitioned into [A], [H], [B,E], [C,F], and [D,G]. The same system was simulated. The output rate increased by about nine percent, thereby favoring the alternative of operating the FMS as a SCJS/PTL for [A] and [H], and two disjoint SCJS/PTL for [B,E], [C,F], and [D,G].

However, if the number of temporary storage stations were increased from two to five, and loading the system with eight parts at all times, the output of the random FMS can be increased to the same level as that obtained by batching.

5.4 THE CATERPILLAR SYSTEM AS A PROGRAMMABLE TRANSFER LINE

Thus far, simple systems consisting of two to three stations have been used in the illustrations. The material handling activities played a minor role in these illustrations and the impact of complex control in the random FMS has not been emphasized adequately. The following case studies illustrate other ramifications of batching.

Consider the Caterpillar system. A simulation model for the system (c.f. Section A.14 of Appendix A) was executed. The processing times were assumed to be constant and that equal proportion of the three part types were processed. The layout of the system was divided into 32 zones and the location of each resource/part was designated by a zone number. The carrier speed was assumed to be constant at 10 zones per minute. Traffic interference, however, was ignored (modelling traffic interference favors the SCJS/PTL alternative).

The number of parts circulating in the system was fixed at 16. To avoid the undesirable (and unrealistic) situations of the pallets, the manual load/unloaders, and/or the manual inspector becoming the bottleneck in a capital-intensive flexible manufacturing system, the pallets and manual load/unloader were not modelled. For Part type 3, the number of inspectors was increased to two. Pooling of machining stations for the SCJS/PTL alternative was assumed.

Table 15 summarizes the statistics collected from the simulation. As shown, the output capacity for the two alternatives are approximately equal. This is very encouraging. Each operation shown in the routing is actually a collection of consecutive operations. For instance, the routing of Part type 1 consists of 49 operations which Caterpillar had aggregated into 9 operation sets using a heuristic algorithm for machine mapping/pooling (see STECK77). The aggregation of operations to operation sets is intended for the simultaneous processing of the three part types. For the SCJS/PTL alternative, the operations may be aggregated to minimize the balance delay. A more optimal job routing for each part type will likely result. For illustration purposes, all subsequent analyses were based on the original operation sets.

TABLE 15

SCJS/PTL Versus Random FMS (Caterpillar System)

I. Random FMS

Estimated Mean Flow Times:

Part	Mean	Variance of the Mean
1	188.94	11.57
2	295.58	9.62
3	410.85	7.19

Output Rate = 3.22 parts/hour.

II. SCJS/PTL

Estimated Mean Flow Times:

Line	Mean	Variance of the Mean
I	169.57	0.07
II	426.58	4.17
III	332.27	1.33

Output Rate = 3.10 parts/hour.

Difference: 0.59 parts/shift(2.28 percent).

5.5 NON-DETERMINISTIC PROCESSING TIMES

CAN-Q is perhaps the most widely known analytical model for the FMS. One assumption of CAN-Q which may limit its application is the assumption of exponentially distributed processing times when in fact, they are more likely to be deterministic. To examine the effect of the probability density function of processing times on the viability of the SCJS/PTL alternative, a simulation study on the Caterpillar system was conducted. Table 16 compares the average output capacity of the two alternatives (SCJS/PTL versus random FMS) for normally distributed, uniformly distributed, and exponentially distributed processing times; Table 17 compares the two alternatives for varying levels of variances, assuming normally distributed processing times. As shown, the SCJS/PTL alternative performed as well as the random FMS alternative, <u>except</u> for the case of exponentially distributed processing times where the output capacity for the SCJS/PTL alternative is more than 16 percent higher.

5.6 VARIETY OF PARTS

The diversity of routing generally increases as the variety of parts in the system increases. Batching alters the pattern of workflows in the system. The flow patterns of random flexible manufacturing systems exhibit characteris-

TABLE 16

SCJS/PTL Versus Random FMS (Non-deterministic Processing times)

UNIFORM DISTRIBUTION

Random FMS:	Part	Mean Flow Time	Variance	Parts/ Shift
	1	186.36	20.96	
	2	311.50	7.89	
	3	412.67	9.47	25.30
SCJS/PTL :				
	I	170.52	0.81	
	II	424.59	5.33	
	III	320.47	4.53	25.16

NORMAL DISTRIBUTION

Random FMS:	Part	Mean Flow Time	Variance	Parts/ Shift
	1	176.08	19.43	
	2	320.09	12.93	
	3	408.66	3.46	25.46
SCJS/PTL :				
	I	172.35	0.73	
	II	429.32	9.22	
	III	320.57	3.49	24.98

EXPONENTIAL DISTRIBUTION

Random FMS:	Part	Mean Flow Time	Variance	Parts/ Shift
	1	211.58	119.60	
	2	446.41	122.73	
	3	496.43	184.77	19.96
SCJS/PTL				
	I	175.78	8.61	
	II	448.65	55.79	
	III	367.73	66.40	23.22

TABLE 17

SCJS/PTL Versus Random FMS (Various Levels of Variances)

Variance to Mean Ratio	0.0	1.0	2.0	3.0	4.0
Random FMS:					
Part-1					
Mean Flow Time	188.94	175.95	180.44	180.11	183.23
Variance of Mean	11.57	14.09	9.95	6.89	8.45
Part-2					
Mean Flow Time	295.58	314.02	321.78	332.85	340.66
Variance of Mean	9.62	10.00	10.42	10.02	9.11
Part-3					
Mean Flow Time	410.85	421.85	429.12	443.12	447.10
Variance of Mean	7.19	7.26	6.15	6.02	8.22
SCJS/PTL:					
Line-I					
Mean Flow Time	169.57	172.17	172.53	170.97	173.46
Variance of Mean	0.07	0.79	1.13	1.47	2.07
Line-II					
Mean Flow Time	426.58	428.63	432.85	433.80	439.78
Variance of Mean	4.17	6.77	8.33	9.00	12.69
Line-III					
Mean Flow Time	332.27	335.79	344.93	348.03	354.95
Variance of Mean	1.33	2.00	3.18	8.34	6.23
Output Capacity (Parts per 8-hour Shift):					
Random FMS	25.73	25.27	24.74	24.10	23.73
SCJS/PTL	24.82	24.60	24.24	24.18	23.80
% Difference	3.54	2.65	2.02	-0.33	-0.29

tics of random job shops, while programmable transfer lines have flow patterns resembling those of sequential flow shops.

In a system where the work-areas are arranged such that subsequent operations are located immediately adjacent to each other, what is commonly called a line production results. In the most refined state the parts moved continuously and at a uniform rate through a series of balanced operation.

Let the matrix A = (a_{ij}, i = 1, 2, ..., m; j = 1, 2, ..., m) where m is the number of machining stations in the system.

a_{ij} = 1, if parts are routed from i to j,
= 0, otherwise.

Then :

1. The system is a sequential flow shop if

   a_{ij} = 1, for j = i + 1
   = 0, otherwise.

   and $\Sigma a_{ij} = m - 1$.

2. The workflow is not necessarily sequential but the work flow is strictly uni-directional (non-backtracking) if

   a_{ij} = 1, for j > i
   = 0, otherwise.

   and $\Sigma a_{ij} \leq m(m-1)/2$.

A system with no backtracking workflow will not lock.

3. A system is a random job shop if

   a_{ij} = 0, for j = i
   = 1, otherwise.

   and $\Sigma\, a_{ij} \leq m(m-1)$.

4. As a measure of the relative diversity of routing, define the d-ratio as: $d = (\Sigma a_{ij}) / m(m-1)$, where m = the number of workstations. (Note: Pooling of machine tools also increases the diversity of routing. However, in the context of this study alternative routes are not considered in the definition of Σa_{ij}.)

The Caterpillar system consists of 10 workstations (not counting the 16 loading/unloading stations). For the routing given in Tables 3, 4, and 5, the d-ratio is 0.20. Suppose the nine machining stations were replaced by:

1. a milling station four times the efficiency of an OM3-Omnimill,
2. a drilling station three times the efficiency of an OD3-Omnidrill, and
3. a turning station twice as productive as a G & L vertical-turret lathe.

Consider the routing in Tables 3 to 5 with the corresponding adjustments in processing times. The new d-ratio is 0.75, indicating extensive backtracking.

Table 18 compares the output capacity estimated in a simulation study. The SCJS/PTL alternative yields a significantly higher output (about 15 percent improvement).

Consider next the original Caterpillar system. Suppose a new part type (routing shown in Table 19) is processed in the same system with the original three part types. For the random FMS, the d-ratio is 0.255, a modest increase from the original d-ratio of 0.20. However, the addition of the new part type significantly improved the balance of the workload for the FMS. The system was simulated to estimate the output capacity of the two alternatives. Table 20 shows that inspite of a perfect workload balance, the output capacity of the random FMS is only slightly higher than the output capacity of the SCJS/PTL alternative.

Consider now an 8 part type system. In addition to the four part part types above, four additional part types with the same processing times but in reverse sequence are processed in the same system. Notice that for the random FMS, the workload is still perfectly balanced. A simulation study shows a decline in output capacity from 27.11 parts per shift in the preceding example, to only 25.49 parts per shift. Incidentally, the d-ratio for this 8-part type system is 0.444.

TABLE 18

SCJS/PTL Versus Random FMS (Embellished Caterpillar System)

I. The Random FMS

Estimated Mean Flow Times:

Part	Mean	Variance of the Mean
1	137.39	0.94
2	266.83	1.23
3	388.23	2.22

II. SCJS/PTL

Estimated Mean Flow Times:

Line	Mean	Variance of the Mean
I	135.92	0.13
II	316.89	0.54
III	237.87	19.30

III. Output Capacity (Equal Proportion of Parts)

Random FMS : 21.81 parts/shift.
SCJS/PTL : 25.02 parts/shift.

Difference : 3.21 parts/shift(-14.73 percent).

TABLE 19

Job Routing for the Fourth Part Type

Operation	Station	Time	Frequency
1	19	10.00	1.00
2	3	11.70	1.00
3	1	14.80	1.00
4	10	11.30	0.50
5	8	10.50	1.00
6	5	11.20	1.00
7	19	10.00	1.00
8	13	10.00	1.00
9	2	11.20	1.00
10	6	10.10	1.00
11	7	10.30	1.00
12	10	11.30	0.50
13	14	10.00	1.00

TABLE 20

SCJS/PTL Versus Random FMS (Four Part Type System)

I. Random FMS (Perfectly Balanced Workload)

Estimated Mean Flow Times:

Part	Mean	Variance of the Mean
1	157.05	0.90
2	300.45	0.71
3	382.95	2.28
4	292.76	1.47

II. SCJS/PTL

Estimated Mean Flow Times:

Line	Mean	Variance of the Mean
I	169.57	*
II	426.58	4.17
III	332.27	1.33
IV	252.75	*

* variance is very small.

III. Output Capacity (Equal Proportion of Parts)

Random FMS : 27.11 parts/shift.
SCJS/PTL : 26.01 parts/shift.

Difference : 1.10 parts/shift(4.06 percent).

5.7 NUMBER OF MACHINE TOOLS

The variety and number of work stations in a flexible manufacturing system are determinants of the balance delay attainable in the programmable transfer line. At the extreme, if there is only one type of station, pooling the machines maximizes flexibility of routing and minimizes the balance delay (the balance delay is zero if all machines are pooled). If there are two types of stations, say, drilling and milling, the balance delay can be minimized if the drilling and milling requirement of each part type are matched by the capacity of the workstations. As the variety of work stations increases, the likelihood of a good match decreases. In general, one or more station types may be under-utilized for some part types. This opens the avenue for:

1. operating the system as two or more disjoint production lines, each line processing one single part type at a time, or
2. partitioning the part types such that each batch consists of one or more part types.

Consider again the original Caterpillar system with two additional part types shown in Tables 21 and 22. A simulation study was conducted to compare the output capacity of the following alternatives:

1. Random FMS,
2. SCJS/PTL, and
3. Launching Part-1, Part-2 and Part-X as one batch, and Part-3 and Part-Y as another batch.

The results of the simulation are summarized in Table 23. As shown, the two-batch alternative has the highest output rate, while the SCJS/PTL alternative is shown to be the least productive. In this example, batching the five part-types into two groups resulted in a perfect workload balance for the system. On the contrary, the two additional part types, when launched individually, resulted in high balance delays.

As a further illustration, consider Part-1, Part-2, and Part-Y only. Instead of processing the three part types simultaneously, what if the system is first operated as a SCJS/PTL to process Part-2, then partitioned into two disjoint flow lines to process Part-1 and Part-Y? The result of the simulation is summarized in Table 24. As shown, operating the system as two disjoint flow lines resulted in higher output than the SCJS/PTL alternative, but the output is slightly lower than processing the three part types simultaneously. The SCJS/PTL alternative in this example yields the lowest output capacity because of high balance delay, while the random FMS alternative yields the highest output because of a perfectly balanced workload.

TABLE 21

Job Routing for Part Type- X

Operation	Station	Time	Frequency
1	18	1.00	1.00
2	2	34.80	1.00
3	3	34.80	1.00
4	5	34.80	1.00
5	6	21.90	1.00
6	16	10.00	1.00
7	17	10.00	1.00
8	8	14.20	1.00
9	7	16.30	1.00
10	15	10.00	1.00
11	16	10.00	1.00
12	9	22.60	1.00
13	4	8.50	1.00
14	10	4.80	0.20
15	19	10.00	1.00

TABLE 22

Job Routing for Part Type- Y

Operation	Station	Time	Frequency
1	13	10.00	1.00
2	1	38.40	1.00
3	4	15.10	1.00
4	10	32.00	0.33
5	6	11.80	1.00
6	10	32.00	0.33
7	14	10.00	1.00
8	15	10.00	1.00
9	8	19.90	1.00
10	7	17.60	1.00
11	10	32.00	0.33
12	18	10.00	1.00

TABLE 23

The FMS, the SCJS/PTL and the Two-Batch Alternatives

Part-Type	1	2	3	4	5
Random FMS					
Mean Flow Time	155.22	340.47	286.34	401.71	414.20
Variance of Mean	1.14	2.56	1.41	2.31	5.17
SCJS/PTL					
Mean Flow Time	169.57	426.58	282.07	332.27	437.77
Variance of Mean	0.07	4.17	8.36	1.33	2.44
Two Batches					
Mean Flow Time	163.87	315.38	280.44	383.07	356.57
Variance of Mean	0.70	1.64	1.81	0.89	0.75

Output Capacity (Equal Proportion of Parts)

Random FMS : 24.03 parts/shift.
SCJS/PTL : 23.30 parts/shift.
Two Batches : 25.61 parts/shift.

TABLE 24

The Disjoint Two Flow Lines Within a System Alternative

Part-Type	1	2	5
SCJS/PTL			
Mean Flow Time	169.57	426.58	437.77
Variance of Mean	0.07	4.17	2.44
Two Batches as Random FMS			
Mean Flow Time	163.87	315.38	356.57
Variance of Mean	0.70	1.64	0.75
Two Batches as Disjoint Flow Lines			
Mean Flow Time	196.57	426.58	233.32
Variance of Mean	3.69	4.17	4.63

Output Capacity (Equal Proportion of Parts)

SCJS/PTL : 22.28 parts/shift.
Two Batches (Random) : 27.57 parts/shift.
Two Batches(Disjoint): 26.90 parts/shift.

5.8 A METHODOLOGY FOR OPTIMAL BATCHING IN FLEXIBLE MANUFACTURING SYSTEMS

Having demonstrated the potential advantages of batching, this section presents a methodology for solving the FMS batching problem. The procedure combines mathematical programming with simulation.

5.8.1 Zero-One Linear Programming Problem

Let I = the number of part types,

J = the number of machines,

B = the number of batches, and

a_{ij} = the processing times of part type i on machine j; sum over all visits and weighted by the product mix.

Let X_{ib} = zero-one variable, for $i = 1, 2, \ldots, I$; and $b = 1, 2, \ldots, B$.

$X_{ib} = 1$, if part type i is in batch b
$= 0$, otherwise.

The workload assigned to machine j for batch b is

$$\Sigma_i \, a_{ij} \, X_{ib}$$

For machine j and j', the difference in assigned workload is

$$\Sigma_i \, a_{ij} \, X_{ib} - \Sigma_i \, a_{ij'} \, X_{ib}$$
$$= \Sigma_i (a_{ij} - a_{ij'}) \, X_{ib}, \text{ for } b = 1, 2, \ldots, B.$$

To balance the workload in each batch, a zero-one integer programming model is shown in Table 25 which minimizes the maximum differences of workload. The model has (I x B + 1) variables and [I + B (J^2- J+ 1)] constraints. The first two sets of constraints bound the differences in workload to be less than or equal to z (which is to be minimized). The third set of constraints make sure that each part type is assigned to one and only one batch, while the fourth set of constraints specifies that at least one part type is assigned to each batch. Notice that if $\Sigma_i X_{ib} = 1$, for all b, the flexible manufacturing system is a SCJS/PTL and B = I.

5.8.2 Batching Strategy

The mathematical model determines the value of the decision variables ($X_{ib's}$) for a given value of B. The optimal value of B is dependent on the diversity of workflow, the type of material handling equipment, the work-in-process storages, etc. and is subject to the tool capacity constraints. The following procedure combines the methodology of mathematical programming with simulation and heuristics for determining the optimal (or near optimal) batching strategy.

The procedure consists of two phases:

1. Arbitrarily choose a value of B ($1 \leq B \leq I$), and solve the zero-one integer program in Table 25. A

TABLE 25

Mathematical Model for Balancing Workload

Minimize $Z = z$

Subject to:

1. $\Sigma_i (a_{ij} - a_{ij'}) X_{ib} - z \leq 0$,
   for $j = j'$ and
   $b = 1, 2, \ldots, B$.
2. $\Sigma_i (a_{ij'} - a_{ij}) X_{ib} - z \leq 0$,
   for $j = j'$ and
   $b = 1, 2, \ldots, B$.
3. $\Sigma_b X_{ib} = 1$,
   for $i = 1, 2, \ldots, I$.
4. $\Sigma_i X_{ib} \geq 1$,
   for $b = 1, 2, \ldots, B$.
5. $z \geq 0$, $X_{ib} = (0,1)$, for all i,b.

small value of B is suggested since the size of the mathematical program increases geometrically with increasing value of B. Check if the optimal solution is acceptable. If the resulting workload difference is acceptable, go to phase 2; otherwise, decrease the value of B and repeat.

2. Examine each of the B batches. If there are homogeneous (single-part type) batches, schedule each of these batches immediately (SCJS/PTL). If there are batches consisting of a large variety of parts (10 or more), or if the tool capacity constraint is violated, repeat Phase 1 for each of these batches. Lastly, for the remaining batches, evaluate the viability of the SCJS, or partitioning the batches into batches of lesser part type by following the methodology illustrated in the preceding section.

5.9 SUMMARY

The advantage of limiting the number of classes of parts processed in the FMS is illustrated in this Chapter. Operating the FMS as a single-class-job-shop reduces the complexity of scheduling and sequencing. The SCJS/PTL alternative also permits the pooling of machine tools which otherwise is not possible because of the tool capacity constraint.

The viability of the SCJS/PTL has been shown to depend, to a large extent, on the balance delay of the SCJS/PTL, the balance of the workload of the random FMS, and the flow pattern. These variables in turn are determined by the diversity of the job routing (variety of parts) and the number and variety of machine tools.

Finally, Section 5.8 presented a two-phase methodology for optimal batching. One clear result of the modelling herein is that each system and product load must be analyzed carefully. Although some general guidelines were mentioned, the effect that configuration has on system performance merits careful examination.

Chapter VI

ECONOMIC DECISION ANALYSIS FOR FMS JUSTIFICATION

It is a concensus among practitioners that the problem of justifying a flexible manufacturing system is complex, difficult, and often paradoxical. In many instances, the FMS is designed for uncertain objectives. Some parts that the FMS is supposed to make may not be conceived/designed yet at the time the system is being evaluated. This chapter presents a methodology which applies queueing theory, decision theory, and simulation to analyze the economic merit of FMS application. At each decision point the decision-maker interacts with the simulation model by specifying alternative courses of actions to be evaluated. The capacity of the flexible manufacturing system may be augmented if projected demand exceeds current capacity. An after-tax analysis provides the decision-maker the performance measure for selecting the best alternative.

6.1 THE ECONOMIC VALUE OF FLEXIBLE AUTOMATION

The economic justification approach developed in this chapter highlights the economic value of the robustness of the FMS.

6.1.1 FMS Applications

The flexible manufacturing system is intended for mid-volume (200 to 30,000 parts per year), mid-variety (5 to 155 part types) production, thus filling the gap between stand-alone NC machines and the transfer line. The system is designed for the production of parts measuring up to 4 cubic feet and not weighing more than 2,000 pounds (HAYSP82). The system is most effective for processing parts with complex machined features, critical relationships between the surfaces, and that is able to be mounted on a fixture. Machining operations in an FMS may include milling, boring, grooving, facing, drilling, and tapping. The workpieces are in general fairly easily machinable material such as steel, iron or aluminum to minimize the need for frequent tool changes. Table 26 shows a list of some FMS systems installed in recent years and the unmanned material handling system used in each system.

6.1.2 Economic Advantages and Difficulties

Various authors in a variety of literature have dramatized the advantages of the flexible manufacturing system. Typical FMS goals include:

1. Reduction in manufacturing cost by lowering direct labor cost and minimizing scrappage, rework, and material wastage.

TABLE 26

Examples of FMS Installed in Recent Years

SOURCE : (MODMH82)

Material Handling System	Model FMS System
Powered Shuttle Car.	Herbert Warnke Kombinat East Germany
Powered Roller Conveyor Systems.	Sundstrand Corp.'s Aviation Division.
Computer-Controlled Towline Conveyor.	John Deere Component Works, Waterloo,Iowa.
Car-on-Track System.	Harris Corporation's Forth Worth, Texas.
Wire-Guided Vehicles.	Murata Machinery Ltd., Inuyama, Japan.
AS/RS Machines.	VUOSO (SKODA) Research Institute, Czechoslovakia.
	Fritz Heckert Prisma II, East Germany.
Industrial Robots.	Sperry Vickers, North American Group, Omaha, Nebraska.

2. Less skilled labor required.
3. Reduction in work-in-process inventory by eliminating the need for batch processing.
4. Reduction in production lead time permitting manufacturers to respond more quickly to the variability of market demand.
5. Better process control resulting in consistent quality.
6. Flexibility to produce spare parts or change product mix as market requirements demand.
7. Flexibility to augment the facility as required by increases in projected product demand.
8. Reduction in floor space requirement.
9. Reduction in production planning/control, design and manufacturing engineering personnel.
10. Staying at the forefront of technology.

The two major features of flexible manufacturing systems are increased product flexibility and production control. These features make FMS best suited for batch production of mixed parts on the same system. The FMS is capable of processing a wide range of part families through its versatile machine tools. Changes in specifications can be accomodated by simply changing the system software. This flexibility is not available in fixed automation. Furthermore, because of low set-up/changeover costs, the FMS is adaptable to a wider range of volume requirement and product variety than either the transfer line or the conventional job shop.

The flexibility of the FMS also provides the decision-maker the options in dealing with the uncertainties of the market. The decision-maker may incrementally upgrade the production capacity of the FMS to simultaneously process many types of parts. Expandability is one major feature of the FMS often neglected in traditional approaches to FMS justification. This is an advantage over a transfer line which is dedicated to a specific product and requires full capital expenditure before production can begin.

The flexible manufacturing system nonetheless has its shortcomings. They include: high initial capital investment, long system gestation period, more intense use of facilities, training and organizational changes, and the need for more complex planning and control procedures. A typical FMS costs about S12 million and takes about two years from initial conception to actual part production. The long system gestation, high investment requirement, compounded by the difficulty in quantifying the many intangible advantages of FMS make economic justification very difficult.

6.2 THE DECISION ANALYSIS PROCESS

Decision analysis provides a rational methodology in decision making under risk and uncertainty. The FMS justification approach incorporates two Operations Research methodologies -- Queueing Theory and Simulation -- within the framework of Decision Analysis.

Typically, a decision analysis cycle consists of three phases. The first phase is the deterministic phase where the variables affecting the decision process are identified, and the relationships among the variables defined. The second phase is the probabilistic phase where probability measures are assigned to the critical variables. This phase also introduces the concept of "risk preferences" of the decision maker in selecting among mutually exclusive alternative courses of action. The third phase is the information phase which provides the decision-maker the required information for interacting with the decision analysis model.

6.2.1 The Deterministic Phase

In the deterministic phase, the efforts devoted to modelling must be distinguished from the efforts devoted to analysis. In modelling, the system variables are defined and selected. In analysis, the extent changes in the variables affect the worth (value) of each alternative is examined. This sensitivity analysis is highly effective in refining the formulation of the problem.

One unique feature of the approach presented here is the capability of the simulator to model interactively the time-phased system synthesis process. The decision to augment production capacity in anticipation of higher future demand is a dynamic process. The decision-maker may decide to add a machining center or an automatic guided vehicle system for the FMS alternative, or add another transfer line to complement the current one. On the other hand, one may defer the investment decision to some future date by dropping or subcontracting one or more product lines from the production plan.

The deterministic phase determines the initial conditions for the analysis as well as for generating alternative FMS system configurations in the succeeding stages of the analysis. The FMS synthesis model described earlier in Section 4.1 constitutes this phase.

6.2.2 The Probabilistic Phase and the Information Phase

The probabilistic phase is also differentiated in terms of modelling and analysis. In the probabilistic modelling phase, probability distributions are assigned to the stochastic variables. The distribution of worth for each alternative course of action is calculated during the analysis phase. If stochastic dominance exists between a pair of alternatives, the stochastically dominated alternative is discarded. If stochastic dominance can not be established, comparison of alternatives may be resolved based on the decision-maker's relative aversion to risk. The decision-maker's risk preference may be represented by a utility curve-like "risk profile" (c.f. HOWAR66). The decision-maker's ranking of alternatives can be based on the "expected utilities" of the alternatives.

The probabilistic phase and the information phase are interweaved to form a dynamic decision process. The entire decision process is dynamically partitioned into stages, each stage corresponding to a year in the planning horizon.

6.3 THE FMS JUSTIFICATION MODEL

The model allows the decision-maker to dynamically interact with a computer program for evaluating the cost effectiveness of the flexible manufacturing system, as well as deciding on a suitable configuration for the system. The following are assumed to be random variables.

1. Product Demand and Part Mix
2. Product Routing/Engineering Design

The decision-maker is expected to have some prior knowledge of the probability density functions of these random variables. The method of subjective probability (see for instance, reference RAIFH57) is recommended in estimating the parameters of the random variables. Also, the decision-maker is expected to classify the different cost items listed in Section 6.4 in terms of recurrent and non-recurrent costs, as well as fixed costs versus variable costs. An excellent reference for the computation of these cost items can be found in reference (HAYSP82).

For cash flow analysis purposes, the expenses are assumed to be "end-of-the-year" cash outflows, while all investments are assumed to occur at the beginning of the year. For after-tax computations, the depreciation allowances are computed using the straight-line method (although it is very easy to accomodate other methods of depreciation). Manufacturing costs and depreciation allowances are tax-deductable and therefore represent reduction in cash outflows. For simplicity, the actual economic value of fixed assets are assumed to be equal to the book value.

A flow chart of the decision analysis model is shown in Figure 9. The computer program prompts the decision-maker for the input data which include: the planning horizon, the number of working hours per year, the corporate tax rate, the production plan and the job routing, the frequency and cost of engineering changes, and the initial system configuration (from Phase I).

The routing (which reflects product design/engineering) and actual production are simulated and the system performance for each configuration is evaluated using CAN-Q. Statistics for each year are summarized. At this point, the decision-maker is asked if there are system configurations not worthy of further evaluation (fathoming). For the following year, the machine requirements are computed based on projected production requirements and the decision-maker is asked to specify a new system configuration to be evaluated. The simulation procedure is repeated until the end of the planning horizon is reached.

The stochastic processes are modelled in Function RANDEM and Subroutine ROUTING. Function RANDEM(I) generates a sample value for the demand of product family I. Subroutine ROUTING models the uncertainty in product design/engineering by generating the routing and processing times for each product family. The routing process is modelled in the example in Section 6.5 by a Markov transition matrix, while the processing times and the actual production quantity are assumed to be uniformly distributed. A program listing of the decision analysis program can be found in Appendix C.

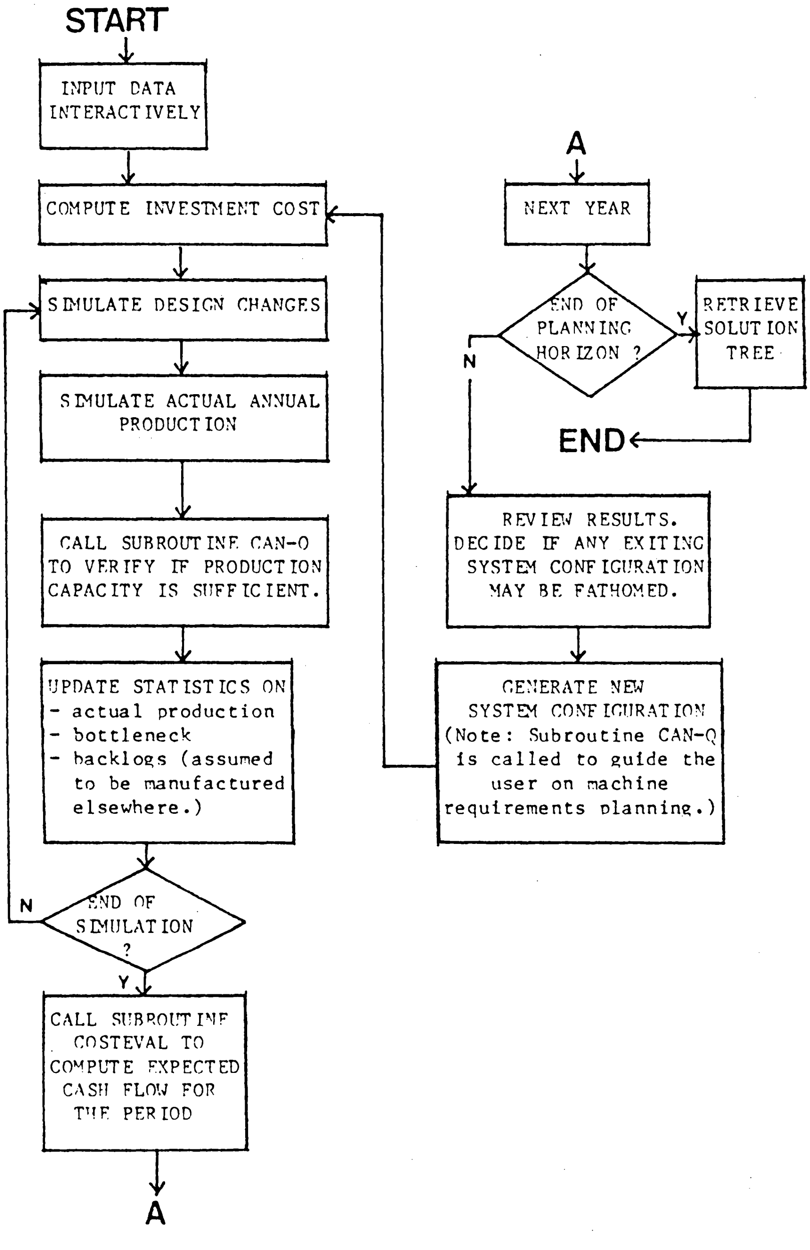


Figure 9: The Logic Flow Chart for the Decision Analysis Model

6.4 COST FACTORS FOR JUSTIFYING THE FMS

A basic part of any project evaluation is identifying and classifying the various cost/benefit elements. Classification of the cost/benefit elements simplifies the process of translating project activities into cash flows, which can be reduced to a single-valued economic measure of project worth.

6.4.1 Direct Versus Indirect Costs/Benefits

Traditionally, costs/benefits are classified as direct or indirect. Direct costs/benefits are readily measured and assignable to specific operations or products while the allocation of indirect costs are obscured. Examples of direct costs are raw material costs and direct labor costs, etc. Examples of indirect costs may include salary and wages of supervisory personnels, lighting, fuel, space, inventory, etc. Some costs/benefits are not readily quantifiable in dollars. Examples may include prestige, flexibility, reliability, etc.

6.4.2 Typical Costs Encountered in Manufacturing Systems

The following is a list of cost items typically encountered in manufacturing systems. Whenever applicable, these costs should be considered in evaluating investment alternatives. It is assumed that these costs can be classified as non-recurrent (one-time cost such as equipment investment or disposal), or recurrent costs. Also, to simplify the analysis, it is assumed that it is possible to express the recurrent costs as either independent of annual production (fixed costs), or proportionate to the quantity of production (variable costs).

1. Raw Material Costs -- FMS normally requires less direct material than a conventional system due to increased process control and repeatability. Raw materials costs are variable costs.
2. Equipment Investment and Disposal Costs -- These costs are incurred at the time of acquisition/disposal and are non-recurrent.
3. Equipment Operating Costs -- a) fuel, replacement parts, etc. are assumed to be variable costs. b) demand/preventive maintenance are assumed to be fixed costs. Although the maintenance cost per machine is higher for FMS, savings are expected because of fewer units of machines and because of the diagnostic sys-

tem built-in. Maintenance costs are considered as fixed costs, although the frequency of demand maintenance is affected by the run time. c) labor costs (direct labor and inspection) are generally lower in FMS. d) tooling costs are lower for FMS due to off-line pre-setting, tool storage and maintenance, and standardization. The tooling cost is a variable cost.

4. Material Handling Costs -- The material handling cost is lower in FMS. The material handling cost is assumed to be a fixed cost.
5. Engineering Design Change Costs -- This cost is determined by the frequency of changes, the extent of changes and the pay rate of the personnels. In this study, the engineering design change cost is assumed to be proportional to the number of changes. Engineering costs may be different for different system configurations.
6. Floor Space, Supervisory Personnel, Production Planning and Control Personnel Cost, etc. are assumed to be fixed costs.
7. Work-in-process Inventory Costs -- The savings for FMS in reducing the work-in-process inventory is reflected in terms of storage costs and the money-tied-

to-inventory costs. This cost item is assumed to be a fixed cost which differs from year to year, depending on the level of the projected production requirement.

8. Intangible Costs such as goodwill, pride in facility, operator morale, and compatibility with existing facilities are not reflected in the model.

6.5 A HYPOTHETICAL EXAMPLE BASED ON THE CATERPILLAR SYSTEM

The viability of the flexible manufacturing system must be evaluated viz a viz the effectiveness of the conventional job shop as well as the transfer line. Because of the "dynamic programming" nature of the model, it is recommendable to evaluate the three alternatives separately to avoid the pitfall of the "curse of dimensionality". For illustration purposes, this section illustrates the decision analysis of the FMS alternative only. The same procedure applies for evaluating the conventional job shop and other alternatives.

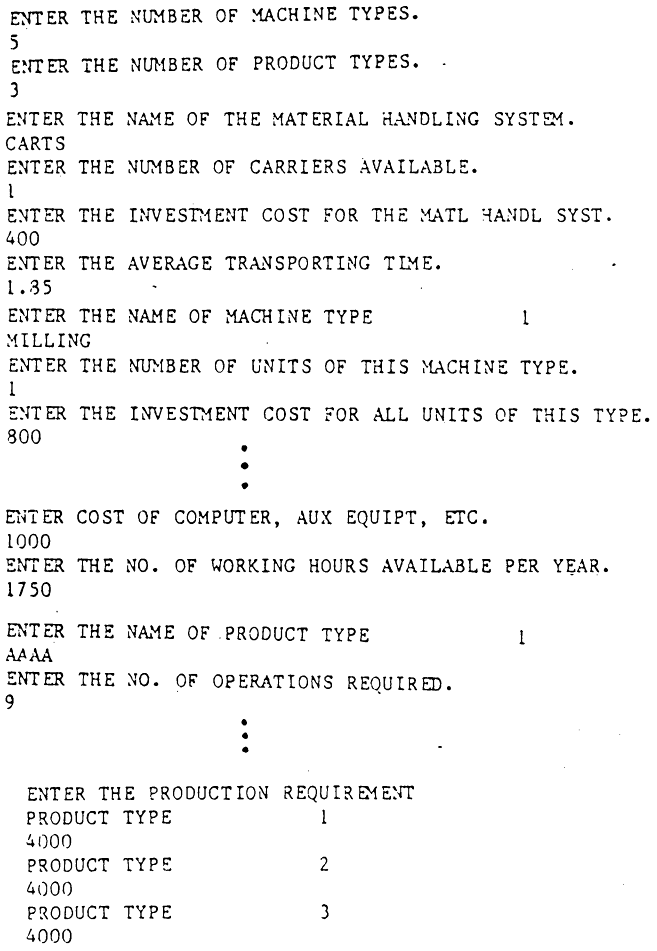
Consider the Caterpillar system with three machining operations -- milling, drilling, and turning. Together with inspection and loading/unloading, the machining stations are integrated by the material handling system under computer control. Three part families are processed in the system. The planned requirement and processing times are assumed to be uniformly distributed random variables, while the process routing is modelled as a random walk described by a Markov transition matrix. In practice, the probability density functions must be estimated either through historical data or the method of "subjective probability".

6.5.1 Phase I : System Configuration Analysis

Phase I is the deterministic phase. The routing and the planned requirements are assumed to be known and the computer program is executed to determine a set of initial system configurations for a given production requirement. First, the computer program prompts the decision-maker for the input data as shown in Figure 10.

Note that the decision-maker is asked to provide a "first guess" for the machine requirement. In the absence of any a-prioi information, the decision-maker may specify the minimum of one machine per machine type. The computer program then proceeds to compute the system performance measures using CAN-Q. The production capacity is compared with the planned requirement and the bottleneck operation is identified. Also, the investment requirement is updated. A typical response is shown in Figure 11.

At this time, the decision-maker interacts with the computer by indicating whether he/she wishes to add one or more



```
ENTER THE NUMBER OF MACHINE TYPES.
5
ENTER THE NUMBER OF PRODUCT TYPES.
3
ENTER THE NAME OF THE MATERIAL HANDLING SYSTEM.
CARTS
ENTER THE NUMBER OF CARRIERS AVAILABLE.
1
ENTER THE INVESTMENT COST FOR THE MATL HANDL SYST.
400
ENTER THE AVERAGE TRANSPORTING TIME.
1.35
ENTER THE NAME OF MACHINE TYPE             1
MILLING
ENTER THE NUMBER OF UNITS OF THIS MACHINE TYPE.
1
ENTER THE INVESTMENT COST FOR ALL UNITS OF THIS TYPE.
800
                      ⋮

ENTER COST OF COMPUTER, AUX EQUIPT, ETC.
1000
ENTER THE NO. OF WORKING HOURS AVAILABLE PER YEAR.
1750

ENTER THE NAME OF PRODUCT TYPE             1
AAAA
ENTER THE NO. OF OPERATIONS REQUIRED.
9
                      ⋮

 ENTER THE PRODUCTION REQUIREMENT
 PRODUCT TYPE            1
 4000
 PRODUCT TYPE            2
 4000
 PRODUCT TYPE            3
 4000
```

Figure 10: Example of Interactive Data Input

```
                 NUMBER OF     VISIT          AVERAGE          RELATIVE     WORKLOAD
   STATION        SERVERS    FREQUENCY   PROCESSING TIME     WORKLOAD     PER SERVER

 1 MILLING           1        0.12821        31.90000          4.08974      4.08974
 2 DRILLING          1        0.20513        14.60000          2.99487      2.99487
 3 LATHE             1        0.10256        21.65000          2.22051      2.22051
 4 INSPECT           1        0.20513         4.67875          0.95974      0.95974
 5 LOADING           1        0.35897         9.35714          3.35897      3.35897
 6 CARTS             1        0.07692         1.85000          1.85000      1.85000


NUMBER OF ITEMS IN SYSTEM =            4
MEAN NO. OF OPERATIONS TO COMPLETE AN ITEM=     13.00000


PRODUCTION RATE =   0.7471721      /HOUR OR     1307.551     /YEAR.
REQUIRED PRODUCTION =    12000.00     /YEAR.
THE BOTTLENECK MACHINE IS MILLING
AVERAGE FLOW TIME =     321.2112      MINUTES
SYSTEM UTILIZATION =     917.7462      PERCENT
SYSTEM COST IS =     3970.000


ENTER 1 TO REVISE PART SELECTION.
ENTER 2 TO ADD MACHINE(S).
OTHERWISE, ENTER 0
2


ENTER NO. OF ADDITIONAL UNITS OF MACHINE MILLING .
1
ENTER ADDITIONAL COST.
800
```

Figure 11: Typical Response of the System Synthesis Model

machines for the bottleneck operations, or that the production plan is to be revised. This procedure is repeated until the desired production capacity is reached, or when the investment requirement exceeds the budget constraint.

For illustration purposes, two system configurations were selected. The first configuration is based on the maximum expected production requirement in the 10-year planning horizon, while the second configuration is based on the expected production requirement for the first year. The two system configurations are shown in Table 27.

6.5.2 Phase II : Interactive Simulation

Phase II is the stochastic phase. As in Phase I, the computer program prompts the decision-maker for all the input data required to run the program. For each year, the actual production requirement and routing are simulated for each run and Subroutine CAN-Q(0) is called to compute the system performance measures. An example of the performance evaluation report is shown in Figure 12.

At the end of each year, the executive program calls Subroutine CAN-Q(1) to verify if the current system configuration is sufficient for the planned production of the following year. Information on which machine type to add, and the number of units to be added is provided. A typical response is shown in Figure 13.

TABLE 27

Initial System Configurations

	Configuration	
	I	II
Machine Type	Units	Units
Milling	2	5
Drilling	1	4
Turning	1	3
Inspection	1	1
Loading/Unloading	1	4
Computer	1	1
Material Handling	1	2

```
           CONFIGURATION  1 :
                                       CHANCE OF
          MACH TYPE    NO. OF UNITS BOTTLENECK
           milling                2        0.27
           drilling               1        0.00
           turning                1        0.00
           inspect                1        0.00
           loading                1        0.73
           carts                  1        0.00


FOR PART AAAA    ,      40. UNITS MUST BE PROCURED AT $     320.86
FOR PART BBBB    ,      35. UNITS MUST BE PROCURED AT $     298.16
FOR PART CCCC    ,      34. UNITS MUST BE PROCURED AT $     321.52
```

Figure 12: Example of a Performance Evaluation Report

```
TO MEET PLANNED REQUIREMENT...
1 ADDITIONAL UNITS OF milling  ARE REQUIRED.
INDICATE THE NUMBER OF UNITS YOU WISH TO ADD.
1
WHAT IS THE COST OF EACH MACHINE ADDED ?
800
1 ADDITIONAL UNITS OF drilling ARE REQUIRED.
INDICATE THE NUMBER OF UNITS YOU WISH TO ADD.
1
WHAT IS THE COST OF EACH MACHINE ADDED ?
700
0 ADDITIONAL UNITS OF turning  ARE REQUIRED.
INDICATE THE NUMBER OF UNITS YOU WISH TO ADD.
0
0 ADDITIONAL UNITS OF inspect  ARE REQUIRED.
INDICATE THE NUMBER OF UNITS YOU WISH TO ADD.
0
1 ADDITIONAL UNITS OF loading  ARE REQUIRED.
INDICATE THE NUMBER OF UNITS YOU WISH TO ADD.
2
WHAT IS THE COST OF EACH MACHINE ADDED ?
20
0 ADDITIONAL UNITS OF carts    ARE REQUIRED.
INDICATE THE NUMBER OF UNITS YOU WISH TO ADD.
0
```

```
FATHOM ANY CONFIGURATION WHICH ARE NOT WORTHY OF FURTHER CONSIDERATION.

HOW MANY CONFIGURATIONS ARE TO BE    EVALUATED FURTHER ?
4
PLEASE LIST THE CONFIGURATION NUMBERSIN INCREASING ORDER.
ONE LINE FOR EACH CONFIGURATION.
1
2
4
6
```

Figure 13: A Typical Response of the Decision Analysis Model

The decision-maker interacts with the computer by introducing a new system configuration to the decision tree. Also, the decision-maker may fathom one of more existing configuration(s) if it is believed that further evaluation of these configurations are not necessary. Finally, a cash flow summary is provided for each year as shown in Figure 14. At the end of the planning horizon, an "optimal" path is traced. The optimal system configuration set for this example is summarized in Table 28.

The same procedure should be followed in evaluating the conventional job shop alternative.

6.5.3 Confidence Interval for Cost Estimation

The output from simulation are observed samples of random variables. Inferences on the performance of the system should consider the inherent variability of the simulation output.

In the preceeding illustration, the conditional costs were estimated as single numbers commonly referred to as point estimates. As a result of chance variation, a point estimate will likely deviate from the unknown population parameter (costs) being estimated. To measure the accuracy

```
CONFIGURATION  3 :

CAPACITY OF THE FMS                   =   3502.568
NUMBER OF ENGG CHANGES                =          2
CASH FLOW ANALYSIS:
  VARIABLE     COST =   93.39083
  FIXED        COST =   100.0000
  PROCUREMENT COST =    2675.584
  ENGINEERING COST =    15.00000
TOTAL ANNUAL COST                     =   2883.975
LESS : TAX DEDUCTABLES
  FROM ANNUAL COST =    1297.789
  FROM DEPRECIATION=    256.9500
AFTER TAX ANNUAL COST                 =   1329.236
PLUS : INVESTMENT THIS PERIOD         =   4440.000
TOTAL CASH FLOW FOR THIS PERIOD       =   5769.236

NET PV UP TO THIS PERIOD              =   13404.39
```

Figure 14: Example of a Cash Flow Summary

TABLE 28

Optimal System Configurations

Year	Milling	Drilling	Turning	Inspection	Loading	Carts
1	2	1	1	1	1	1
2	3	2	1	1	3	2
3	3	2	1	1	3	2
4-10	5	4	3	1	5	2

of an estimate, the confidence interval provides a probability statement specifying the likelihood that the parameter being estimated falls within the prescribed bounds. Interval estimates, together with the test of hypotheses regarding the expected value of costs for each configuration provide a better decision criteria for fathoming in the decision analysis process.

6.6 SUMMARY

A dynamic decision approach for the justification of a flexible manufacturing system is developed in this paper by imbedding queueing theory (CAN-Q) with simulation in a decision analysis framework. The methodology calls for the decision-maker to interact with the model based on the statistics generated at each stage of analysis.

A two-phase procedure for the economic justification of the FMS is recommended. Phase I is an interactive model based on queueing theory for resolving the part selection and machine requirements planning problems. The system configurations generated in Phase I are evaluated in a decision analysis model which consititutes Phase II. At each decision point the decision-maker interacts with the simulation model by specifying alternative courses of actions to be evaluated. The interactions involve:

1. Fathoming one or more configuration(s) which are not worthy of further considerations, and
2. Generating new system configurations based on knowledge about the current state (year), and projections for the subsequent stage (year).

The decision process is carried out dynamically. The capacity of the flexible manufacturing system may be augmented if the planned requirements projected for the following year exceed current capacity. An after-tax analysis provides the decision maker the performance measure for selecting the best alternative.

Chapter VII

SUMMARY, CONCLUSIONS AND RECOMMENDATION

This study presented some analysis concepts and decision tools for resolving the problems encountered in designing and implementing a flexible manufacturing system. The research area centered on four topics -- system synthesis, machine mapping and pooling, input control, and economic justification. This chapter summarizes the study and presents suggestions for future research.

7.1 SUMMARY AND CONCLUSIONS

Three computer software packages were developed in this study. These are:

1. SIM-Q, an input-generator for simulation modelling was developed to provide a powerful and expedient tool for resolving the material handling system selection, work scheduling, input control, and real time operation problems. The problem of input control was examined using SIM-Q by exploring the viability of operating a flexible manufacturing system as a programmable transfer line. SIM-Q was also used in this study to test the robustness of CAN-Q in modelling an existing FMS.

2. The system synthesis problem can be resolved by applying CAN-Q in an interactive computer program developed in this study. The model provides an integrated approach to the product selection and machine requirements planning problems.
3. A dynamic decision model for the economic justification of FMS was developed. This model imbeds queueing theory with simulation in a decision analysis framework. A hypothetical example was presented as illustration.

The problem of machine mapping and pooling was examined. A zero-one linear programming model was formulated.

7.1.1 System Synthesis

Four FMS design problems -- part pre-selection, part selection, machine selection and requirements planning, and material handling system selection and configuration constitute the System Synthesis Problem. Having determined the manufacturing process, the type of raw materials, and the type of equipment needed, a critical problem in designing and implementating a FMS is that of machine requirements planning. Machine requirements planning is the problem of specifying the number of machine units in each period within some planning horizon, to satisfy a required production output; subject to resource availability (financial, space, manpower, etc.).

The problem of FMS system synthesis has never been addressed in the literature. Chapter four of this study presented a design tool for configuring the system, taking into account production requirement, budget constraint, and the incremental productivity of adding/dropping one or more part type(s) from the production plan.

7.1.2 Machine Mapping and Pooling

The machine mapping/pooling problem has been examined previously by Stecke and Solberg in (STECK80) and (STECK83). The solution techniques presented include both heuristic procedures and mathematical programming (non-linear mixed integer programming models).

A zero-one linear programming formulation of the machine mapping/pooling problem was presented in Chapter four of this study. The model allocates operations and the required tools to the machining stations (machine mapping), and determines how the machining stations are partitioned into machine groups to perform a common set of operations (machine pooling) ;subject to technological and tool capacity constraints.

Padberg et. al. have shown that a combination of problem pre-processing, cutting planes, and clever branch-and-bound techniques permit the efficient solution of sparse large-scale zero-one linear programming problems (see reference PADBM83). The model presented in this study, therefore, has obvious advantage over that of a non-linear mixed integer formulation.

7.1.3 Input Control/Batching

Chapter five presented an analysis of the production flow attributes in FMS. System performance (production rate) was examined using SIM-Q. The correlation of system performance was analyzed with regards to the flow characteristics of the parts in the system, the batching strategies for release of parts, the number of machines in the system, the number of in-process storage stations, the system control, and the variability of processing times.

There are three reasons for batching:

1. tool capacity constraint limits the number of operations requiring dissimilar tools,
2. due-date constraints requires a pre-determined quantity of some part types to be processed first, and
3. for minimizing scheduling-related difficulties and traffic congestion.

The viability of operating the flexible manufacturing system as a single-class job shop or programmable transfer line was examined by investigating various performance dependencies/ settings. It has been shown that for systems with excessive backtracking of workflow, the SCJS/PTL alternative may result in a significantly higher output even if the random FMS is fairly balanced. The viability of the SCJS/PTL, however, hinges on the resulting balance delay of the system after batching. Finally, a methodology for solving the FMS batching problem was presented, combining simulation methodology with mathematical programming.

In the realm of automated manufacturing, the transfer line is a rigid flow shop while the flexible manufacturing system is essentially an automated job shop. The FMS combines the flexibility of the job shop and the low variable cost of the flow shop resulting in improved machine utilization, reduced in-process inventory, and effective material usages. Perhaps the most frequently cited advantage of the FMS is its flexibility (for random routing). Ironically, this flexibility also results in more complex control problems. By transferring certain decisions in the FMS from the operation level to the input level, its flexibility is reduced but simpler/better control is possible.

7.1.4 SIM-Q versus CAN-Q

The computational expediency and portability (CAN-Q can be run on a programmable hand-held calculator) of CAN-Q makes it an ideal tool for many design/planning situations. A statistical test on the robustness of CAN-Q (Chapter three) showed that the errors in CAN-Q may in fact, cancel each other neatly. However, since the user has no control on the compensatory effect of these errors, the use of CAN-Q as an analytical tool requires prudent judgement. Moreover, CAN-Q is insensitive to many control variables (for instance, blocking and locking) which have major impact on the perfomance of the system.

A problem with computer simulation is the amount of time and effort required for information gathering, modelling and computer programming. The advent of simulation languages such as GASP, Q-GERT, DYNAMO, GPSS, SIMULA, and lately, SLAM and SIMAN is a big step towards the simplification of the modelling process. SIM-Q is a further step in providing the system analysts a model building utility program for generating SLAM discrete-network models of manufacturing systems.

To exploit the structural simplicity of CAN-Q, SIM-Q was developed using the conceptual framework of CAN-Q The input data requirement of SIM-Q is essentially the same as in CAN-Q. However, SIM-Q provides the user the flexibility of modelling the system to fit specific system design/analysis requirements. Moreover, the cost of running a simulation model generated by SIM-Q is modest.

7.2 RECOMMENDATION FOR FURTHER STUDY

Apparently, only four of the eight FMS design/implementation issues have been examined directly in this study. The remaining issues merit further examination. This section lists other promising areas for further study.

7.2.1 On CAN-Q

CAN-Q is perhaps the most widely known analytic model for the planning of flexible manufacturing systems. The model is based on some restrictive assumptions which are not expected to be satisfied in real systems. Theoretical developments with regards to the relaxation of these assumptions are available in the literature (See for instance, reference HORDA81 for networks with blocking, or Section 3.1 of KELLP79 for a more general network). A promising area for future research is an extension of CAN-Q, perhaps through numerical analysis.

7.2.2 On SIM-Q

SIM-Q has been developed primarily as a tool to facilitate the simulation modelling requirement of this study. Although a library of USERF functions has been provided in the User's Manual, they are by no means complete. Specific simulation requirement will dictate the content of the various USERF functions and the INTLC data-input routine. A rewarding area for future research is to develop SIM-Q into a programming language for the design, planning and analysis of flexible manufacturing systems.

7.2.3 On Machine Mapping / Pooling

The problem of batching and machine mapping/pooling have been treated as independent decisions in this study. Batching determines the variety of parts in the system which is an input to the machine mapping/pooling problem. Similarly, the machine mapping/pooling problem defines the workstation-to-operations assignments which affects the batching decision. Clearly, the two issues are interactive. An area for future research is to provide a unifying approach for the resolution of these problems, instead of solving the problems sequentially and iteratively.

7.2.4 On Input Control / Batching

No closed-form solution is provided for the batching problem. The batching strategy presented in Chapter five attempts to limit the variety of parts flowing in the FMS at a time, while seeking a balanced workload for all batches. The two objectives are generally conflicting and the trade-off is to be evaluated by simulation (using SIM-Q). One obvious direction for future research is to provide a more stream-lined methodology for optimal batching. The use of surrogate objective functions and/or goal programming techniques is a possibility.

7.2.5 On Economic Justification

The economic justification of a flexible manufacturing system requires a more sophisticated approach to calculate the expected savings than conventional approaches for machine justification. Chapter six presented such an approach. However, the model presented still requires refinements. This include the development of new search strategies to guarantee a successful search, and the introduction of weighting method for the final decision path evaluation. Another area for future research is the incorporation of risk and preference analysis.

Appendix A

A USER'S GUIDE TO SIM-Q

This user's guide presents an introduction to SIM-Q and describes a library of subprograms for various decision/control requirements.

A.1 PROCEDURE DESCRIPTION

The methodology of SIM-Q follows the conceptual framework of CAN-Q. The output of SIM-Q is the network "world view" of a combined discrete-network SLAM (Simulation Language for Alternative Modelling) simulation model. All the housekeeping specification statements needed to run the SLAM model are built-in to the network model.

As in CAN-Q, the flexible manufacturing system is modelled as a closed network of queues (resources). These resources may consist of machine tools, inspection station(s), material handling system (henceforth referred to as the transporters), load/unload stations, temporary storage stations, pallets/fixtures, and manual loaders. Parts enter the system through a load/unload station and go through a sequence of operations before leaving the system. Since the number of entities in a closed network is fixed, it is assumed that a new part is launched whenever a completed part leaves the system. If a workstation (machine tool or load/unload station) is not available when requested by a part, the part may be routed to a temporary storage station. Blocking occurs if either the transporter is busy or the temporary storage stations are full.

A.2 INPUT DATA REQUIREMENT

The users familiar with CAN-Q will find the data requirement of SIM-Q to closely parallel those of CAN-Q. The major difference is the routing data required in CAN-Q which is not part of the SIM-Q input data. The routing data is needed only when running the SIM-Q generated SLAM model.

The first data card is the LABEL card where the user enters his/her name (20 alphanumeric characters), the project title (20 aphanumeric characters), and the date of run (two blanks followed by mm/dd/yy). The second data card is the OPTION card which has three fields in free-format (meaning, the entries are separated by a comma and/or one or more blanks). The user specifies in this card:

1. the length of the simulation,
2. the time to clear statistical arrays, and
3. the option code for releasing workstations.

An option code of "0" releases the workstations immediately after processing while an option code of "1" holds the workstation until the workpiece is ready to be transported by the material handling device to the next station in the routing (i.e., both resources must be available).

The third data card is the PROBLEM SIZE card. This card has seven free-format fields. The user enters:

1. the number of machine tools and inspection stations,
2. the number of material handling devices,
3. the number of dedicated loading/unloading stations,
4. the number of in-process storage stations,
5. the number of types of pallets/fixtures,
6. the number of parts circulating through the system (system loading), and
7. the number of part types.

SIM-Q automatically models the loading/unloading operations as multi-resource operations if the number of types of pallets/fixtures is not zero.

The next cards are the RESOURCE SPECIFICATION cards defining the name of the resources, the number of parallel servers of each resource, and the dispatching rule associated with each resource. The format of the input data is given in Table 29.

Finally, the PART NAME cards input the name of each part type. Each card consists of eight alphanumeric characters for each part type.

TABLE 29

SIM-Q Resource Specification Card

Columns	Meaning
1- 8	name of the resource
11-15	number of parallel servers
16-20	sequencing rule :
	=1 based on value of ATRIB(6)* high value first.
	=2 based on the value of ATRIB(6), low value first. (ATRIB(6) is defined in FUNCTION USERF for each individual part)
	=3 based on the last-in-first-out (LIFO) sequencing rule.
	=4 based on the first-come-first-serve (FIFO) sequencing rule.

A.3 THE SIM-Q PROGRAM LISTING

The following is the program listing for SIM-Q:

```
      DIMENSION PRIORI(4),PRINUM(4)
      DATA PRIORI/'HVF(','LVF(','LIFO','FIFO'/
      DATA PRINUM/'6)  ','6)  ','    ','    '/
      DIMENSION NAME(50),NS(50),IPRTY(50)
      DIMENSION USER(5),LABEL(5),DATE(2)
      DOUBLE PRECISION NAME,PNAME(50)
C
C  DATA INPUT FOR "SIM-Q"
C
C  ---------- LABEL CARD ----------
      READ(5,1)USER,LABEL,DATE
1     FORMAT (5A4,5A4,2X,2A4)
C  --------- OPTION CARD ----------
      READ(5,*)TTFIN,TFRST,IOPT
C  ------ PROBLEM SIZE CARD -------
      READ(5,*)M,MT,MD,MU,MP,MNTRY,NPART
C  - RESOURCE SPECIFICATION CARD--
      MM=0
      IF(MP.GT.0)MM=1
      N1=1
      N2=M+MT+MD+MU+MP+MM
C  READ THE NAME,NO. OF SERVER, & PRIORITY RULE FOR
C  THE WORKSTATIONS, TRANSPORTERS, LOAD/UNLOAD STATIONS,
C  IN-PROCESS STORAGE, PALLET/FIXTURES, & THE MANUAL LOADER
      DO 2 I=N1,N2
2     READ(5,3)NAME(I),NS(I),IPRTY(I)
3     FORMAT (A8,2X,2I5)
C  ------- PART NAME CARD ---------
      READ(5,4) (PNAME(I),I=1,NPART)
4     FORMAT(A8)
      WRITE(6,10)
10    FORMAT(' ;   ****************************************************'
     Z      /' ;   **                                                **'
     1      /' ;   **                S  I  M   -  Q                  **'
     2      /' ;   **                                                **'
     3      /' ;   **   SLAM INTEGRATED MODEL FOR QUEUEING NETWORKS  **'
     4      /' ;   **                                                **'
     5      /' ;   ****************************************************'
     6/' ;'/' ;       D E F I N I T I O N   O F   C O D E S       '/' ;'
     7/' ; ATRIB(1) - PART TYPE'
     8,T38,'ATRIB(2) - TIME ENTERED SYSTEM'
     9,/' ; ATRIB(3) - OPERATION NUMBER'
     Z,T38,'ATRIB(4) - PART''S STATION'
     1,/' ; ATRIB(5) - PART''S LOCATION'
     2,T38,'ATRIB(6) - PART''S PRIORITY CODE'
     3,/' ; ATRIB(7) - PALLET TYPE USED'
     4,T38,'ATRIB(8) - PART''S NEXT STATION'
     5,/' ; ATRIB(9) - PART''S NEXT LOCATION'
     6,T38,'ATRIB(10)- TRANSPORTER SELECTED'
     7,/' ; XX(I)    - LOCATION OF RESOURCE I'/' ;')
```

```
      WRITE(6,11)USER,LABEL,DATE
11    FORMAT (1X,'GEN,',5A4,',',5A4,',',2A4,',1,YES,NO,YES,NO;')
      WRITE(6,12)
12    FORMAT(' ;'/' ;        U S E R F   A N D   E V E N T        '/' ;'
     1/' ; USERF(1) - RETURNS THE PART TYPE'
     2,/' ; USERF(2) - RETURNS THE SERVICE TIME'
     3,/' ; USERF(3) - RETURNS THE MOVING TIME'
     4,/' ; USERF(4) - DETERMINES IF VISIT IS REQUIRED'
     5,/' ; USERF(5) - RETURNS THE PART''S CURRENT LOCATION'
     6,/' ; USERF(6) - RETURNS THE PART''S PRIORITY CODE'
     7,/' ; USERF(7) - RETURNS THE PALLET TYPE USED'
     8,/' ; USERF(8) - RETURNS THE PART''S NEXT STATION'
     8,/' ;            RETURNS 0  IF ALL OPERATIONS COMPLETED'
     8,/' ;            RETURNS 99 BRANCHES TO "TIME DELAY NODE"'
     9,/' ; USERF(9) - RETURNS THE PART''S NEXT LOCATION'
     A,/' ; USERF(10)- SELECTS TRANSPORTER'/' ;'
     B,/' ; EVENT(1) TO EVENT(5) : FREES (1) A PALLET/FIXTURE, (2) A'
     C,/' ;    WORKSTATION, (3) A TRANSPORTER, (4) A TEMPORARY L/UL'
     D,/' ;    STATION, OR (5) A MANUAL LABOUR, RESPECTIVELY'/' ;')
      MFIL=N2
      MATR=10
      MXNTRY=MNTRY
      IF(MXNTRY.LT.100) MXNTRY=100
      WRITE(6,13)MFIL,MATR,MXNTRY
13    FORMAT (1X,'LIMITS,',I2,',',I2,',',I3,';'
     1/1X,';'/1X,';       S E Q U E N C I N G  R U L E S       '/1X,';')
      NN=1
14    IF((N2-NN+1).LT.5)GOTO 16
      NNN=NN+4
      WRITE(6,15)(I,PRIORI(IPRTY(I)),PRINUM(IPRTY(I)),I=NN,NNN)
15    FORMAT(1X,'PRIORITY',5('/',I2,',',A4,A2),';')
      NN=NN+5
      GOTO 14
16    IF((N2-NN+1).LT.4)GOTO 18
      NNN=NN+3
      WRITE(6,17)(I,PRIORI(IPRTY(I)),PRINUM(IPRTY(I)),I=NN,NNN)
17    FORMAT(1X,'PRIORITY',4('/',I2,',',A4,A2),';')
      NN=NN+4
      GOTO 14
18    IF((N2-NN+1).LT.3)GOTO 20
      NNN=NN+2
      WRITE(6,19)(I,PRIORI(IPRTY(I)),PRINUM(IPRTY(I)),I=NN,NNN)
19    FORMAT(1X,'PRIORITY',3('/',I2,',',A4,A2),';')
      NN=NN+3
      GOTO 14
20    IF(N2.LE.NN)GOTO 22
      NNN=NN+1
      WRITE(6,21)(I,PRIORI(IPRTY(I)),PRINUM(IPRTY(I)),I=NN,NNN)
21    FORMAT(1X,'PRIORITY',2('/',I2,',',A4,A2),';')
      NN=NN+2
```

```
      GOTO 14
22    IF(NN.GT.N2)GOTO 24
      WRITE(6,23)(I,PRIORI(IPRTY(I)),PRINUM(IPRTY(I)),I=NN,NNN)
23    FORMAT(1X,'PRIORITY/',I2,',',A4,A2,';')
24    CONTINUE
      WRITE(6,25)
25    FORMAT (1X,';'/1X,'NETWORK;'/1X,';'
     1/1X,';       D E C L A R E   R E S O U R C E S        '/1X,';')
      DO 27 I=1,N2
      WRITE(6,26)NAME(I),NS(I),I
26    FORMAT (T8,'RESOURCE/',A8,'(',I2,'),',I2,';')
27    CONTINUE
      WRITE(6,28) MNTRY
28    FORMAT(1X,';'
     */' ;- - - -   D E F I N E   A T T R I B U T E S     - - - -'/' ;'
     1/T8,'CREATE,0,0,,',I3,';'
     2/T8,'ACT;'
     3/1X,'ENTR  ASSIGN,ATRIB(1)=USERF(1) , ATRIB(2)=TNOW , ATRIB(3)=1,'
     4/T15,'ATRIB(7)=      0.0,ATRIB(8)=USERF(8),ATRIB(9)=USERF(9),'
     5/T15,'ATRIB(4)=ATRIB(8),ATRIB(5)=ATRIB(9),ATRIB(6)=USERF(6),1;'
     6/T8,'ACT;')
      NT=M+MT+MD
      WRITE(6,30) M
30    FORMAT(' ;'/' NEXT  GOON,1;'
     1/' ;- - - -     T  R  A  N  S  I  T  I  O  N        - - - -'/' ;'
     2/T8,'ACT,,ATRIB(8).LE.',I2,',MACH;')
      IF(MP.GT.0.OR.MU.GT.0)WRITE(6,31)
      IF(MU.GT.0)WRITE(6,32)NT
      WRITE(6,33)
31    FORMAT(T8,'ACT,,ATRIB(8).EQ.99,DLAY;')
32    FORMAT(T8,'ACT,,ATRIB(8).GT.',I2,',TEMP;')
33    FORMAT(T8,'ACT,,,LOAD;')
      WRITE(6,40)
40    FORMAT(' ;'/' MACH  GOON,1;'
     */' ;- - - -  M  A  C  H  I  N  E   T  O  O  L  S - - - -'/' ;'
     1/T8,'ACT,,USERF(4).EQ.0,UPDT;')
      DO 43 J=1,M
      IF(J.GE.10) WRITE(6,41)J,J
      IF(J.LT.10) WRITE(6,42)J,J
41    FORMAT(T8,'ACT,,ATRIB(8).EQ.',I2,',WS',I2,';')
42    FORMAT(T8,'ACT,,ATRIB(8).EQ.',I2,',WS',I1,';')
43    CONTINUE
      DO 46 J=1,M
      IF(J.GE.10) WRITE(6,44) J,J,NAME(J)
      IF(J.LT.10) WRITE(6,45) J,J,NAME(J)
44    FORMAT(1X,'WS',I2,'  AWAIT(',I2,'),',A8,'/1,1;'/T8,'ACT,,,TRAN;')
45    FORMAT(1X,'WS',I1,'   AWAIT(',I2,'),',A8,'/1,1;'/T8,'ACT,,,TRAN;')
46    CONTINUE
      WRITE(6,50)
50    FORMAT(' ;'/' LOAD  GOON,1;'
     */' ;- - - -     L O A D I N G   S T A T I O N S     - - - -'/' ;'
```

```
     1/T8,'ACT,,USERF(4).EQ.0,UPDT;')
      NL1=M+MT+1
      NL2=M+MT+MD
      DO 53 J=NL1,NL2
      IF(J.GE.10) WRITE(6,41)J,J
      IF(J.LT.10) WRITE(6,42)J,J
53    CONTINUE
      DO 56 J=NL1,NL2
      IF(J.GE.10) WRITE(6,54) J,J,NAME(J)
      IF(J.LT.10) WRITE(6,55) J,J,NAME(J)
54    FORMAT(1X,'WS',I2,'  AWAIT(',I2,'),',A8,'/1,1;'/T8,'ACT,,,PALL;')
55    FORMAT(1X,'WS',I1,'   AWAIT(',I2,'),',A8,'/1,1;'/T8,'ACT,,,PALL;')
56    CONTINUE
      IF(MU.LE.0)GOTO 61
      WRITE(6,60)
60    FORMAT(' ;'/' TEMP  GOON,1;'
     1/' ;- - - -   T E M P O R A R Y   S T O R A G E    - - - -'/' ;')
      NL1=M+MT+MD+1
      NL2=M+MT+MD+MU
      DO 63 J=NL1,NL2
      IF(J.GE.10) WRITE(6,41)J,J
      IF(J.LT.10) WRITE(6,42)J,J
63    CONTINUE
      DO 66 J=NL1,NL2
      IF(J.GE.10) WRITE(6,44) J,J,NAME(J)
      IF(J.LT.10) WRITE(6,45) J,J,NAME(J)
66    CONTINUE
61    CONTINUE
      IF(MP.GT.0.OR.MU.GT.0) WRITE(6,70)
70    FORMAT(' ;'/' DLAY  GOON,1;'
     */' ;- - - -     D  E  L  A  Y        N  O  D  E      - - - -'/' ;'
     1/T8,'ACT,1;  DELAY BY 1 TIME UNIT')
      IF(MP.GT.0.OR.MU.GT.0)WRITE(6,71)
71    FORMAT(1X,'RDEF  ASSIGN,ATRIB(8)=USERF(8),ATRIB(9)=USERF(9),1;'
     3/T8,'ACT,, ATRIB(3).NE.1,NEXT;'
     4/T8,'ACT;'
     5/1X,'       ASSIGN,ATRIB(4)=ATRIB(8),ATRIB(5)=ATRIB(9),1;'
     6/T8,'ACT,,,NEXT;')
      IF(MP.NE.0)GOTO 73
      WRITE(6,72)
72    FORMAT(' ;'/' PALL  GOON,1;'/T8,'ACT,,,TRAN;')
      GOTO 74
73    WRITE(6,80)
80    FORMAT(' ;'/' PALL  GOON,1;'
     */' ;- - - -     P  A  L  L  E  T  I  Z  I  N  G      - - - -'/' ;'
     1/T8,'ACT,,ATRIB(7).NE.USERF(7),PALT;'
     2/T8,'ACT,,,TRAN;'
     3/1X,'PALT  GOON,1;')
      WRITE(6,81) N2,NAME(N2)
81    FORMAT(T8,'ACT;'/T8,'AWAIT(',I2,'),',A8,'/1,1;')
```

```
82    DO 86 I=1,MP
83    IF(I.GE.10) WRITE(6,84) I,I
      IF(I.LT.10) WRITE(6,85) I,I
84    FORMAT(T8,'ACT,,USERF(7).EQ.',I2,',PL',I2,';')
85    FORMAT(T8,'ACT,,USERF(7).EQ.',I1,',PL',I1,';')
86    CONTINUE
      DO 89 I=1,MP
      J=I+M+MT+MD+MU
      IF(I.GE.10) WRITE(6,87) I,J,NAME(J)
      IF(I.LT.10) WRITE(6,88) I,J,NAME(J)
87    FORMAT(1X,'PL',I2,'  AWAIT(',I2,'),',A8,'/1,1;'
     1/T8,'ACT,,,TRAN;')
88    FORMAT(1X,'PL',I1,'   AWAIT(',I2,'),',A8,'/1,1;'
     1/T8,'ACT,,,TRAN;')
89    CONTINUE
74    NFT=M+MT+MD
      WRITE(6,90)
90    FORMAT(' ;'/' TRAN  GOON,1;'
     */' ;- - - -     T R A N S P O R T E R S    - - - -')
      IF(MU.GT.0)WRITE(6,120)NFT
120   FORMAT(' ;'/T8,'ACT,,ATRIB(4).GT.',I2,',FTEM;')
      IF(IOPT.EQ.1)WRITE(6,121)
121   FORMAT(T8,'ACT,,ATRIB(3).NE.1,FRWS;')
      IF(MU.GT.0.OR.IOPT.EQ.1)WRITE(6,122)
122   FORMAT(T8,'ACT,,NOFT;')
      IF(IOPT.EQ.1)WRITE(6,123)
123   FORMAT(' FRWS  EVENT,2,1;  FREE OLD WORKSTATION'/T8,'ACT,,NOFT;')
      IF(MU.GT.0)WRITE(6,124)
124   FORMAT(' FTEM  EVENT,4,1;  FREE TEMPORARY STORAGE'/T8,'ACT;')
      IF(MU.GT.0.OR.IOPT.EQ.1)WRITE(6,125)
125   FORMAT(' NOFT  GOON,1;')
98    WRITE(6,99)
99    FORMAT(T8,'ACT,,ATRIB(3).EQ.1,OPER;'
     2/T8,'ACT;'
     3/T8,'ASSIGN,ATRIB(10)=USERF(10),1;')
      DO 93 I=1,MT
      IF(I.GE.10) WRITE(6,91)I,I
      IF(I.LT.10) WRITE(6,92)I,I
91    FORMAT(T8,'ACT,,ATRIB(10).EQ.',I2,',MH',I2,';')
92    FORMAT(T8,'ACT,,ATRIB(10).EQ.',I2,',MH',I1,';')
93    CONTINUE
      DO 97 I=1,MT
      J=M+I
      IF(I.GE.10) WRITE(6,94) I,J,NAME(J)
      IF(I.LT.10) WRITE(6,95) I,J,NAME(J)
94    FORMAT(1X,'MH',I2,'  AWAIT(',I2,'),',A8,'/1,1;')
95    FORMAT(1X,'MH',I1,'   AWAIT(',I2, '),',A8,'/1,1;')
      WRITE(6,96) J
96    FORMAT(T8,'ACT,USERF(3);  ADVANCE MOVING TIME'
     1/T8,'ASSIGN,XX(',I2,')=ATRIB(9),ATRIB(4)=ATRIB(8),'
     2/T8,'                             ATRIB(5)=ATRIB(9),1;'/T8,'ACT;'
```

```
     3/T8,'EVENT,3,1;  FREE TRANSPORTER'/T8,'ACT,,,OPER;')
97    CONTINUE
      WRITE(6,100)
100   FORMAT(' ;'/' OPER  GOON,1;'
     1/' ;- - - -       P  R  O  C  E  S  S  I  N  G        - - - -'
     2/' ;')
      IF(MU.GT.0)WRITE(6,101) NT
101   FORMAT(T8,'ACT,,ATRIB(8).GT.',I2,',RDEF;'
     1/T8,'ACT;'/T8,'GOON,1;')
      DO 104 IWS=1,M
      IF(IOPT.EQ.1) WRITE(6,102) IWS,IWS
102   FORMAT(T8,'ACT/',I2,',USERF(2),ATRIB(8).EQ.',I2,',PROC;')
      IF(IOPT.EQ.0) WRITE(6,103) IWS,IWS
103   FORMAT(T8,'ACT/',I2,',USERF(2),ATRIB(8).EQ.',I2,',RELS;')
104   CONTINUE
      WRITE(6,105)
105   FORMAT(T8,'ACT,USERF(2);     ADVANCE TIME')
      IF(IOPT.EQ.1) WRITE(6,106)
106   FORMAT(' PROC  GOON,1;')
      IF(IOPT.EQ.0) WRITE(6,107)
107   FORMAT(' RELS  EVENT,2,1;  FREE WORKSTATION')
      IF(MP.NE.0)WRITE(6,108)
108   FORMAT(T8,'ACT,,USERF(7).EQ.ATRIB(7),UPDT;'
     1/      T8,'ACT,,ATRIB(3).EQ.1,       FRML;'/T8,'ACT;'
     2/'        EVENT,1,1;  FREE  OLD  PALLET'/T8,'ACT;'
     3/' FRML  EVENT,5,1;  FREE MANUAL LABOR'/T8,'ACT;'
     4/T8,'ASSIGN,ATRIB(7)=USERF(7),1;')
      WRITE(6,109)
109   FORMAT(T8,'ACT;'
     1/1X,'UPDT  ASSIGN,ATRIB(3)=ATRIB(3)+1,ATRIB(6)=USERF(6),'
     2/T8,'       ATRIB(8)=USERF(8) , ATRIB(9)=USERF(9),1;'
     4/T8,'ACT,,ATRIB(8).EQ.0.0,FINI;'
     5/T8,'ACT,,,NEXT;')
      WRITE(6,110)
110   FORMAT(' ;'/' FINI  GOON,1;'
     */' ;- - - -  C O L L E C T  S T A T I S T I C S  - - - -'/' ;')
      IF(MP  .NE.0)WRITE(6,118)
      IF(IOPT.EQ.1)WRITE(6,119)
      DO 113 I=1,NPART
      IF(I.GE.10) WRITE(6,111) I,I
      IF(I.LT.10) WRITE(6,112) I,I
111   FORMAT(T8,'ACT,,ATRIB(1).EQ.',I2,',CL',I2,';')
112   FORMAT(T8,'ACT,,ATRIB(1).EQ.',I2,',CL',I1,';')
113   CONTINUE
      DO 116 I=1,NPART
      IF(I.GE.10) WRITE(6,114)I,PNAME(I)
      IF(I.LT.10) WRITE(6,115)I,PNAME(I)
114   FORMAT(1X,'CL',I2,'  COLCT,INT(2),',A8,'FLOW TIME,,1;'
     1/T8,'ACT,,,ENTR;')
115   FORMAT(1X,'CL',I1,'   COLCT,INT(2),',A8,'FLOW TIME,,1;'
```

```
     1/T8,'ACT,,,ENTR;')
116   CONTINUE
      WRITE(6,117) TTFIN,TFRST
117   FORMAT(T8,'ENDNETWORK;'
     1/1X,'INIT,0,',F6.0,';'
     2/1X,'MONTR,CLEAR,',F6.0,';'
     3/1X,'FIN;')
118   FORMAT(T8,'ACT;'/T8,'EVENT,1,1;  FREE LAST PALLET/FIXTURE')
119   FORMAT(T8,'ACT;'/T8,'EVENT,2,1;  FREE LAST WORKSTATION')
      STOP
      END
```

A.4 THE SIM-Q NETWORK MODEL

The output of SIM-Q is a self documented network simulation model. The first part of the program defines the attributes of the system and describes the purpose of the user-supplied subprograms EVENT and USERF. The second part of the output is the network model itself.

The last part of the SIM-Q output specifies the length of the simulation run and the time for clearing the file and array statistics. The purpose of clearing statistics is to weed out erratic transient responses. If clearing of statistics is not desired, the user can specify the time of clearing to be zero.

A.5 THE TIME DELAY NODE

The time delay node is a hypothetical node (no transporting time and no processing time involved) created to facilitate the modelling of multi-resource operations (for example, palletizing). If one or more required resources for an operation is not available, the part may be routed to a temporary load/unload station. If the part is currently in a temporary storage station, or if the current operation is the initial (launching) operation, the part may be routed to the time delay node by returning a value of "99" for USERF(8). The time delay node will hold the part for one time unit before returning control to FUNCTION USERF.

A.6 SIMULATION OUTPUT REPORTS

The SLAM output report consists of a general section followed by a section for the statistical results. The general section identifies the project title, the user's name, the month/day/year, the length of the simulation run, and the time when the statistical arrays were cleared. The statistical results are categorized by types as follows:

1. The sojourn times (flow times) of completed parts of each part type are summarized in the Flow Time Statistics section of the output report. The mean value, the standard deviation, the coefficient of variation, minimum value, maximum value, and the number of completed parts for each part type are summarized.
2. The Waiting Line Statistics section of the output report summarizes the queue length (the number of parts waiting) statistics for each resource. The average number of parts waiting, the standard deviation, the maximum number of parts waiting, the current number of parts waiting (at the end of the simulation run), and the average waiting time are reported.
3. The Resource Utilization Statistics reports the average utilization, the standard deviation, the maximum utilization, and the current utilization (at the end of the simulation run) of each resource. Two sets of

statistics are provided. The SLAM RESOURCE UTILIZATION statistics gives the average time that each resource is occupied by a part. The REGULAR ACTIVITY STATISTICS gives the fraction of time each workstation is actually processing a part.

A.7 DISCRETE WORLD-VIEW INTERFACE

The network model is interfaced with three user-supplied FORTRAN subprograms. The routing information, processing times, pallet/fixture requirement for each operation, probability of visits, location of resources, pool number (if some workstations are pooled), the probability of failure and time of repair (if reliability is to be incorporated into the model), the speed of the transporter(s), and other pertinent data are inputted through subroutine INTLC.

Subroutine EVENT is called to release a resource when it is no longer needed (service completion). Five events can be modelled. Events number 1, 2, 3, 4 and 5 release a pallet/fixture, a workstation, a transporter, a temporary storage station, or a manual loader, respectively.

Function subprogram USERF comprises of ten functions which provide the user the flexibility in modelling the network to fit specific simulation requirement. USERF(1) controls the part mix of the system by determining the part type of an entering part. USERF(2) and USERF(3) model the processing time and the transporting time, respectively. Modelling of traffic interference can be incorporated into USERF(3). USERF(4) simulates whether a particular operation will be carried out or bypassed. USERF(5) returns the current location of the part. USERF(6) returns the priority code of a particular part. The priority codes may be computed or inputted through Subprogram INTLC. USERF(7) returns the pallet/fixture type required for the current operation.

USERF(8) identifies the workstation a particular part will be visiting next, and USERF(9) returns the location of this workstation. If the current operation is the last operation, USERF(8) returns a value of zero. This is accomplished by adding a dummy operation at the end of each routing.

Finally, USERF(10) returns the transporter number chosen to carry a particular part at a particular time. Decision criteria such as proximity (the distance between the transporter and the part), availability, and queue length (number of parts waiting to be moved) can be modelled easily.

A.8 FUNCTION USERF

This section outlines the modelling of the user-supplied function subprogram USERF. The examples presented are not by any means comprehensive. Specific simulation requirement will dictate the content of the various USERF functions.

The resources must be numbered in the following order: machine tools/inspection stations first, followed by the transporter, then the dedicated load/unload stations, then the temporary storage stations (if any), and finally, the pallets/fixtures and the manual loaders. Through the SIM-Q resource specification cards, each resource may be modelled as a single server or as a multi-server resource.

All user-defined variable must be specifed in a user-common statement. The user-common statement must be named and it must be present in all three user-supplied subprograms.

A.8.1 Launching of Parts

USERF(1) returns the type of a part entering the network (FMS). The part type is stored in the first entry of the part's attribute vector ATRIB.

Illustration 1.1 Equal Proportion of Parts

To model a flexible manufacturing system where "N" part types of equal proportion are processed, the following FORTRAN codes may be written:

```
      DATA KOUNT/0/
```

```
11     KOUNT=KOUNT+1
       USERF=KOUNT
       IF(USERF.LE.N)RETURN
       KOUNT=0
       GOTO 11
```

Illustration 1.2 Unequal Proportion of Parts

If the part types are not of equal proportion, the user may input the percentage part mix (proportion) through subroutine INTLC and compute the cumulative distribution function as follows:

```
12     READ (5,*) (PMIX(I),I=1,N)
       DO 121 I=2,N
121    PMIX(I)=PMIX(I)+PMIX(I-1)
```

The following FORTRAN codes may be written for USERF(1):

```
11     RNUM=DRAND(1)
       USERF=1
       N1=N-1
       DO 121 I=1,N1
       IF(RNUM.LE.PMIX(I))GOTO 122
       USERF=I+1
121    CONTINUE
122    RETURN
```

DRAND(1) is a SLAM function. The function generates a pseudo-random number between 0 and 1. The argument 1 in DRAND(1) specifies that the first of ten available random number streams is used.

Illustration 1.3 Part Type Determined by Time of Entry

Suppose it is desired to control the part mix over time. Consider the simple case where all parts entering the system before time=9999 are to be type 1 parts and all parts entering subsequently are to be part 2 types. This is easily modelled as:

```
13     USERF=1
       IF(TNOW.GT.9999)USERF=2
       RETURN
```

TNOW is a SLAM variable which keeps track of the current clock time.

A.8.2 Processing Times Distribution

Modelling the stochastic nature of one or more elements in a system is an important aspect of simulation modelling. SLAM provides the user a number of functions for modelling stochastic processes. A list of such functions is summarized in Table 7-1 of Reference (PRITA79).

Illustration 2.1 Exponentially Distributed Processing Times

To model exponential processing times, the user may input the routing information and the mean processing times through subroutine INTLC as follows:

```
21     DO 211 I=1,N
       READ(5,*) MAX
       DO 212 J=1,MAX
       READ(5,*) JROUTE(I,J), TIME(I,J)
212    CONTINUE
       JROUTE(I,MAX+1)=0
211    CONTINUE
```

Where MAX= the total number of operations for each part type; JROUTE(I,J)= workstation the part of type I will visit for its Jth operation; and TIME(I,J)= the mean processing time for the Jth operation of type I parts. Notice that a dummy operation has been added. The dummy operation indicates that the immediate preceeding operation is the last operation.

In SIM-Q, the operation pointer (indicating the current stage of operation) is stored as the third entry of the attribute vector ATRIB.

For this example, USERF(2) may be written as follows:

```
21     ITYPE=INT(ATRIB(1))
       IOPER=INT(ATRIB(3))
       XMEAN=TIME(ITYPE,IOPER)
       USERF=EXPON(XMEAN,2)
       RETURN
```

EXPON is a SLAM function which returns an exponentially distributed random variate from a distribution whose mean is XMEAN. The second argument of EXPON indicates that the random number is generated from the second of ten available streams of random numbers.

Illustration 2.2 Modelling Machine Breakdowns

It is very easy to incorporate reliability into the model. The probability of failure and the mean repair time may be input through subroutine INTLC as follows:

```
22     DO 221 I=1,N
```

```
      READ(5,*) PFAIL(I), REPAIR(I)
221   CONTINUE
```

The following codes may be added to the program segment of USERF(2) in the preceeding example (just before the RETURN statement):

```
22    MACHNE=JROUTE(ITYPE,IOPER)
      IF(DRAND(2).GT.PFAIL(MACHNE))RETURN
      YMEAN=REPAIR(MACHNE)
      USERF=USERF+ EXPON(YMEAN,2)
```

In this example, repair time is modelled as an exponentially distributed random process.

A.8.3 Material Handling Process

USERF(3) returns the time it takes for the material handling device to fetch and deliver a part to its destination.

Illustration 3.1 Simple Random Process/Constant Times

If transporting times are assumed constant or as a random processes independent of the (a) type of the carrier, (b) location of the part, (c) location of the carrier, and (d) destination of the part, USERF(3) may be modelled as follows:

```
31    USERF= constant
      RETURN
        or
31    USERF=EXPON(constant,3)
      RETURN
```

Illustration 3.2 Location Zones and Carrier Speed

SIM-Q keeps track of the location of the various resources as well as the current and next location of each part in the system. The locations are indicated by zone numbers. For example, milling machine number 4 is in zone 45, lathe number 1 is in zone 2, etc. The zone number of all stationary resources and the initial zone number of all transporters must be input through subroutine INTLC. Let NR= total number of resources, excluding the pallets/fixtures and the manual loaders. The following READ statement may be written for subroutine INTLC:

```
32    READ(5,*) (XX(J),J=1,NR)
```

Where XX(I) stores the location of resource I.

The user has the option to model the material handling activities as simple as in Illustration 3.1, or as elaborate as the simulation requirement calls for. The following example models the transporting times as a function of the carrier speed, the relative positions of the carrier, the current part location, and the part's next location.

```
32    DEST=ATRIB(9)
      ORIG=ATRIB(5)
      ICAR=INT(ATRIB(10))
      TLOC=XX(ICAR)
      USERF=ABS(TLOC-ORIG)/SPEED(ICAR)
         +ABS(DEST-ORIG)/SPEED(ICAR)
      RETURN
```

ATRIB(5) and ATRIB(9) store the current and next location of the part, respectively. The resource number of the transporter selected to carry the part is stored in ATRIB(10). SPEED(ICAR) is the speed of the selected transporter. The dimension of SPEED(ICAR) is "number of zones per unit time". XX(I), defined earlier, is the location of resource number I. The transporter is selected through USERF(10).

Illustration 3.3 Modelling Traffic Interference

Suppose it is desired to model traffic interference. Consider the Caterpillar (see reference MAYER76) system where two carts must run on a common rail. Traffic congestion occurs whenever the path of the carrier is blocked by the other cart. The following program segment tests the presence of traffic congestion:

```
33     TLOC=XX(ICAR1)
       DEST=ATRIB(9)
       ORIG=ATRIB(5)
       ONOW=XX(ICAR2)
       BLK=0
       IF(ONOW.LT.TLOC.AND.ONOW.GE.ORIG) BLK=1
       IF(ONOW.GT.TLOC.AND.ONOW.LE.ORIG) BLK=1
       IF(ONOW.LT.ORIG.AND.ONOW.GE.DEST) BLK=1
       IF(ONOW.GT.ORIG.AND.ONOW.LE.DEST) BLK=1
```

Where BLK= 1 if blocking occurs and =0, otherwise; ICAR1= the resource number of the selected carrier and ICAR2= the resource number of the other cart. Notice that in the above model, it is assumed that the other cart is stationary while the selected cart is moving. If both carts are moving simultaneously, then it will be necessary to keep track of the position of both carts at any time.

Traffic congestions increase total transporting times. If blocking occurs, the transporting times may be incremented by inserting the following:

```
IF(BLK.EQ.1) USERF=USERF+EXPON(TBLK,3)
```

Where TBLK is the expected value of the increase in transporting time brought about by traffic congestion. TBLK must be input through subroutin INTLC. The traffic congestion process is modelled as an exponential process.

A.8.4 Visit Frequency

Sometimes, a particular operation may be bypassed. An example of this is the in-process inspection operation where 100 percent inspection is not required. The user can input the probability of visit, PVISIT, together with the routing information through subroutine INTLC.

Illustration 4.1 Visit Frequency

The FORTRAN code may be written as follows:

```
41     DO 411 I=1,N
       READ(5,*) MAX
       DO 411 J=1,MAX
       READ(5,*) JROUTE(I,J),TIME(I,J),PVISIT(I,J)
411    CONTINUE
```

USERF(4) returns a value of "0" if the current operation is to be bypassed, and a value of "1" if otherwise. The following FORTRAN codes may be written for USERF(4):

```
41     USERF=1
       I=INT(ATRIB(1))
       J=INT(ATRIB(3))
       IF(DRAND(4).GT.PVISIT(I,J))USERF=0
       RETURN
```

A.8.5 Current Location of Parts

USERF(5) returns the current location of the part.

Illustration 5.1 Parts' Location

The FORTRAN code consists of the following:

```
51     ISTA=INT(ATRIB(4))
       USERF=XX(ISTA)
       RETURN
```

Where ATRIB(4) stores the resource number of the current workstation. XX(I) stores the location of resource I.

A.8.6 Sequencing Rules

USERF(6) returns the priority code of the part at every stage of the operation. The priority code is needed whenever the user specifies either "1" (for the "high value first" sequencing rule) or "2" (for the "low value first" sequencing rule) in the SIM-Q resource specification cards.

Illustration 6.1 The Seniority Rule

If priority is based on the time of entry to the system, the earliest part to enter the system (i.e., the part that has been in the system longest) gets the highest priority. The following FORTRAN codes model this case:

```
61     USERF=ATRIB(2)
       RETURN
```

Where ATRIB(2) stores the time of entry of the part. The sequencing rule specified in SIM-Q is "2" for "low value first".

Illustration 6.2 The Shortest Processing Time (SPT) Rule

By specifying "2" (low value first) in the SIM-Q resource specification card, the following may be written for USERF(6) to model the shortest processing time (SPT) rule:

```
62     ITYPE=INT(ATRIB(1))
       IOPER=INT(ATRIB(3))
       USERF=TIME(ITYPE,IOPER)
       RETURN
```

Illustration 6.3 The Random Sequencing Rule

The random sequencing rule can be modelled by specifying either "1" or "2" in the SIM-Q resource specification cards and using the following codes for USERF(6):

```
63     USERF=DRAND(6)*100
       RETURN
```

Illustration 6.4 The Shortest Remaining Processing Time Rule

The following FORTRAN codes model the shortest remaining processing time rule:

```
64     ITYPE=INT(ATRIB(1))
       IOPER=INT(ATRIB(3))
       USERF=0
       DO 641 J=IOPER,100
       IF(JROUTE(ITYPE,J).EQ.0)GOTO 642
       USERF=USERF+TIME(ITYPE,J)
641    CONTINUE
642    RETURN
```

The SIM-Q resource specification card for the above example should specify "2" for "low value first".

Illustration 6.5 The Number of Remaining Operations Rule

Instead of using the remaining processing time rule, the user may model the sequencing rule based on the stage of completion or the number of remaining operations. By specifying "1" (high value first) for the former, and "2" (low value first) for the latter. The following codes model these sequencing rules:

```
65     USERF=ATRIB(3)
       RETURN
             or
65     ITYPE=INT(ATRIB(1))
       IOPER=INT(ATRIB(3))
       USERF=0
       DO 651 J=IOPER,100
       IF(JROUTE(ITYPE,J).EQ.0)GOTO 652
       USERF=USERF+1
651    CONTINUE
652    RETURN
```

Illustration 6.6 The Cycle Time of the Next Station Rule

The Caterpillar system in Peoria, Illinois uses this sequencing rule. To model this, the expected cycle time for each workstations is computed in subroutine INTLC as follows:

```
      DO 66 I=1,NRR
66    PRTY(I)=0
      DO 661 I=1,N
      DO 662 J=1,100
      JR=JROUTE(I,J)
      IF(JR.EQ.0)GOTO 661
      PRTY(JR)=PRTY(JR)+TIME(I,J)*PVISIT(I,J)
662   CONTINUE
661   CONTINUE
```

Where NRR= the total number of resources, excluding pallets/fixtures and the manual loaders; and PRTY(JR)= the expected cycle time of resource JR. Note that the priority code for the transporters is zero (not used).

USERF(6) is modelled as:

```
66    NEXT=INT(ATRIB(3))+1
      IF(JROUTE(ITYPE,NEXT).NE.0)GOTO 661
      USERF=10000
      RETURN
661   ITYPE=INT(ATRIB(1))
      IOPER=INT(ATRIB(3))
      JR=JROUTE(ITYPE,IOPER)
      USERF=PRTY(JR)
      RETURN
```

In this example, a very high priority (=10000) is defined for the last operation. The SIM-Q resource specification card should specify "1" of "high value first".

Illustration 6.7 Composite Rule

Combining different sequencing rules, each rule applying to a specific subset of resources, is relatively easy to model. For example, the following codes specify that resources 1, 2, and 6 use the Seniority Rule; resources 3,7 use the Shortest Processing Time Rule; and resources 4 and 5 ues the Cycle Time of the Next Station Rule:

```
67    ITYPE=INT(ATRIB(1))
      IOPER=INT(ATRIB(3))
      JR=JROUTE(ITYPE,IOPER)
      GOTO (61,61,62,66,66,61,62), JR
```

A.8.7 Modelling Pallets and Fixtures

USERF(7) returns the pallet type required for the current operation. If the pallet type used in the immediate preceeding operation is not the same as the pallet type required for the current operation, a new pallet will be requested. SIM-Q automatically models this as a manual operation and a manual loader will also be requested. The pallet type for each operation is input through subroutine INTLC, together with the routing information.

Illustration 7.1 Pallets and Fixtures

The following codes may be written:

```
      DO 71 I=1,N
      READ(5,*) MAX
      DO 72 J=1,MAX
      READ(5,*) JROUTE(I,J),TIME(I,J),
                JPALET(I,J),PVISIT(I,J)
72    CONTINUE
```

The following FORTRAN codes model USERF(7) for this example:

```
      ITYPE=INT(ATRIB(1))
      IOPER=INT(ATRIB(3))
      USERF=JPALET(ITYPE,IOPER)
      RETURN
```

A.8.8 Station of Next Visit

USERF(8) returns the part's next workstation. USERF(8) returns a value "0" if all operations have been performed. A value "99" returned by USERF(8) routes the part to a delay node "DLAY". "DLAY" holds the part for one time unit before returning control back to function USERF.

Illustration 8.1 Random Routing

CAN-Q assumes random routing. This can be easily modelled. The transition probabilities, the mean processing times, and the average number of operations are input through subroutine INTLC. A cumulative probability function is also computed. The FORTRAN codes may be written as follows:

```
81    DO 811 I=1,M
      READ(5,*)PROB(I),TIME(I)
811   CONTINUE
      N1=M+MT+1
      N2=M+MT+MD
      DO 812 I=N1,N2
      READ(5,*)PROB(I),TIME(I)
812   CONTINUE
```

```
      DO 813 I=2,M
      PROB(I)=PROB(I-1)+PROB(I)
813   CONTINUE
      PROB(N1)=PROB(M)+PROB(N1)
      N1=N1+1
      DO 814 I=N1,N2
      PROB(I)=PROB(I-1)+PROB(I)
814   CONTINUE
```

Where M= the number of machining and inspection stations; MT= the number of transporter; and MD= the number of load/ unload stations. USERF(8) consists of the following FORTRAN codes:

```
81    RNUM=DRAND(8)
      DO 811 I=1,M
      USERF=I
      IF(RNUM.GE.PROB(I))GOTO 813
811   CONTINUE
      N1=M+MT+1
      N2=M+MT+MD
      DO 812 I=N1,N2
      USERF=I
      IF(RNUM.GE.PROB(I))GOTO 813
812   CONTINUE
813   RETURN
```

The program segment above is not complete. The "last operation" has not been taken into consideration. Let ANOCO= the Average Number of Operations required to Complete One item. Then 1.0/ANOCO is the probability that an operation is the last operation. The following FORTRAN codes must be inserted before the above program segment:

```
      USERF=0
      IF(DRAND(8).LE.1.0/ANOCO) RETURN
```

Illustration 8.2 Deterministic Routing

For deterministic routing, the routing information is input through subroutine INTLC. USERF(8) consists of the following codes:

```
82     ITYPE=INT(ATRIB(1))
       IOPER=INT(ATRIB(3))
       USERF=JROUTE(ITYPE,IOPER)
       RETURN
```

Illustration 8.3 Temporary Storage Stations

Suppose a part is routed to a temporary load/unload station whenever the requested workstation is not available. This can be modelled as follows:

```
83     ITYPE=IFIX(ATRIB(1))
       IOPER=IFIX(ATRIB(3))
       USERF=JROUTE(ITYPE,IOPER)
C IF JROUTE(ITYPE,IOPER)=0, JOB IS DONE
       IF(JROUTE(ITYPE,IOPER).EQ.0) RETURN
C FOR SINGLE RESOURCE OPERATION, IS REQUIRED RESOURCE AVAILABLE?
       IF (NNRSC(  INT(USERF)  )  .GT. 0 ) RETURN
C IS THE PART CURRENTLY IN TEMPORARY STORAGE?
       IF(ATRIB(4).GT.(M+MT+MD)) RETURN
C NO AVAILABLE STATION - MUST FIND A TEMPORARY L/UL STATION
       QUE=99999
       N1=M+MT+MD+1
       N2=M+MT+MD+MU
       DO 831 IR=N1,N2
       IF(NNRSC(IR).EQ.0.AND.NNQ(IR).GE.QUE)GOTO 831
       USERF=IR
       QUE=NNQ(IR)
   831 CONTINUE
       RETURN
```

NNRSC(IR) and NNQ(IR) are SLAM functions. NNQ(IR) returns the number of parts waiting and in-process at resource IR; NNRSC(IR) returns the number of currently available units of resources type IR. M is the total number of machine types and inspection station types , MT is the number of transporter types, MD is the number of loading/unloading station types, and MU is the number of temporary storage station types. The values of M, MT, MD, and MU must be defined through subroutine INTLC.

Illustration 8.4 Multi-Resource Operations

In multi-resource operations, a part may not request for a resource unless all the required resources are available. Otherwise, this may lead to a stalemate situation. To model this, USERF(8) routes the part to a temporary storage station if one or more of the required resources is not available. If the part is currently in a temporary storage station, or if the current operation is the first operation, USERF(8) routes the part to the delay node by returning "99". USERF(8) can be modelled as follows:

```
84    ITYPE=IFIX(ATRIB(1))
      IOPER=IFIX(ATRIB(3))
      USERF=JROUTE(ITYPE,IOPER)
C IF JROUTE(ITYPE,IOPER)=0, JOB IS DONE
      IF(JROUTE(ITYPE,IOPER).EQ.0) RETURN
C IS THIS NOT A MULTI-RESOURCE OPERATION?
      IF(ATRIB(7).EQ.JPALET(ITYPE,IOPER))GOTO 841
C ARE ALL RESOURCES AVAILABLE?
      IF (NNRSC(  M + MT + MD + MU + MP + MM  )*
     1    NNRSC(JPALET(ITYPE,IOPER)+M+MT+MD+MU)*
     2    NNRSC(  INT(USERF)  )  .GT. 0 ) RETURN
C IS THE PART CURRENTLY IN A TEMPORARY STORAGE STATION?
      IF(ATRIB(4).GT.(M+MT+MD))GOTO 843
C IS THIS NOT A LAUNCHING OPERATION?
      IF(ATRIB(3).NE.1) GOTO 842
843   USERF=99
      RETURN
```

```
C FOR SINGLE RESOURCE OPERATION, IS REQUIRED RESOURCE AVAILABLE?
841   IF (NNRSC(  INT(USERF)  )  .GT. 0 ) RETURN
C IS THE PART IN TEMPORARY STORAGE?
842   IF(ATRIB(4).GT.(M+MT+MD)) RETURN
C NO AVAILABLE STATION - MUST FIND A TEMPORARY L/UL STATION
      QUE=99999
      N1=M+MT+MD+1
      N2=M+MT+MD+MU
      DO 844 IR=N1,N2
      IF(NNRSC(IR).EQ.0.AND.NNQ(IR).GE.QUE)GOTO 844
      USERF=IR
      QUE=NNQ(IR)
844   CONTINUE
      RETURN
```

Illustration 8.5 Pooled Machines

Studies have shown that pooling improves machine utilization. To model machine pooling, the following codes may be written for USERF(8):

```
85    ITYPE=INT(ATRIB(1))
      IOPER=INT(ATRIB(3))
      IR=JROUTE(ITYPE,IOPER)
      IF(NNRSC(IR.LE.0) GOTO 851
      USERF=IR
      RETURN
851   DO 852 I=1,M
      IF(I.EQ.IR)GOTO 852
      IF(POOLED(IR).NE.POOLED(I))GOTO 852
      IF(NNRSC(I).LE.0)GOTO 852
      USERF=I
      RETURN
852   CONTINUE
C NOT AVAILABLE.  FIND TEMPORARY STATION
      ... etc ...
```

Where POOLED(IR) is the pool number of resource IR. The pool number of each pooled resource should be input through subroutine INTLC.

A.8.9 Next Location

USERF(9) returns the part's next location (zone number).

Illustration 9.1 Next Location of a Part

The FORTRAN code is as follows:

```
91     ISTA=INT(ATRIB(8))
       USERF=XX(ISTA)
       RETURN
```

A.8.10 Material Handling Selection

USERF(10) returns the resource number of the transporter selected to carry the part. The decision criteria may consist of one or more of the following: (a) availability, (b) proximity, and (c) queue length (the number of parts waiting for a transporter).

Illustration 10.1 Transporter Selection

The following FORTRAN code models the process of transporter selection using these criteria:

```
100    USERF=0
       DO 101 I=1,MT
       IF(NNRSC(I+M).LE.0)GOTO 101
       USERF=I
       GOTO 102
101    CONTINUE
       GOTO 103
C   ...IS THERE A NEARER TRANSPORTER WHICH IS ALSO AVAILABLE ?
102    DO 104 I=1,MT
       IF(I.EQ.INT(USERF))GOTO 104
       IF(NNRSC(I+M).LE.0)GOTO 104
       IF(ABS(XX(INT(USERF)+M)-ATRIB(5)).LT.ABS(XX(I+M)-ATRIB(5)))
      1GOTO 104
       USERF=I
104    CONTINUE
```

```
      RETURN
C   ...NO AVAILBLE CAR, MUST FIND THE CAR WITH THE SHORTEST QUEUE
103   QUE=100000
      DO 105 I=1,MT
      IF(NNQ(I+M).GT.QUE)GOTO 105
      QUE=NNQ(I+M)
      USERF=I
105   CONTINUE
      RETURN
```

A.8.11 USERF Not Used

In many cases, not all of the ten USERF functions are needed for modelling. For example, USERF(6) is not used if the sequencing rule is FIFO or LIFO; USERF(7) is not used if the palletizing operation is not modelled as a multi-resource operation; USERF(5), USERF(9) and USERF(10) are not used if the transporting activity is modelled as in Example 3.1.

Illustration 11.1 Dummy USERF

The user may use the following statement to define these unused USERF functions:

```
      USERF=DUMMY
      RETURN
```

Where DUMMY is set equal 1.

A.9 SUBROUTINE EVENT

There are two event types: EVENT to release a resource, and EVENT for data logging and graphics.

A.9.1 Releasing of a Resource

SIM-Q uses subroutine EVENT to release a resource that is no longer needed (completion of operation). An event is scheduled to free a workstation, a transporter, a temporary storage, a pallet/fixture, or a manual loader.

Unlike the USERF functions, subroutine EVENT is fixed. All users of SIM-Q uses the same basic codes for the subroutine.

Illustration 12.1 Modelling Discrete Events

Five events are modelled. Event 1 releases a pallet/fixture; EVENT 2 releases a workstation, which may be a machining station, an inspection station, or a loading station; EVENT 3 releases a transporter; Event 4 releases a temporary load/unload station; Event 5 releases a manual loader. The following FORTRAN codes model these events:

```
      GOTO(10001,10002,10003,10004,10005),INDEX
10001 IPAL=INT(ATRIB(7))
      CALL FREE(IPAL+M+MT+MD+MU,1)
      RETURN
10002 IR=INT(ATRIB(4))
      CALL FREE(IR,1)
      RETURN
10003 ICAR=INT(ATRIB(10))
      CALL FREE(M+ICAR,1)
      RETURN
10004 ITEMP=INT(ATRIB(4))
```

```
      CALL FREE(ITEMP,1)
      RETURN
10005 CALL FREE(M+MT+MD+MU+MP+MM,1)
      RETURN
```

Where MP is the total number of pallet types; and MM is equal to 1 (there is only one type of manual loaders). The values of M, MT, MD, MU, MP and MM should be input through subroutin INTLC as follows:

```
      READ(5,*) M,MT,MD,MU,MP,MM
```

A.9.2 Graphing Individual Observations

Illustration 5.2 Print and Plotting Individual Observations

The standard SLAM summary report gives only a statistical summary of the simulation flow times. If it is desired to print and plot the individual observations, an event (EVENT 6) may be scheduled each time a part is completed and leaves the system. The network model generated by SIM-Q must be extended. Two lines of codes have to be added to the "COLLECT STATISTICS" segment of the network program. Immediately after the statement "FINI GOON,1;', the following codes are to be inserted:

```
      ACT;
      EVENT,6,1;
```

The plot of the flow times generated from the simulation is useful in determining the truncation point for the simulation. The following FORTRAN codes for EVENT 6 will cause each observed data to be printed and graphed. Notice that the data are differentiated by part type (Types 1, 2, and 3).

```
      DIMENSION LINE(51)
      INTEGER ONE,TWO,THREE,BLANK,ZERO,LINE
             .          .
             .          .
             .          .
10006 FTIM=TNOW-ATRIB(2)
      DATA ICALL /0/
      DATA ONE /'1 '/
      DATA TWO /'2 '/
      DATA THREE/'3 '/
      DATA ZERO /'+ '/
      DATA BLANK /'  '/
      DATA FMIN /100.0/
      DATA FMAX/700.0/
      IF(ICALL.GT.0)GOTO 33
      DO 11 I=1,51
11    LINE(I)=BLANK
      LINE(1)=ZERO
      WRITE(6,22)
22    FORMAT('1',T2,' NO.',2X,'  TIME NOW',
     1'     F  L  O  W     T  I  M  E'/T60,51('-'))
33    ICALL=ICALL+1
      IPLOT=(FTIM-FMIN)/(FMAX-FMIN)*51+1
      IF(IPLOT.LT.1)IPLOT=1
      IF(IPLOT.GT.51)IPLOT=51
      IF(ATRIB(1).EQ.1.)LINE(IPLOT)=ONE
      IF(ATRIB(1).EQ.2.)LINE(IPLOT)=TWO
      IF(ATRIB(1).EQ.3.)LINE(IPLOT)=THREE

IF(ATRIB(1).EQ.1.)WRITE(6,44)ICALL,TNOW,FTIM,(LINE(I),I=1,51)

IF(ATRIB(1).EQ.2.)WRITE(6,55)ICALL,TNOW,FTIM,(LINE(I),I=1,51)

IF(ATRIB(1).EQ.3.)WRITE(6,66)ICALL,TNOW,FTIM,(LINE(I),I=1,51)
44    FORMAT(T2,I4,2X,2F10.2,T60,51A1)
55    FORMAT(T2,I4,2X,F10.2,10X,F10.2,T60,51A1)
66    FORMAT(T2,I4,2X,F10.2,20X,F10.2,T60,51A1)
      LINE(IPLOT)=BLANK
      LINE(1)=ZERO
      RETURN
```

A.10 ILLUSTRATIVE EXAMPLES

The preceding sections provide an application guide to SIM-Q. For the remainder of this user's guide, four examples of SIM-Q generated simulation models are presented. These models differ by the extent of model abstraction (amount of details modelled), and by such system parameters as the number of part types, the number of parts circulating in the system (machine loading), the sequencing rules, the routing process, the number of workstations, the types of transporters, the number of temporary storages, the transporter speed the processing time distributions, and the time of release of a workstation.

The first model demonstrates the minimal effort required by SIM-Q for modelling a flow shop. Since most of the USERF functions are not required, modelling is as simple as running CAN-Q. The second model, based on the flexible manufacturing system at Caterpillar Tractor Company, Peoria, Illinois but assumes random routing and other assumptions of CAN-Q. A more realistic version is presented in the third model. This model assumes exponentially distributed processing times and transporting times, but the part routes are deterministic. The last example is a further embellishment of the two Caterpillar models. The palletizing process is modelled as a multi-resource process, and temporary load/unload stations are incorporated into the model. Transporting times are based on distance travelled. The data used for the Caterpillar models are taken from (MAYER76), (LENZJ77) and (RUNNJ78).

A.11 A SIMPLE CONVEYORIZED FLOW SHOP

Consider a flow shop of four workstations linked by a conveyor system. Suppose the processing time is 80.0 minutes on station number 1, 92.0 minutes on station number 2, 76.0 minutes on station number 3, and 84.0 minutes on station number 4. Also, assume the loading time is 10.0 minutes, equal to the unloading time. The transporting time is assumed to be 1.5 minutes between adjacent stations.

A.11.1 Input Data for Example 1

SIM-Q can model up to six groups of resources - the machining center/inspection stations, the transporters, the loading/unloading stations, the temporary storages, the pallets/fixtures, and the manual loaders. For this example, there are only four machine tools, one loading station, one unloading station and one transporter. The transporter is a conveyor hence the "number of servers" equals the number of parts circulating in the system. Each station has sufficient in-process storage capacity such that no blocking will occur. There is no need to model the temporary storage stations.

Table 30 shows the input data. The model is to be executed for 5,670 minutes (equivalent to twelve 8-hour shifts) and statistics are cleared at the 960th minute (after the second shift). The dispatching rule for all resources is FIFO. The system is loaded with six parts at all time. Except for the conveyor which has to be modelled as a six-servers resource (theoretically, infinite), all resources are modelled as single-server resources.

The option code (field number eight in the second data card) for releasing workstations was specified such that a workstation was released as soon as an operation was completed. The implication is that upon completing processing, each part was immediately picked-up by a transporter, or was ejected to an off-shuttle, thus releasing the workstation. The assumption is not as restrictive as it appears to be. Although there is an elapsed time before a part leaves a workstation, it also takes time for another part to enter the workstation.

TABLE 30

Input Data for Example 1

The LABEL Card

col 1-20	HENRY C. CO
21-40	A SIMPLE FLOW SHOP
43-50	12/25/82

The OPTION Card

```
5780,960,0
```

The PROBLEM SIZE Card

```
4,1,0,0,0,6,1
```

The RESOURCE SPECIFICATION Card

col 1-8	col 15	col 20
MILL1	1	4
DRILL1	1	4
LATHE1	1	4
INSPECT	1	4
CONVEYOR	6	4
LOADING	1	4

The PART NAME Card

col 1-8	PART1

A.11.2 Discrete Interface for Example 1

Subroutine INTLC inputs the routing information. In this example, only one READ statement is required as follows:

```
READ (5,*) (JROUTE(J),TIME(J),J=1,6)
```

Where JROUTE(J) is the resource number (workstation) requested for the Jth operation; and TIME(J) is the corresponding processing time. The arrays JROUTE and TIME should be dimensioned in a user-COMMON statement. Recall that all user-COMMON statement must be present in all user-supplied subroutines.

The only USERF functions needed for this model are USERF(1), USERF(2), USERF(3) and USERF(8). USERF(1) returns the part type of an entering part. Since there is only one part type, the value returned is always 1. USERF(2) returns the processing time and USERF(8) returns the workstation number, for the next operation. USERF(3) returns the transporting time, which is a constant. The program segment for FUNCTION USERF consists of the following:

```
1       USERF=1
        RETURN
2       USERF=TIME(INT(ATRIB(3))
        RETURN
3       USERF=2.5
        RETURN
8       USERF=JROUTE(INT(ATRIB(3))
        RETURN
```

Where ATRIB(3) is defined within the Network model as the operation number of the next visit. In Subroutine EVENT, only two events are needed -- EVENT 2 to release a workstation, and EVENT 3 to release a transporter.

A.11.3 The SIM-Q Generated SLAM Model (Example 1)

The following is the SLAM network model generated for this example.

```
;   ****************************************************
;   **                                                **
;   **              S  I  M   -  Q                    **
;   **                                                **
;   **   SLAM INTEGRATED MODEL FOR QUEUEING NETWORKS  **
;   **                                                **
;   ****************************************************
;
;       D E F I N I T I O N   O F   C O D E S
;
; ATRIB(1) - PART TYPE                ATRIB(2) - TIME ENTERED SYSTEM
; ATRIB(3) - OPERATION NUMBER         ATRIB(4) - PART'S STATION
; ATRIB(5) - PART'S LOCATION          ATRIB(6) - PART'S PRIORITY CODE
; ATRIB(7) - PALLET TYPE USED         ATRIB(8) - PART'S NEXT STATION
; ATRIB(9) - PART'S NEXT LOCATION     ATRIB(10)- TRANSPORTER SELECTED
; XX(I)    - LOCATION OF RESOURCE I
;
GEN,HENRY C. CO         ,A SIMPLE FLOW SHOP  ,12/25/82,1,YES,NO,YES,NO;
;
;        U S E R F   A N D   E V E N T
;
; USERF(1) - RETURNS THE PART TYPE
; USERF(2) - RETURNS THE SERVICE TIME
; USERF(3) - RETURNS THE MOVING TIME
; USERF(4) - DETERMINES IF VISIT IS REQUIRED
; USERF(5) - RETURNS THE PART'S CURRENT LOCATION
; USERF(6) - RETURNS THE PART'S PRIORITY CODE
; USERF(7) - RETURNS THE PALLET TYPE USED
; USERF(8) - RETURNS THE PART'S NEXT STATION
;            RETURNS 0  IF ALL OPERATIONS COMPLETED
;            RETURNS 99 BRANCHES TO "TIME DELAY NODE"
; USERF(9) - RETURNS THE PART'S NEXT LOCATION
; USERF(10)- SELECTS TRANSPORTER
;
; EVENT(1) TO EVENT(5) : FREES (1) A PALLET/FIXTURE, (2) A
;    WORKSTATION, (3) A TRANSPORTER, (4) A TEMPORARY L/UL
;    STATION, OR (5) A MANUAL LABOUR, RESPECTIVELY
;
LIMITS, 5,10,100;
;
;       S E Q U E N C I N G   R U L E S
;
PRIORITY/ 1,FIFO  / 2,FIFO  / 3,FIFO  / 4,FIFO  / 5,FIFO  ;
;
NETWORK;
;
;       D E C L A R E   R E S O U R C E S
;
      RESOURCE/MILL1   ( 1), 1;
      RESOURCE/DRILL1  ( 1), 2;
      RESOURCE/LATHE1  ( 1), 3;
```

```
      RESOURCE/INSPECT ( 1), 4;
      RESOURCE/CONVEYOR( 6), 5;
;
;- - - -    D E F I N E   A T T R I B U T E S     - - - -
;
      CREATE,0,0,,  6;
      ACT;
ENTR  ASSIGN,ATRIB(1)=USERF(1) , ATRIB(2)=TNOW , ATRIB(3)=1,
             ATRIB(7)=     0.0,ATRIB(8)=USERF(8),ATRIB(9)=USERF(9),
             ATRIB(4)=ATRIB(8),ATRIB(5)=ATRIB(9),ATRIB(6)=USERF(6),1;
      ACT;
;
NEXT  GOON,1;
;- - - -     T R A N S I T I O N         - - - -
;
      ACT,,ATRIB(8).LE. 4,MACH;
;
MACH  GOON,1;
;- - - -   M A C H I N E    T O O L S - - - -
;
      ACT,,USERF(4).EQ.0,UPDT;
      ACT,,ATRIB(8).EQ. 1,WS1;
      ACT,,ATRIB(8).EQ. 2,WS2;
      ACT,,ATRIB(8).EQ. 3,WS3;
      ACT,,ATRIB(8).EQ. 4,WS4;
WS1   AWAIT( 1),MILL1   /1,1;
      ACT,,,TRAN;
WS2   AWAIT( 2),DRILL1  /1,1;
      ACT,,,TRAN;
WS3   AWAIT( 3),LATHE1  /1,1;
      ACT,,,TRAN;
WS4   AWAIT( 4),INSPECT /1,1;
      ACT,,,TRAN;
;
PALL  GOON,1;
      ACT,,,TRAN;
;
TRAN  GOON,1;
;- - - -   T R A N S P O R T E R S    - - - -
      ACT,,ATRIB(3).EQ.1,OPER;
      ACT;
      ASSIGN,ATRIB(10)=USERF(10),1;
      ACT,,ATRIB(10).EQ. 1,MH1;
MH1   AWAIT( 5),CONVEYOR/1,1;
      ACT,USERF(3);  ADVANCE MOVING TIME
      ASSIGN,XX( 5)=ATRIB(9),ATRIB(4)=ATRIB(8),
                                ATRIB(5)=ATRIB(9),1;
      ACT;
      EVENT,3,1;  FREE TRANSPORTER
      ACT,,,OPER;
;
```

```
OPER  GOON,1;
;- - - -       P  R  O  C  E  S  S  I  N  G         - - - -
;
      ACT/ 1,USERF(2),ATRIB(8).EQ. 1,RELS;
      ACT/ 2,USERF(2),ATRIB(8).EQ. 2,RELS;
      ACT/ 3,USERF(2),ATRIB(8).EQ. 3,RELS;
      ACT/ 4,USERF(2),ATRIB(8).EQ. 4,RELS;
      ACT,USERF(2);     ADVANCE TIME
RELS  EVENT,2,1;  FREE WORKSTATION
      ACT;
UPDT  ASSIGN,ATRIB(3)=ATRIB(3)+1,ATRIB(6)=USERF(6),
             ATRIB(8)=USERF(8) , ATRIB(9)=USERF(9),1;
      ACT,,ATRIB(8).EQ.0.0,FINI;
      ACT,,,NEXT;
;
FINI  GOON,1;
;- - - -   C  O  L  L  E  C  T   S  T  A  T  I  S  T  I  C  S   - - - -
;
      ACT,,ATRIB(1).EQ. 1,CL1;
CL1   COLCT,INT(2),LOADING FLOW TIME,,1;
      ACT,,,ENTR;
      ENDNETWORK;
INIT,0, 5780.;
MONTR,CLEAR,  960.;
FIN;
```

A.12 THE CAN-Q MODEL OF THE CATERPILLAR SYSTEM

The second example is a model of the Caterpillar DNC system at Peoria, Illinois. The model is based on the assumptions of CAN-Q. CAN-Q was run to compute the transition probabilities, the average processing times, and the average number of operations needed to complete a part. These data are used as input data for running the SLAM network model generated by SIM-Q.

A.12.1 Input Data for Example 2

Table 31 shows the input data for SIM-Q. The resources consist of four milling centers, three drilling centers, two lathes, an inspection station, a two-carts transporter, and seven dedicated loading/unloading stations. Except for the transporter, each resource is modelled as a single-server resource. The transporter consists of two carts and is modelled as a two-server resource. The expected transport time is arbitrarily assumed to be 3.5 minutes. CAN-Q assumes random routing, exponentially distributed service times and unlimited in-process storage capacity.

As shown, the model is to be run for 5,760 minutes and statistics cleared at the 960th minute. There are a total of 18 resources. The dispatching rule for all resources is FIFO. The system is loaded with 16 parts at all times.

TABLE 31

Input Data for Example 2

```
The LABEL Card

col  1-20   HENRY C. CO
    21-40   CATERPILLAR CANQ
    43-50   12/25/82

The OPTION Card

   5780,960,0

The PROBLEM SIZE Card

   10,1,7,0,0,16,3

The RESOURCE SPECIFICATION Card

col 1-8  col 15  col 20     col 1-8  col 15  col 20

MILL1       1       4       INSPECT     1       4
MILL2       1       4       CARTS       2       4
MILL3       1       4       LOAD1       1       4
MILL4       1       4       LOAD2       1       4
DRIL1       1       4       LOAD3       1       4
DRLL2       1       4       LOAD4       1       4
DRLL3       1       4       LOAD5       1       4
LATH1       1       4       LOAD6       1       4
LATH2       1       4       LOAD7       1       4

The PART NAME Card

col 1-8     PART1
            PART2
            PART3
```

Three part types of equal proportion are processed in the system.

A.12.2 Discrete Interface for Example 2

Seven USERF functions are needed in this model, all of which have been described in the preceding sections. Table 32 lists the corresponding Illustration Numbers where these USERF functions are presented.

Since the palletizing operation is not modelled as a multi-resource operation, EVENT 1 and EVENT 5 are not used.

TABLE 32

USERF Functions for Example 2

USERF	Illustration
1	1.1
2	2.1
3	3.1
4	Dummy
5	Dummy
6	Dummy
7	Dummy
8	8.1
9	9.1
10	Dummy

A.12.3 The SIM-Q Generated SLAM Model (Example 2)

The SLAM network model generated by SIM-Q is as follows:

```
;   ****************************************************
;   **                                                **
;   **                S  I  M    -  Q                 **
;   **                                                **
;   **   SLAM INTEGRATED MODEL FOR QUEUEING NETWORKS  **
;   **                                                **
;   ****************************************************
;
;       D E F I N I T I O N   O F   C O D E S
;
; ATRIB(1) - PART TYPE                ATRIB(2) - TIME ENTERED SYSTEM
; ATRIB(3) - OPERATION NUMBER         ATRIB(4) - PART'S STATION
; ATRIB(5) - PART'S LOCATION          ATRIB(6) - PART'S PRIORITY CODE
; ATRIB(7) - PALLET TYPE USED         ATRIB(8) - PART'S NEXT STATION
; ATRIB(9) - PART'S NEXT LOCATION     ATRIB(10)- TRANSPORTER SELECTED
; XX(I)    - LOCATION OF RESOURCE I
;
GEN,HENRY C. CO         ,CATERPILLAR CANQ    ,12/25/82,1,YES,NO,YES,NO;
;
;        U S E R F   A N D   E V E N T
;
; USERF(1) - RETURNS THE PART TYPE
; USERF(2) - RETURNS THE SERVICE TIME
; USERF(3) - RETURNS THE MOVING TIME
; USERF(4) - DETERMINES IF VISIT IS REQUIRED
; USERF(5) - RETURNS THE PART'S CURRENT LOCATION
; USERF(6) - RETURNS THE PART'S PRIORITY CODE
; USERF(7) - RETURNS THE PALLET TYPE USED
; USERF(8) - RETURNS THE PART'S NEXT STATION
;            RETURNS 0  IF ALL OPERATIONS COMPLETED
;            RETURNS 99 BRANCHES TO "TIME DELAY NODE"
; USERF(9) - RETURNS THE PART'S NEXT LOCATION
; USERF(10)- SELECTS TRANSPORTER
;
; EVENT(1) TO EVENT(5) : FREES (1) A PALLET/FIXTURE, (2) A
;    WORKSTATION, (3) A TRANSPORTER, (4) A TEMPORARY L/UL
;    STATION, OR (5) A MANUAL LABOUR, RESPECTIVELY
;
LIMITS,18,10,100;
;
;       S E Q U E N C I N G   R U L E S
;
PRIORITY/ 1,FIFO  / 2,FIFO  / 3,FIFO  / 4,FIFO  / 5,FIFO  ;
PRIORITY/ 6,FIFO  / 7,FIFO  / 8,FIFO  / 9,FIFO  /10,FIFO  ;
PRIORITY/11,FIFO  /12,FIFO  /13,FIFO  /14,FIFO  /15,FIFO  ;
PRIORITY/16,FIFO  /17,FIFO  /18,FIFO  ;
;
NETWORK;
;
;       D E C L A R E   R E S O U R C E S
;
```

```
      RESOURCE/MILL1    ( 1), 1;
      RESOURCE/MILL2    ( 1), 2;
      RESOURCE/MILL3    ( 1), 3;
      RESOURCE/MILL4    ( 1), 4;
      RESOURCE/DRIL1    ( 1), 5;
      RESOURCE/DRLL2    ( 1), 6;
      RESOURCE/DRLL3    ( 1), 7;
      RESOURCE/LATH1    ( 1), 8;
      RESOURCE/LATH2    ( 1), 9;
      RESOURCE/INSPECT  ( 1),10;
      RESOURCE/CARTS    ( 2),11;
      RESOURCE/LOAD1    ( 1),12;
      RESOURCE/LOAD2    ( 1),13;
      RESOURCE/LOAD3    ( 1),14;
      RESOURCE/LOAD4    ( 1),15;
      RESOURCE/LOAD5    ( 1),16;
      RESOURCE/LOAD6    ( 1),17;
      RESOURCE/LOAD7    ( 1),18;
;
;- - - -    D E F I N E   A T T R I B U T E S     - - - -
;
      CREATE,0,0,, 16;
      ACT;
ENTR  ASSIGN,ATRIB(1)=USERF(1) , ATRIB(2)=TNOW , ATRIB(3)=1,
             ATRIB(7)=     0.0,ATRIB(8)=USERF(8),ATRIB(9)=USERF(9),
             ATRIB(4)=ATRIB(8),ATRIB(5)=ATRIB(9),ATRIB(6)=USERF(6),1;
      ACT;
;
NEXT  GOON,1;
;- - - -      T   R   A   N   S   I   T   I   O   N        - - - -
;
      ACT,,ATRIB(8).LE.10,MACH;
      ACT,,,LOAD;
;
MACH  GOON,1;
;- - - -   M   A   C   H   I   N   E    T   O   O   L   S - - - -
;
      ACT,,USERF(4).EQ.0,UPDT;
      ACT,,ATRIB(8).EQ. 1,WS1;
      ACT,,ATRIB(8).EQ. 2,WS2;
      ACT,,ATRIB(8).EQ. 3,WS3;
      ACT,,ATRIB(8).EQ. 4,WS4;
      ACT,,ATRIB(8).EQ. 5,WS5;
      ACT,,ATRIB(8).EQ. 6,WS6;
      ACT,,ATRIB(8).EQ. 7,WS7;
      ACT,,ATRIB(8).EQ. 8,WS8;
      ACT,,ATRIB(8).EQ. 9,WS9;
      ACT,,ATRIB(8).EQ.10,WS10;
WS1   AWAIT( 1),MILL1   /1,1;
      ACT,,,TRAN;
WS2   AWAIT( 2),MILL2   /1,1;
```

```
      ACT,,,TRAN;
WS3   AWAIT( 3),MILL3   /1,1;
      ACT,,,TRAN;
WS4   AWAIT( 4),MILL4   /1,1;
      ACT,,,TRAN;
WS5   AWAIT( 5),DRIL1   /1,1;
      ACT,,,TRAN;
WS6   AWAIT( 6),DRLL2   /1,1;
      ACT,,,TRAN;
WS7   AWAIT( 7),DRLL3   /1,1;
      ACT,,,TRAN;
WS8   AWAIT( 8),LATH1   /1,1;
      ACT,,,TRAN;
WS9   AWAIT( 9),LATH2   /1,1;
      ACT,,,TRAN;
WS10  AWAIT(10),INSPECT /1,1;
      ACT,,,TRAN;
;
LOAD  GOON,1;
;- - - -     L O A D I N G   S T A T I O N S     - - - -
;
      ACT,,USERF(4).EQ.0,UPDT;
      ACT,,ATRIB(8).EQ.12,WS12;
      ACT,,ATRIB(8).EQ.13,WS13;
      ACT,,ATRIB(8).EQ.14,WS14;
      ACT,,ATRIB(8).EQ.15,WS15;
      ACT,,ATRIB(8).EQ.16,WS16;
      ACT,,ATRIB(8).EQ.17,WS17;
      ACT,,ATRIB(8).EQ.18,WS18;
WS12  AWAIT(12),LOAD1   /1,1;
      ACT,,,PALL;
WS13  AWAIT(13),LOAD2   /1,1;
      ACT,,,PALL;
WS14  AWAIT(14),LOAD3   /1,1;
      ACT,,,PALL;
WS15  AWAIT(15),LOAD4   /1,1;
      ACT,,,PALL;
WS16  AWAIT(16),LOAD5   /1,1;
      ACT,,,PALL;
WS17  AWAIT(17),LOAD6   /1,1;
      ACT,,,PALL;
WS18  AWAIT(18),LOAD7   /1,1;
      ACT,,,PALL;
;
PALL  GOON,1;
      ACT,,,TRAN;
;
TRAN  GOON,1;
;- - - -   T R A N S P O R T E R S     - - - -
      ACT,,ATRIB(3).EQ.1,OPER;
      ACT;
```

```
      ASSIGN,ATRIB(10)=USERF(10),1;
      ACT,,ATRIB(10).EQ. 1,MH1;
MH1   AWAIT(11),CARTS   /1,1;
      ACT,USERF(3);  ADVANCE MOVING TIME
      ASSIGN,XX(11)=ATRIB(9),ATRIB(4)=ATRIB(8),
                                 ATRIB(5)=ATRIB(9),1;
      ACT;
      EVENT,3,1;  FREE TRANSPORTER
      ACT,,,OPER;
;
OPER  GOON,1;
;- - - -       P  R  O  C  E  S  S  I  N  G        - - - -
;
      ACT/ 1,USERF(2),ATRIB(8).EQ. 1,RELS;
      ACT/ 2,USERF(2),ATRIB(8).EQ. 2,RELS;
      ACT/ 3,USERF(2),ATRIB(8).EQ. 3,RELS;
      ACT/ 4,USERF(2),ATRIB(8).EQ. 4,RELS;
      ACT/ 5,USERF(2),ATRIB(8).EQ. 5,RELS;
      ACT/ 6,USERF(2),ATRIB(8).EQ. 6,RELS;
      ACT/ 7,USERF(2),ATRIB(8).EQ. 7,RELS;
      ACT/ 8,USERF(2),ATRIB(8).EQ. 8,RELS;
      ACT/ 9,USERF(2),ATRIB(8).EQ. 9,RELS;
      ACT/10,USERF(2),ATRIB(8).EQ.10,RELS;
      ACT,USERF(2);     ADVANCE TIME
RELS  EVENT,2,1;  FREE WORKSTATION
      ACT;
UPDT  ASSIGN,ATRIB(3)=ATRIB(3)+1,ATRIB(6)=USERF(6),
             ATRIB(8)=USERF(8) , ATRIB(9)=USERF(9),1;
      ACT,,ATRIB(8).EQ.0.0,FINI;
      ACT,,,NEXT;
;
FINI  GOON,1;
;- - - -   C O L L E C T   S T A T I S T I C S   - - - -
;
      ACT,,ATRIB(1).EQ. 1,CL1;
      ACT,,ATRIB(1).EQ. 2,CL2;
      ACT,,ATRIB(1).EQ. 3,CL3;
CL1   COLCT,INT(2),PART1   FLOW TIME,,1;
      ACT,,,ENTR;
CL2   COLCT,INT(2),PART2   FLOW TIME,,1;
      ACT,,,ENTR;
CL3   COLCT,INT(2),PART3   FLOW TIME,,1;
      ACT,,,ENTR;
      ENDNETWORK;
INIT,0, 5780.;
MONTR,CLEAR,  960.;
FIN;
```

A.13 DETERMINISTIC ROUTING MODEL

The third example is a refinement of the preceeding model. The routing is deterministic. The input data required for SIM-Q is the same as those used in the second model.

Table 33 lists the appropriate Illustration Numbers where the USERF functions can be found.

A.14 THE DETAILED MODEL OF THE CATERPILLAR FMS SYSTEM

In this model, the transporting times are computed based on the distance travelled. The loading/unloading operations are modelled as multi-resource operations.

The input data for SIM-Q in this example is basically the same as that of the preceding example except for the nine temporary storage stations incorporated into the model. Also, instead of a two-server transporter, the two carts are modelled as separate resources.

All ten USERF functions are needed in this model. The Illustration Numbers where each of these functions have been presented are shown in Table 34.

TABLE 33

USERF Functions for Example 3

USERF	Example
1	1.1
2	2.1
3	3.1
4	4.1
5	Dummy
6	Dummy
7	Dummy
8	8.2
9	9.1
10	Dummy

TABLE 34

USERF Functions for Example 4

USERF	Example
1	1.1
2	2.2
3	3.2
4	4.1
5	5.1
6	6.6
7	7.1
8	8.2
9	9.1
10	10.1

Appendix B

PROGRAM LISTING FOR THE SYSTEM SYNTHESIS MODEL

```
      SUBROUTINE ANALYZE(I1,I2,I3.M.MPART,PCOST)
      COMMON NAME(50).NS(50),Q(50),PT(50),G(250),PIDEN(50),NOPS(50),
     1FRAC(50),APTS(50),ANOPS(50),V(50),R(50),COST(51),NST(50,20),
     2PTO(50,20),FRE(50,20),F(250)
      CHARACTER*8 NAME,PIDEN
      WRITE(6,1)
1     FORMAT(20X,45H* * * * * * * * * * * * * * * * * * * * * * *)
      WRITE(6,2)
2     FORMAT(20X,45H*                                           *)
      WRITE(6,3)
3     FORMAT(20X,45H*             PART SELECTION AND            *
     1/      20X,45H*        FMS CONFIGURATION ANALYSIS         *)
      WRITE(6,4)
4     FORMAT(20X,45H*                     BY                    *)
      WRITE(6,5)
5     FORMAT(20X,45H*                  C A N Q                  *)
      WRITE(6,2)
      WRITE(6,1)
      WRITE(6,6)
6     FORMAT(//)
      IF(I1.EQ.0)GOTO 30
C
C  SET-UP THE SYSTEM
C
      PCOST=0.
      WRITE(6,*) 'ENTER THE NUMBER OF MACHINE TYPES.'
      READ(6,*) M
      WRITE(6,*) 'ENTER THE NUMBER OF PRODUCT TYPES.'
      READ(6,*) NPART
      MM=M+1
      WRITE(6,6)
      WRITE(6,*) 'ENTER THE NAME OF THE MATERIAL HANDLING SYSTEM.'
      READ(6,7) NAME(MM)
7     FORMAT(A8)
      WRITE(6,*) 'ENTER THE NUMBER OF CARRIERS AVAILABLE.'
      READ(6,*) NS(MM)
      WRITE(6,*) 'ENTER THE INVESTMENT COST FOR THE MATL HANDL SYST.'
      READ(6,*) COST(MM)
      PCOST=PCOST+COST(MM)
      WRITE(6,*) 'ENTER THE AVERAGE TRANSPORTING TIME.'
      READ(6,*) PT(MM)
      WRITE(6,6)
      DO 10 I=1,M
      WRITE(6,6)
      WRITE(6,*) 'ENTER THE NAME OF MACHINE TYPE ',I
      READ(6,7) NAME(I)
      WRITE(6,*) 'ENTER THE NUMBER OF UNITS OF THIS MACHINE TYPE.'
      READ(6,*) NS(I)
      WRITE(6,*) 'ENTER THE INVESTMENT COST FOR ALL UNITS OF THIS TYPE.'
      READ(6,*) COST(I)
      PCOST=PCOST+COST(I)
```

```
10    CONTINUE
      WRITE(6,6)
      WRITE(6,*) 'ENTER COST OF COMPUTER, AUX EQUIPT, ETC.'
      READ(6,*) COST(MM+1)
      PCOST=PCOST+COST(MM+1)
      WRITE(6,*) 'ENTER THE NO. OF WORKING HOURS AVAILABLE PER MONTH.'
      READ(6,*) HOURS
      DO 20 K=1,NPART
      ANOPS(K)=0
      WRITE(6,6)
      WRITE(6,*) 'ENTER THE NAME OF PRODUCT TYPE ',K
      READ(6,7) PIDEN(K)
      WRITE(6,*) 'ENTER THE NO. OF OPERATIONS REQUIRED.'
      READ(6,*) NOPS(K)
      NEND=NOPS(K)
        DO 25 NO=1,NEND
        WRITE(6,*) 'ENTER MACHINE TYPE FOR OPERATION NO. ',NO
        READ(6,*) NST(K,NO)
        WRITE(6,*) 'ENTER CORRESPONDING PROCESSING TIME.'
        READ(6,*) PTO(K,NO)
        WRITE(6,*) 'ENTER THE FREQUENCY OF VISIT (0.0 TO 1.0).'
        READ(6,*) FRE(K,NO)
        ANOPS(K)=ANOPS(K)+FRE(K,NO)
25      CONTINUE
20    CONTINUE
30    IF(I2.EQ.0)GOTO 50
C
C  PRODUCTION REQUIREMENT
C
31    WRITE(6,6)
      WRITE(6,*) 'ENTER THE PRODUCTION REQUIREMENT (UNITS/MONTH)'
      TOTAL=0.
      DO 35 K=1,NPART
      WRITE(6,*) 'PRODUCT TYPE ',K
      READ(6,*) FRAC(K)
      TOTAL=TOTAL+FRAC(K)
35    CONTINUE
      DO 40 K=1,NPART
40    FRAC(K)=FRAC(K)/TOTAL
50    IF(I3.EQ.0)GOTO 260
C
C  PRE-PROCESS ROUTING DATA FOR USE IN QUEUEING-NETWORK ANALYSIS
C
51    WRITE(6,6)
      DO 160 I=1,M
      Q(I)=0.
      V(I)=0.
      PT(I)=0.
      APTS(I)=0.
160   CONTINUE
      ANOP=0.
```

```
      DO 255 K=1,NPART
      IF(FRAC(K).LE.0)GOTO 255
      NEND=NOPS(K)
        DO 200 NO=1,NEND
        NSTA=NST(K,NO)
        V(NSTA)=V(NSTA)+FRE(K,NO)
        APTS(NSTA)=APTS(NSTA)+PTO(K,NO)*FRE(K,NO)
200     CONTINUE
      ANOP=ANOP+FRAC(K)*ANOPS(K)
        DO 235 J=1,M
        PT(J)=PT(J)+APTS(J)*FRAC(K)
        Q(J) =Q(J) +V(J)   *FRAC(K)
        APTS(J)=0.
        V(J)=0.
235     CONTINUE
255   CONTINUE
260   CONTINUE
C
C  COMPUTE RELATIVE WORKLOAD AND IDENTIFY BOTTLENECK
C
      Q(MM)=1./ANOP
      R(MM)=PT(MM)
      RMAX=R(MM)/FLOAT(NS(MM))
      IMAX=MM
      DO 270 I=1,M
      IF(Q(I).GT.0.)PT(I)=PT(I)/Q(I)
      Q(I)=Q(I)/ANOP
      R(I)=Q(I)*PT(I)
      TEST=R(I)/FLOAT(NS(I))
      IF(TEST.LE.RMAX)GOTO 265
      IMAX=I
      RMAX=TEST
265   CONTINUE
270   CONTINUE
C
C  SUMMARY OF THE SYSTEM CONFIGURATION
C
      WRITE(6,6)
      WRITE(6,285)
285   FORMAT(17X,9HNUMBER OF,5X,5HVISIT,9X,7HAVERAGE,8X,8HRELATIVE,5X,
     18HWORKLOAD)
      WRITE(6,290)
290   FORMAT(5X,7HSTATION,6X,7HSERVERS,4X,9HFREQUENCY,3X,15HPROCESSING
     1 TIME,4X,8HWORKLOAD,4X,10HPER SERVER/)
      DO 305 I=1,MM
      WPS=R(I)/FLOAT(NS(I))
      WRITE(6,300) I,NAME(I),NS(I),Q(I),PT(I),R(I),WPS
300   FORMAT(1X,I2,1X,A8,7X,I3,4X,F10.5,7X,F10.5,6X,F10.5,2X,F10.5)
305   CONTINUE
      N=-1
      DO 306 I=1,M
```

```
306   N=N+NS(I)
      IF(N.LE.1)N=2
      WRITE(6,6)
      WRITE(6,*) 'NUMBER OF ITEMS IN SYSTEM =',N
      WRITE(6,*) 'MEAN NO. OF OPERATIONS TO COMPLETE AN ITEM= ',ANOP
C
C  COMPUTE G MATRIX
C
      NN=N+5
      DO 325 J=1,NN
      F(J)=0.
      G(J)=0
325   CONTINUE
      DO 360 I=1,MM
      IF(R(I).LE.0) GOTO 355
      TEMP=1
      IF(NS(I).GT.1)GOTO 335
C
C  SINGLE SERVER CASE
C
        DO 330 J=1,NN
        TEMP=R(I)*TEMP+G(J)
        G(J)=TEMP
330     CONTINUE
      GOTO 350
C
C  MULTIPLE SERVER CASE
C
335   CONTINUE
        DO 345 J=1,NN
        JJ=NN-J+1
          DO 340 K=1,JJ
          A=FLOAT(NS(I))
          CHK=FLOAT(JJ-K+1)
          IF(CHK.LT.A)A=CHK
          TEMP=(R(I)*TEMP)/A+G(K)
340       CONTINUE
        G(JJ)=TEMP
        TEMP=1
345     CONTINUE
350   CONTINUE
355   CONTINUE
360   CONTINUE
C
C  COMPUTE AND PRINT SYSTEM PERFORMANCE
C
      P=60.*Q(MM)*G(N-1)/G(N)
      PM=P*HOURS
      WRITE(6,6)
      WRITE(6,*) 'PRODUCTION RATE = ',P,'/HOUR OR ',PM,'/MONTH.'
      WRITE(6,*) 'REQUIRED PRODUCTION =',TOTAL,'/MONTH.'
```

```
      WRITE(6,*) 'THE BOTTLENECK MACHINE IS ',NAME(IMAX)
      T=FLOAT(N)*G(N)/(Q(MM)*G(N-1))
      WRITE(6,*) 'AVERAGE FLOW TIME = ',T,' [illegible]
      [illegible] *100
      WRITE(6,*) 'SYSTEM UTILIZATION = ',CAPUTL,' PERCENT'
      WRITE(6,*) 'SYSTEM COST IS = ',PCOST
      WRITE(6,6)
C
C
C
      WRITE(6,*) 'ENTER 1 TO REVISE PART SELECTION.'
      WRITE(6,*) 'ENTER 2 TO ADD MACHINE(S).'
      WRITE(6,*) 'OTHERWISE, ENTER 0'
      READ (6,*) MODE
      IF(MODE.EQ.0)RETURN
      IF(MODE.EQ.1)GOTO 31
      WRITE(6,6)
      WRITE(6,*) 'ENTER NO. OF ADDITIONAL UNITS OF MACHINE ',
     1NAME(IMAX),'.'
      READ(6,*) NADD
      NS(IMAX)=NS(IMAX)+NADD
      WRITE(6,*) 'ENTER ADDITIONAL COST.'
      READ(6,*) ADCOST
      PCOST=PCOST+ADCOST
      GOTO 51
      END
```

Appendix C

PROGRAM LISTING FOR THE FMS JUSTIFICATION MODEL

```
        DIMENSION NS(50),NOPS(50),NST(50,20),PTO(50,20),REQMT(50),FRAC(50),
     1  NCOUNT(20),NAME(50),PIDEN(50),ADINVES1(20,20),TOCAP(20),
     1  SALVAGE(20),NSHIFT(20)
        DIMENSION NCNF(20),AVCAP(20,50),AVBOT(20,20),SUBCOST(50),COST(20,20),
     1  PV(20),CHANCOST(20),ENCOST(20)
        DIMENSION ADINVEST(20,20),NSO(20,20,20),SHIFT(20),CONCODE(20)
        COMMON/FASE/NAME,PIDEN
        COMMON/ONE/NS,NOPS,NST,PTO,REQMT,HOUR,M,NPART,PY,FRAC,IMAX,MM,
     1  AVETRANS
        COMMON/TWO/AA(50),BB(50),CC(50,20,20),DD(50,20),EE(50,20),
     1  AVDEM(50),NOPAV1(50),NOPAV2(50),NST1(50,20),NOPS1(50),NOCH(50),
     1   ISEED
        COMMON/THREE/NCNF,AVCAP,SUBCOST,COST,PV,ADINVEST,SHIFT,KYR,RETRNRT,
     1  MYR,AVBOT,NSO,CONCODE,ENCOST,NCOUNT,NSH,ADINVES1,TOCAP,NOCHAV,
     1   SALVAGE
        CHARACTER*8 NAME,PIDEN,CHAR
        ISEED=35678993
        MSIM=1
      WRITE(6,*)'            *************************************'
      WRITE(6,*)
      WRITE(6,*)'                 A MANAGEMENT GAME APPROACH          '
      WRITE(6,*)'                    TO FMS JUSTIFICATION             '
      WRITE(6,*)
      WRITE(6,*)'                         MARCH 9, 1984               '
      WRITE(6,*)'            *************************************'
      WRITE(6,*)
      WRITE(6,*)
C
C  *********************************************************************
C
C                                 ABSTRACT
C             THE DYNAMIC DECISION-MAKING METHOD DEVELOPED RECENTLY
C      IN << DECISION ANALYSIS >> BY R. HOWARD, IS CARRIED OUT HERE
C      INTERACTIVELY BY USING  SIMULATION TECHNIQUES AND THE THEORY
C      OF QUEUEING NETWORK.
C             THE PROCEDURE CONSISTS OF TWO PHASES AS FOLLOWS:
C        PHASE 1 : PART SELECTION AND CONFIGURATION GENERATION.
C        PHASE 2 : DECISION MAKING BY MANAGEMENT GAMES. THIS IS THE
C                  FORTRAN CODE FOR PHASE 2.
C
C  **********************************************************************
C
C  ***********************************************************************
C
C   VARIABLE DEFINITIONS:
C  NS (I)   = NUMBER OF UNITS OF MACHINES FOR TYPE I MACHINE.
C          NOTE: THE INPUT OF NS(I) IS THE CURRENT NUMBER OF MACHINES.
C          THIS SUBROUTINE RETURNS THE REQUIRED MACHINES IF INDEX = 1.
C  NOPS(I) = NUMBER OF OPERATIONS FOR PRODUCT FAMILY I
C  NST(I,J)= THE MACHINE TYPE VISITED BY PRODUCT FAMILY I IN ITS JTH
C          OPERATION.
```

```
C  PTO(I,J)= THE PROCESSING TYPE FOR PRODUCT I FAMILY IN ITS JTH OPERA
C           TION.
C  REQMT(I)= THE YEARLY PRODUCTION REQUIREMENT FOR PRODUCT FAMILY I.
C  HOUR     = NUMBER OF WORKING HOURS IN ONE YEAR.
C  M        = NUMBER OF MACHINE TYPES.
C  NPART    = NUMBER OF PRODUCT FAMILIES.
C  TOTAL    = THE ANNUAL PRODUCTION REQUIREMENT
C  PY       = THE ANNUAL PRODUCTION CAPACITY (UNITS/YEAR).
C  IMAX     = THE BOTTLENECK MACHINE TYPE
C
C  *********************************************************************
C
C
C  *********************************************************************
C    INITIAL CONFIGURATION GENERATION:
C       PHASE 1 WILL GENERATE A SET OF POSSIBLE CONFIGURATION CANDIDATES
C     AND CAN-Q WILL COMPUTE SOME MAJOR PERFORMANCE INDICES FOR EACH OF
C     THOSE CANDIATES. BASED ON THESE INFORMATION, HERE BY MAN-MACHINE
C     CONVERSATION A SET OF NON-DOMINATED CONFIGURATIONS ARE SELECTED AS
C     INITIAL CONFIGURATIONS.
C  **********************************************************************
C
        WRITE(6,*) 'PLEASE ENTER THE LIFETIME OF THE SYSTEM.'
        READ (6,*) MYR
        WRITE(6,*) 'PLEASE ENTER THE NUMBER OF WORKING HOURS IN A YEAR.'
        READ(6,*) HOUR
        WRITE(6,*)'PLEASE ENTER THE RATE OF RETURN.'
        READ(6,*) RETRNRT
        WRITE(6,*)'PLEASE ENTER THE APPLICABLE TAX RATE.'
        READ(6,*) TAXR
        TAXRA=TAXR
        WRITE(6,*)'PLEASE ENTER THE NUMBER OF MACHINE TYPES.'
        READ(6,*) M
        MM=M+1
        DO I=1,M
          WRITE(6,6) I
6       FORMAT(' PLEASE ENTER THE NAME OF MACHINE TYPE',I3)
          READ(6,7) NAME(I)
7       FORMAT(A8)
        END DO
          WRITE (6,*)'ENTER THE NAME OF THE MATERIAL HANDLING SYSTEM'
          READ(6,7) NAME(MM)
          WRITE(6,*)'ENTER THE AVERAGE TRANSPORT TIME.'
          READ(6,*)AVETRANS
        WRITE(6,*)'HOW MANY TYPES OF PRODUCTS ARE IN THE SYSTEM?'
        READ(6,*) NPART
        WRITE(6,*)'PLEASE ENTER THE NAME OF EACH PRODUCT TYPE.'
        WRITE(6,*)'ARRANGE THE PRODUCT TYPE ACCORDING TO '
      1,'DECREASING PRIORITY'
        DO I=1,NPART
          WRITE(6,8) I
```

```
8       FORMAT(' PLEASE ENTER THE NAME OF PRODUCT',I3,'.')
          READ(6,7) PIDEN(I)
          WRITE(6,*)' ENTER THE AVERAGE ANNUAL DEMAND FOR ',PIDEN(I)
          READ(6,*) AVDEM(I)
          REQMT(I)= AVDEM(I)
          AA(I)=AVDEM(I)*0.15
          WRITE(6,*)' ENTER THE PROBABILITY OF ENGINEERING CHANGE
     1  FOR ',PIDEN(I)
          READ(6,*) BB(I)
C ***********************************************************************
C      AVNOP1 AND AVNOP2 ARE USED FOR RANDOM ROUTING GENERATING IN RANDOM
C              NUMBER GENERATING FUCTIONS OF THIS PROGRAM.
C************************************************************************
          WRITE(6,*)'ENTER THE MINIMUM NUMBER OF OPERATIONS FOR PRODUCT ',
     1 PIDEN(I)
          READ(6,*) NOPAV1(I)
          WRITE(6,*)'ENTER THE MAXIMUM NUMBER OF OPERATIONS FOR PRODUCT ',
     1 PIDEN(I)
          READ(6,*) NOPAV2(I)
          NOPS1(I)=(FLOAT(NOPAV2(I)-NOPAV1(I))/2+0.5)
          DO IO=1,NOPAV2(I)
            WRITE(6,*)'MACHINE FOR OPERATION ',IO,'?'
            READ(6,*) NST1(I,IO)
            NST(I,IO)=NST1(I,IO)
            WRITE(6,*)'PROCESSING TIME FOR OPERATION ',IO,'?'
            READ(6,*) PTO(I,IO)
            DD(I,IO)= PTO(I,IO)
            EE(I,IO)= PTO(I,IO)*.10
          END DO
        END DO
        KYR=1
C
C  ***********************************************************************
C   THE FOLLOWING IS BASED ON THE RESULTS FROM PHASE 1.
C  ***********************************************************************
C
        WRITE(6,*) 'HOW MANY ALTERNATIVE SYSTEM CONFIGURATIONS ',
     1  '(INCLUDING CONVENTIONAL SYSTEM). DO YOU WISH TO ',
     1  'INCLUDE AS INITIAL CANDIDATES?'
        READ (6,*) NCNF(KYR)
        DO I=1,NCNF(KYR)
          CONCODE(I)=I
          ADINVEST(I,KYR)=0.
          WRITE(6,9) I
9       FORMAT(' FOR CONFIGURATION #',I3,':')
          DO J=1,M
            WRITE(6,*) '  PLEASE ENTER THE NUMBER OF UNITS OF ',
     1  NAME(J)
            READ(6,*) NSO(I,J,KYR)
            WRITE(6,*)'  ENTER THE INITIAL INVESTMENT FOR ',NAME(J)
            READ(6,*) RINITIAL
```

```
            ADINVEST(I,KYR) = ADINVEST(I,KYR) + RINITIAL
          END DO
          WRITE(6,*)'  PLEASE ENTER THE NUMBER OF CARRIERS.'
          READ(6,*) NSO(I MM,KYR)
          WRITE(6,*)'  ENTER THE INITIAL INVESTMENT FOR ',NAME(MM)
          READ(6,*) RINITIAL
          ADINVEST(I,KYR) = ADINVEST(I,KYR) + RINITIAL
          COST(I,MYR+1)=ADINVEST(I,KYR)
          WRITE(6,*)'  ENTER THE ENGINEERING CHANGE COST FOR THE
     1  CONFIGURATION.'
          READ(6,*) CHANCOST(I)
        END DO
        SMALL=0.5/(M-1)
        DO I=1,NPART
          DO J=1,M
            TEST=0.
            DO K=1,M
              IF (J.EQ.K) THEN
                TEST=TEST+0.5
              ELSE
                TEST=TEST+SMALL
              END IF
              CC(I,J,K)=TEST
            END DO
          END DO
        END DO
C
C  *********************************************************************
C     SIMULATE THE NEW CONFIGURATIONS FOR THE NEXT YEAR. ALL SYSTEM
C  PARAMETERS ARE SIMULATED BY RANDOM NUMBER GENERATING FUNCTIONS AND
C  ARE SENT TO CAN-Q ELEMENT TO GET A SET OF SAMPLE SYSTEM PERFORMANCES.
C     FINAL SYSTEM PERFORMANCES ARE EVALUATED BASED ON THE MEAN VALUE
C  OF SIMULATION SAMPLES.
C  *********************************************************************
C
900     NSIM=1
1000    DO I=1,NPART
          REQMT(I)=RANDEM(I)
          CALL ROUTING(I)
        END DO
        DO I=1,NCNF(KYR)
          DO J=1,MM
            NS(J)=NSO(I,J,KYR)
          END DO
          CALL CANQ(0)
          DO J=1,NPART
            AVCAP(I,J)=AVCAP(I,J)+PY*FRAC(J)
          END DO
          DO J=1,MM
          IF (IMAX.EQ.J) AVBOT(I,J)=AVBOT(I,J)+1
          END DO
```

```
      END DO
      NSIM=NSIM+1
      IF (NSIM.LT.MSIM)  GO TO 1000
C
C  OUTPUT FOR THE END OF THE YEAR.
C
2000  DO I=1,NCNF(KYR)
        DO J=1,NPART
          NOCHAV=NOCHAV+NOCH(J)
          AVCAP(I,J)=AVCAP(I,J)/MSIM
        END DO
        NOCHAV=NOCHAV/MSIM
        ENCOST(I)=NOCHAV*CHANCOST(I)
        DO J=1,MM
          AVBOT(I,J)=AVBOT(I,J)/MSIM
        END DO
      END DO
      CALL COSTEVAL(0)
      IF (KYR.EQ.MYR) GO TO 5000
      KYR=KYR+1
C
C ***********************************************************************
C    ASK INFORMATION ABOUT ADDING NEW MACHINES. A SPECIAL ATTENTION HAS
C  BEEN PAID TO UPDATING THE CONFIGURATIONS FOR THE NEXT YEAR, BUT AT THE
C  SAME SAVING THE OLD INFORMATION IN THE WAY WHICH IS EFFICIENT FOR
C  INFORMATION RETREIVING LATER ON.
C    NCOUNT(I)=THE ORDER NUMBER OF THIS YEAR'S CONFIGURATIONS WHICH WILL BE
C  THE I-th CONFIGURATION FOR THE NEXT YEAR.
C    CONCODE(I) IS COMBINED BINARY CODE FOR STORING THE CONFIGURATION
C  INFORMATION.  CONCODE(I) STORES THE DECISION BRANCH INFORMATION FOR THE
C  I-th CURRENT CONFIGURATION. IF IT IS THE 4th YEAR, THEN FOR THE 1st
C  CONFIGRATION OF THIS YEAR CONCODE(1) WILL TELL YOU HOW THIS CONFIGURATION
C  WAS DEVELOPED FROM THE 1st YEAR. CONCODE(1) WOULD BE AN 8-DIGIT NUMBER
C  WITH 2-DIGIT CORRESPONDING TO EACH YEAR.
C ************************************************************************
C
       WRITE(6,*)'GIVEN THE RESULTS ABOVE, ARE THERE ANY CONFIGURATION',
     1  ' THAT ARE NOT WORTHY OF FURTHER CONSIDERATION?  IF THERE ARE,',
     1  ' FATHOM THESE CONFIGURATIONS.'
       WRITE(6,*)'HOW MANY CONFIGURATIONS ARE TO BE THE EVALUATED',
     1  ' FURTHER ?'
       READ(6,*) NCNF(KYR)
       IF (NCNF(KYR).EQ.0)  GO TO 4000
       WRITE(6,*)'PLEASE LIST THE CONFIGURATION NUMBERS',
     1  ' IN INCREASING ORDER.'
       WRITE(6,*)'ONE LINE FOR EACH CONFIGURATION.'
       DO I=1,NCNF(KYR)
         READ(6,*) NCOUNT(I)
         CONCODE(I)=CONCODE(NCOUNT(I))
       END DO
C
```

```
C ****************************************************************************
C    ASK THE INFORMATION OF THE NEXT YEAR.
C *****************************************************************************
         WRITE(6,12) KYR
12       FORMAT(/' NEXT YEAR IS, THE ',I3,'th YEAR.')
         DO I=1,NPART
           WRITE(6,13) PIDEN(I),AVDEM(I)
13       FORMAT(' THE AVERAGE DEMAND FOR PRODUCT ',A8,' WAS ',F10.0,
     1   ' UNITS/YR')
           WRITE(6,*)'WHAT WILL IT BE FOR THE NEXT YEAR?'
           READ(6,*) AVDEM(I)
           WRITE(6,*)'THIS DEMAND MAY CHANGE AN AMOUNT OF X UP OR DOWN.',
     1   'WHAT IS THE VALUE OF X?'
           READ(6,*) AA(I)
           WRITE(6,*)'IF PRODUCT ',PIDEN(I),'NEEDS SUBCONTRACT, WHAT IS THE ',
     1   'PRICE FOR SUBCONTRACT?'
           READ(6,*) SUBCOST(I)
         END DO
C
C *************************************************************************
C        ASK FOR INFORMATION OF CHANGES IN THE NEXT YEAR
C *************************************************************************
C
         NPASS=0
         DO I=1,NCNF(KYR)
           DO J=1,NPART
             REQMT(J)=AVDEM(J)
             NOPS(J)=NOPS1(J)
             DO K=1,NOPS(I)
               NST(J,K)=NST1(J,K)
               PTO(J,K)=DD(J,K)
             END DO
           END DO
           DO J=1,M
             NS(J)=NSO(I,J,KYR-1)
           END DO
           WRITE(6,10) I
10       FORMAT(' FOR THE ',I3,'th CONFIGURATION SELECTED ABOVE,',
     1   ' PLEASE CHECK THE FOLLOWING :')
           CALL CANQ(1)
           ADINVES1(I,KYR)=0.
           NADDM=0
           DO J=1,MM
             NS(J)=NS(J)-NSO(I,J,KYR-1)
             WRITE(6,11) NS(J),NAME(J)
11       FORMAT('  TO MEET PRODUCTION CAPACITY WITHOUT SUBCONTRACTING, ',
     1   I3,'UNITS OF ',A8,' ARE REQUIRED.')
             WRITE(6,*)'  HOW MANY OF THIS TYPE OF MACHINES WOULD YOU LIKE
     1   TO ADD FOR THE NEXT YEAR?'
             READ(6,*) NSHIFT(J)
             IF (NSHIFT(J).EQ.0) GO TO 15
```

```
            WRITE(6,*)'WHAT IS THE COST OF EACH MACHINE ADDED.'
            READ(6,*) ADINVESO
            ADINVES1(I,KYR)=ADINVES1(I,KYR)+ADINVESO*NSHIFT(J)
            NADDM=NADDM+NSHIFT(J)
15          CONTINUE
          END DO
          WRITE(6,*)'WHAT IS THE ENGINEERING CHANGE COST?'
          READ(6,*) CHANCOSO
          IF (NADDM.GT.0) THEN
            DO J=1,MM
              NSO(I+NPASS,J,KYR)=NSO(NCOUNT(I),J,KYR-1)
              ADINVEST(I+NPASS,KYR)=0.
              NSO(I+NPASS+1,J,KYR)=NSO(NCOUNT(I),J,KYR-1)+NSHIFT(J)
              ADINVEST(I+NPASS+1,KYR)=ADINVES1(I,KYR)
              CHANCOST(I+NPASS)=CHANCOSO
              CHANCOST(I+NPASS+1)=CHANCOSO
            END DO
            SHIFT(I+NPASS)=CONCODE(I)*100+I+NPASS
            SHIFT(I+NPASS+1)=CONCODE(I)*100+I+NPASS+1
            NPASS=NPASS+1
          ELSE
            DO J=1,MM
              NSO(I+NPASS,J,KYR)=NSO(NCOUNT(I),J,KYR-1)
              ADINVEST(I+NPASS,KYR)=0.
              CHANCOST(I+NPASS)=CHANCOSO
            END DO
            SHIFT(I+NPASS)=CONCODE(I)*100+I+NPASS
          END IF
        END DO
        NSH=NCNF(KYR)
        NCNF(KYR)=NCNF(KYR)+NPASS
        DO I=1,NCNF(KYR)
          CONCODE(I)=SHIFT(I)
      WRITE(6,*)'CODE =',CONCODE(I)
          NOCH(I)=0.
        END DO
        CALL COSTEVAL(1)
        DO I=1,20
          SHIFT(I)=0
          DO J=1,50
            AVCAP(I,J)=0.
          END DO
          DO J=1,20
            AVBOT(I,J)=0.
          END DO
        END DO
        GO TO 900
C
C ****************************************************************************
C    FINAL REPORT.
C  ***************************************************************************
```

```
C
5000     WRITE(6,24) MYR
24       FORMAT(' NOW IT IS THE END OF ',I3,'th YEAR')
         CALL COSTEVAL(2)
         WRITE(6,*)'GIVEN THE RESULTS ABOVE, SPECIFY THE BEST ONE: '
         READ(6,*) I
         WRITE(6,25)'************************************************'
25       FORMAT(3X,48A,//,5X,'FINAL DOMINATING DYNAMIC CONFIGURATION IS :')
         DO J=1,MYR
             LYR=MYR-J+1
             N1=CONCODE(I)/(100**LYR)
             NCONVERT=(CONCODE(I)/(100**LYR)-N1)*100
             WRITE(6,26) J
26       FORMAT(' IN THE ',I3,'th YEAR:')
             DO K=1,MM
               WRITE(6,27) NAME(K),NSO(NCONVERT,K,J)
27       FORMAT(1X,A8,'----',I3,'UNITS REQUIRED')
             END DO
             WRITE(6,28) COST(NCONVERT,J)
28       FORMAT(' TOTAL COST FOR THAT YEAR =',F10.0)
           END DO
           WRITE(6,29) PV(I)
29       FORMAT(' PRESENT VALUE OF THIS CONFIGURATION =',F10.0)
         WRITE(6,30)'************************************************'
30       FORMAT(/,3X,48A,/)
         GO TO 6000
4000     WRITE(6,*)'NO COFIGURATIONS SATISFY YOU.'
6000     WRITE(6,*)'NOW IT IS TIME FOR YOU TO MAKE THE DECISION.'
         STOP
         END
C
C        RANDOM DEMAND GENERATOR
C
         FUNCTION RANDEM(IR)
         COMMON/TWO/AA(50),BB(50),CC(50,20,20),DD(50 20),EE(50,20),
       1 AVDEM(50),NOPAV1(50),NOPAV2(50),NST1(50,20),NOPS1(50),NOCH(50),
       1  ISEED
         R=RAN(ISEED)
         LA=AVDEM(IR)-AA(IR)
         RANDEM=LA+AA(IR)*R
         RETURN
         END
C
C        RANDOM NUMBER OF OPERATIONS FOR JOB I
C
C        FUNCTION RNOP(I)
C        DIMENSION AA(50),BB(50),CC(50,20,20),DD(50,20),EE(50,20),AVDEM(50),
C      1 NOPAV1(50),NOPAV2(50),NST1(50,20),NOPS1(50),NOCH(50)
C        COMMON/TWO/AA,BB,CC,DD,EE,AVDEM,NOPAV1,NOPAV2,NST1,NOPS1,NOCH,ISEED
C        R=RAN(ISEED)
C        RNOP=NOPAV1(I)+(NOPAV2(I)-NOPAV1(I))*R
```

```
C     RETURN
C     END
C
C     RANDOM ROUTING GENERATE
C
      SUBROUTINE ROUTING(I)
      DIMENSION AA(50),BB(50),CC(50,20,20),DD(50,20),EE(50,20),
     1 AVDEM(50),NOPAV1(50),NOPAV2(50),NOCH(50),NST1(50,20)
      DIMENSION NOPS1(50),NS(50),NOPS(50),NST(50,20),PTO(50,20),
     1 REQMT(50),FRAC(50)
      COMMON/ONE/NS,NOPS,NST,PTO,REQMT,HOUR,M,NPART,PY,FRAC,IMAX
      COMMON/TWO/AA,BB,CC,DD,EE,AVDEM,NOPAV1,NOPAV2,NST1,NOPS1,
     1 NOCH,ISEED
      R=RAN(ISEED)
      IF (R.LE.BB(I)) THEN
        NOCH(I)=NOCH(I)+1
10      R=RAN(ISEED)
        NOPR=NOPAV1(I)+FLOAT(NOPAV2(I)-NOPAV1(I))*R+0.5
      WRITE(6,*)'RNOP=',RNOP
        IF (NOPR.EQ.NOPS1(I)) GO TO 10
        NOPS(I)=NOPR
      WRITE(6,*)'NOPS=',NOPS(I)
        DO J=1,NOPS(I)
          R=RAN(ISEED)
          NN1=NST(I,J)
          DO K=1,M
            IF (R.LE.CC(I,NN1,K)) THEN
              NST(I,J)=K
              PTO(I,J)=RPRCS(I,J)
              GO TO 18
            END IF
          END DO
18        CONTINUE
        END DO
      ELSE
        NOPS(I)=NOPS1(I)
        DO J=1,NOPS(I)
          NST(I,J)=NST1(I,J)
          PTO(I,J)=DD(I,J)
        END DO
      END IF
      RETURN
      END
C
C     RANDOM PROCESSING TIME GENERATE
C
      FUNCTION RPRCS(I,J)
      DIMENSION AA(50),BB(50),CC(50,20,20),DD(50,20),EE(50,20),AVDEM(50),
     1 NOPAV1(50),NOPAV2(50),NOCH(50),NST1(50,20),NOPS1(50)
      COMMON/TWO/AA,BB,CC,DD,EE,AVDEM,NOPAV1,NOPAV2,NST1,NOPS1,NOCH,ISEED
      R=RAN(ISEED)
```

```
      LA=DD(I,J)-EE(I,J)
      RPRCS=LA+EE(I,J)*R
      RETURN
      END
C
C     AFTER TAX COST EVALUATION
C
      SUBROUTINE COSTEVAL(ID)
      DIMENSION NS(50),NOPS(50),NST(50,20),PTO(50,20),REQMT(50),FRAC(50)
      DIMENSION NCNF(20),AVCAP(20,50),SUBCOST(50),COST(20,20),PV(20),
     1 AVBOT(20,20),NSO(20,20,20),ADINVEST(20,20),SHIFT(20),TOCAP(20)
      DIMENSION VARCOST(50),FIXCOST(20),DEPRECI(20),DECOST(20),
     1 SUBPEN(50),CONCODE(20),ENCOST(20),NCOUNT(20),
     1 ADINVES1(20,20),SALVAGE(20)
      CHARACTER*8 NAME(20),PIDEN(50)
      COMMON/FASE/NAME,PIDEN
      COMMON/ONE/NS,NOPS,NST,PTO,REQMT,HOUR,M,NPART,PY,FRAC,IMAX
      COMMON/THREE/NCNF,AVCAP,SUBCOST,COST,PV,ADINVEST,SHIFT,KYR,RETRNRT,
     1 MYR,AVBOT,NSO,CONCODE,ENCOST,NCOUNT,NSH,ADINVES1,TOCAP,NOCHAV,
     1 SALVAGE
      IF (ID.EQ.1) THEN
        GO TO 150
      ELSE IF (ID.EQ.2) THEN
        NSH=NCNF(KYR)
        NENYR=KYR
        GO TO 160
      END IF
      WRITE(6,*)'NOW EVALUATE THE COST FOR THE YEAR.'
      DO I=1,NCNF(KYR)
        VARCOST(I)=0
        DEPRECI(I)=0
        SUBPEN(I)=0
        TOCAP(I)=0.
        PV(I)=0.
        SALVAGE(I)=0.
        WRITE(6,19) I
19    FORMAT('  CONFIGURATION',I3' :')
        DO J=1,MM
          WRITE(6,20) NAME(J),NSO(I,J,KYR)
20    FORMAT(' NUMBER OF MACHINE TYPE ',A8,' =',I3)
          WRITE(6,21) NAME(J),AVBOT(I,J)
21    FORMAT(' CHANCE OF MACHINE TYPE ',A8,' BECOMING BOTTLENECK=',F6.3)
        END DO
        DO J=1,NPART
          WRITE(6,*)'WHAT IS THE UNIT VARIABLE COST FOR PRODUCT',J,'?'
          READ(6,*) VARPRICE
          VARCOST(I)=VARCOST(I)+VARPRICE*AVCAP(I,J)
        END DO
        WRITE(6,*)'WHAT IS THE OTHER FIXED COST FOR THIS CONFIGURATION?'
        READ(6,*) FIXCOST(I)
        DO J=1,KYR
```

```
            LYR=KYR-J+1
            N1=CONCODE(I)/(100**LYR)
            NCONVERT=(CONCODE(I)/(100**LYR)-N1)*100
            DEPRECI(I)=DEPRECI(I)+ADINVEST(NCONVERT,J)/MYR
            SHIFT(I)=SHIFT(I)+ADINVEST(NCONVERT,J)*(KYR-J+1)/MYR
            SALVAGE(I)=SALVAGE(I)+ADINVEST(NCONVERT,J)
          END DO
          DEPRECI(I)=TAXRA*DEPRECI(I)
          SALVAGE(I)=SALVAGE(I)-SHIFT(I)
          DO J=1,NPART
            TEST=REQMT(J)-AVCAP(I,J)
            TOCAP(I)=TOCAP(I)+AVCAP(I,J)
            IF (TEST.GT.0) THEN
              WRITE(6,30) TEST,PIDEN(J)
30      FORMAT(F10.3,' UNIT(S) OF ',A8,' MUST BE SUBCONTRACTED.')
              TEST=TEST*SUBCOST(J)
              WRITE(6,31) TEST
31      FORMAT('THE SUBCONTRACT COST =',F10.0)
              SUBPEN(I)=SUBPEN(I)+TEST
            END IF
          END DO
          VARCOST(I)=VARCOST(I)+SUBPEN(I)+ENCOST(I)
          DECOST(I)=TAXR*(FIXCOST(I)+VARCOST(I))
          COST(I,KYR)=FIXCOST(I)+VARCOST(I)-DEPRECI(I)
     1  -DECOST(I)
          DO J=1,KYR
            LYR=KYR-J+1
            N1=CONCODE(I)/(100**LYR)
            NCONVERT=(CONCODE(I)/(100**LYR)-N1)*100
            IF (J.EQ.1) NX=NCOUNVERT
            K1=J
            K2=(RETRNRT+1)**K1
            PV(I)=PV(I)+COST(NCONVERT,J)/K2
          END DO
          PV(I)=PV(I)+COST(NX,MYR+1)
          WRITE(6,38) COST(I,KYR)
38      FORMAT(' COST WITHOUT NEW INVESTMENT =',F10.0)
          WRITE(6,39) PV(I)
39      FORMAT(' PRESENT VALUE WITHOUT NEW INVESTMENT =',F10.0)
        END DO
        RETURN
150     NENYR=KYR-1
160     WRITE(6,40)'---------------------------------------------'
40      FORMAT(/,10X,45A,//)
        WRITE(6,18) NENYR
18      FORMAT(10X,' REPORT FOR THE ',I3,'th YEAR :',/)
        DO I=1,NSH
          IF (KYR.EQ.MYR) THEN
            K=I
            ADINVES1(K,KYR)=0.
          ELSE
            K=NCOUNT(I)
```

```
          END IF
          ADX=ADINVES1(K,KYR)-ADINVES1(K,KYR)/10*TAXR
          COST(K,NENYR)=COST(K,NENYR)+ADX
          K1=KYR-1
          K2=(RETRNRT+1)**K1
          PV(K)=PV(K)+ADX/K2
          VARCOST(K)=VARCOST(K)-SUBPEN(K)
          WRITE(6,41) K
41        FORMAT(12X,'CONFIGURATION',I3' :',/)
          WRITE(6,42) TOCAP(K)
42        FORMAT(14X,'TOTAL CAPACITY OF FMS =',F10.0)
          WRITE(6,43) VARCOST(K)
43        FORMAT(16X,'VARIABLE COST =',F10.0)
          WRITE(6,44) FIXCOST(K)
44        FORMAT(16X,'FIXED COST =',F10.0)
          WRITE(6,45) SUBPEN(K)
45        FORMAT(14X, 'SUBCONTRACT COST =',F10.0)
          WRITE(6,46) ADINVES1(K,NENYR)
46        FORMAT(14X,'ADDITIONAL INVESTMENT =',F10.0)
          WRITE(6,47) NOCHAV
47        FORMAT(14X,'# OF ENGINEERING CHANGES =',I3)
          WRITE(6,48) ENCOST(K)
48        FORMAT(14X,'ENGINEERING COST =',F10.0)
          WRITE(6,49) COST(NCONVERT,NENYR)
49        FORMAT(14X,'AFTER TAX COST =',F10.0)
          WRITE(6,50) PV(K)
50        FORMAT(14X,'PRESENT VALUE =',F10.0,//)
        END DO
        WRITE(6,40)'---------------------------------------------'
        RETURN
        END

C
C   CANO ELEMENT WHICH WILL BE CALLED IN EACH SIMULATION RUN
C
      SUBROUTINE CANQ (INDEX)
C  INDEX    = 0 TO COMPUTE OUTPUT CAPACITY AND IDENTIFY BOTTLENECK ONLY
C           = 1 TO DETERMINE THE MACHINE REQUIREMENT FOR A FORECASTED OUTPUT
      DIMENSION NS(50),Q(50),PT(50),G(250),NOPS(50),
     1          FRAC(50),REQMT(50),APTS(50),
     1          V(50),R(50),NST(50,20),PTO(50,20),F(250)
      COMMON/ONE/NS,NOPS,NST,PTO,REQMT,HOUR,M,NPART,PY,FRAC,IMAX,MM,
     1           AVETRANS
C
C  COMPUTE EACH PRODUCT TYPE'S SHARE OF TOTAL PRODUCTION REQUIREMENT
C
      TOTAL=0.
      PT(MM)=AVETRANS
      DO 35 K=1,NPART
      TOTAL=TOTAL+REQMT(K)
35    CONTINUE
```

```
      DO 40 K=1,NPART
40    FRAC(K)=REQMT(K)/TOTAL
C
C  PRE-PROCESS ROUTING DATA FOR USE IN QUEUEING-NETWORK ANALYSIS
C
51    DO 160 I=1,M
      Q(I)=0.
      V(I)=0.
      PT(I)=0.
      APTS(I)=0.
160   CONTINUE
      ANOP=0.
      DO 255 K=1,NPART
      IF(FRAC(K).LE.0)GOTO 255
      NEND=NOPS(K)
        DO 200 NO=1,NEND
        NSTA=NST(K,NO)
        V(NSTA)=V(NSTA)+1
        APTS(NSTA)=APTS(NSTA)+PTO(K,NO)
200     CONTINUE
      ANOP=ANOP+FRAC(K)*NOPS(K)
        DO 235 J=1,M
        PT(J)=PT(J)+APTS(J)*FRAC(K)
        Q(J) =Q(J) +V(J)    *FRAC(K)
        APTS(J)=0.
        V(J)=0.
235     CONTINUE
255   CONTINUE
260   CONTINUE
C
C  COMPUTE RELATIVE WORKLOAD AND IDENTIFY BOTTLENECK
C
        WRITE(6,*)'AN=',ANOP
      Q(MM)=1./ANOP
      R(MM)=PT(MM)
      RMAX=R(MM)/FLOAT(NS(MM))
      IMAX=MM
      DO 270 I=1,M
        WRITE(6,*)'Q(',I,')=', Q(I)
      IF(Q(I).GT.0.)PT(I)=PT(I)/Q(I)
      Q(I)=Q(I)/ANOP
      R(I)=Q(I)*PT(I)
      TEST=R(I)/FLOAT(NS(I))
      IF(TEST.LE.RMAX)GOTO 265
      IMAX=I
      RMAX=TEST
265   CONTINUE
270   CONTINUE
      N=-1
      DO 306 I=1,M
306   N=N+NS(I)
```

```
      IF(N.LE.1)N=2
C
C  COMPUTE G MATRIX
C
      NN=N+5
      DO 325 J=1,NN
      F(J)=0.
      G(J)=0
325   CONTINUE
      DO 360 I=1,MM
      IF(R(I).LE.0) GOTO 355
      TEMP=1
      IF(NS(I).GT.1)GOTO 335
C
C  SINGLE SERVER CASE
C
        DO 330 J=1,NN
        TEMP=R(I)*TEMP+G(J)
        G(J)=TEMP
330     CONTINUE
      GOTO 350
C
C  MULTIPLE SERVER CASE
C
335   CONTINUE
        WRITE(6,*)'NN=',NN
        DO 345 J=1,NN
        JJ=NN-J+1
          DO 340 K=1,JJ
          A=FLOAT(NS(I))
          CHK=FLOAT(JJ-K+1)
          IF(CHK.LT.A)A=CHK
          TEMP=(R(I)*TEMP)/A+G(K)
340       CONTINUE
        G(JJ)=TEMP
        TEMP=1
345     CONTINUE
350   CONTINUE
355   CONTINUE
360   CONTINUE
C
C  COMPUTE AND PRINT SYSTEM PERFORMANCE
C
        WRITE(6,*)'G(N)=',G(N)
      P=60.*Q(MM)*G(N-1)/G(N)
      PY=P*HOUR
      IF(TOTAL.LE.PY)IMAX=0
      IF(INDEX.LE.0.OR.TOTAL.LE.PY) RETURN
      NS(IMAX)=NS(IMAX)+1
      GOTO 51
      END
```

REFERENCES

AGEEM80 Agee, M. H., and J. Ma. Tanchoco, "Cost Factors for Justifying Projects," Plant Engineering, October 16, 1980.

ARCUA66 Arcus, A. L., "COMSOAL - A Computerized Method of Sequencing Operations for Assembly Lines," International Journal of Production Research IV, 4 (1966), pp. 259-277.

BAZAM77 Bazaraa, M. S. and J. Jarvis, Linear Programming and Network Flows, New York: John Wiley & Sons Inc., 1977.

BERMO82 Berman, Oded, "Efficiency and Production Rate of a Transfer Line with two Machines and Finite Storage Buffer," European Journal of Operations Research, IX, 3 (1982), pp. 295-308.

BELLD73 Bell, D., The Coming of Post-Industrial Society: A Venture in Social Forecasting, New York: Basic Book, 1973.

BERTR73 Bertha, R. W., "Will Your Assembly System Be a Winner ?," Automation, September (1973).

BOOTG68 Boothroyd, G. and A. H. Redford, Mechanized Assembly, London: McGraw-Hill Publishing Co., 1968.

BOXGE69 Box, G. E. P. and N. R. Draper, Evolutionary Operations, New York: John Wiley and Sons Inc., 1969.

BOWEH66 Bowen, H. R. and G. L. Mangum (ed.), Automation and Economic Progress, Englewood Cliff, New Jersey: Prentice-Hall, Inc., 1966.

BRUMS71 Brumelle, S., "On the Relation Between Customer and Time Averages in Queues," Journal of Applied Probability, VIII, (1971), pp. 508- 520.

BUCKW61 Buckingham, W., Automation, New York: Harper and Row , Publishers, Inc., 1961.

BUFFE72 Buffa, E. S. and W. H. Taubert, *Production-Inventory Systems: Planning and Control*, Homewood, Illinois: Richard D. Irwin Inc., 1972.

BURKP56 Burke, P. J., "The Output of a Queueing System", *Operations Research*, IV, 6, (1956), pp. 669- 704.

BUXLG73 Buxley, G. M. , H. D. Slack, and R. Wild, "Production Flow Line System Design - a Review," *AIIE Transactions*, V, 1 (1973), pp. 37-48.

BUZAJ74 Buzacott, J. A., On the Optimal Control of Input to a Job Shop, ORSA- TIMS Meeting, Boston (1974).

BUZAJ76 Buzacott, J. A. "The Production Capacity of Job Shops with Limited Storage Space," *International Journal of Production Research*, Vol XXIV, No. 5, pp. 597- 606 (1976).

BUZAJ78 Buzacott, J. A., "Models of Automatic Transfer Lines with Inventory Banks - A Review and Comparison," *AIIE Transactions*, Vol X, No. 2, pp. 197- 207 (1978).

BUZAJ80 Buzacott, J. A., "Models for Understanding Flexible Manufacturing Systems," *AIIE Transactions*, (December, 1980).

BUZAJ82 Buzacott, J. A., "Optimal Operating Rules for Automated Manufacturing Systems," *IEEE Transactions, Automatic Control*, V-AC-27n1 (February, 1982), pp. 80-86.

BUZEJ73 Buzen, J. P., "Computational Algorithm for Closed Queueing Networks with Exponential Servers," *Comm. Assoc. Comput. Mach*, Vol. XVI, No. 9, pp. 527- 531 (Sept., 1973).

BUZEJ75 Buzen, J. P., "Cost Effective Analytic Tools for Computer Performance Evaluation," *Proceedings of IEEE COMPCON*, IEEE, New York, pp. 293- 296, (1975).

CINLE75 Çinlar, E., *Introduction to Stochastic Processess*, Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

CHANT82 Chang, T. C., *TIPPS- A Totally Integrated Process Planning System*, Unpublished Ph. D. Dissertation, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg (1982).

CHENP80 Chen, P. H. and J. J. Talavage, *The Production Decision Support System for CMS*, NSF Grant No. APR74, 15256 Report No. 16, Indiana: Purdue University, 1980.

CHINF81 Chin, F. Y. and L. L. Tsai, "On J-Minimal and J-Maximal FLow Shop Scheduling," *J. Association Computing Machines*, XXVIII, 3, (July, 1981).

CONWR67 Conway, R. W., W. L. Maxwell and L. W. Miller, *Theory of Scheduling*, Reading, Massachusetts: Addison- Wesley Publishing Company, 1967.

COOKN75 Cook, N. H., "Computer Managed Parts Manufacture," *Scientific American*, (February, 1975), pp. 22- 29.

DAREL73 Dar-El, E. M., "MALB - A Heuristic Technique for Balancing Single-Model Assembly Lines," *AIIE Transactions*, V, 4, (1973), pp. 343-356.

DAREL75 Dar-El, E. M., "Solving Large Single- Model Assembly Line Balancing Problems - A Comparative Study," *AIIE Transactions*, VII, 3, (1975), pp. 302-310.

DEVRM77 DeVries, M. F., "Two Case Studies of the Investment Justification for N/C Machines," *MAPEC (The Manufacturing Productivity Education Committee)*, August, 1977.

DISNR83 Disney, R. L. and D. Konig, "Queueing Networks: A Survey of Their Theory and Some of Their Applications," Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, (1983).

DRAPL81 "Flexible Machining System Quarterly Progress Report," The Charles Stark Draper Laboratory, Inc., Cambridge (October, 1981).

EILOS69 Eilon, S., "A Simpler Proof of $L= \lambda W$," *Operations Research*, XXVII, (1969), pp. 915- 917.

FISHG73 Fishman, G. S., "Statistical Analysis for Queueing Simulation," *Management Science*, XX, 3 (1973).

FISHG78 Fishman, G. S., "Grouping Observations on Digital Simulation," *Management Science* XXIV, 5 (1978).

FRANR74 Francis, R. L. and J. A. White, *Facility Layout and Location - An Analytical Approach*, Englewood Cliff, New Jersey: Prentice-Hall, Inc., 1974.

GALLH59 Galliher, H. P. *Notes on Operations Research*, Cambridge, Mass.: M.I.T. Technology Press, Operations Research Center, 1959, Chapter 4.

GELDL78 Gelders, L. F. and N. Sambandam, "Four Simple Heuristics for Scheduling a Flow Shop," *International Journal of Production Research*, XVI, 3 (1978), pp. 221-231.

GENSG81 Gens, G. V. and E. V. Leuner, "Fast Approximation Algorithm for Job Shop Sequencing with Deadlines," *Discrete Applied Mathematics*, III, 4 (1981), pp. 313-318.

GERWD81 Gerwin, D., "Control and Evaluation in the Innovation Process - the Case of FMS," *IEEE Transactions, Engineering Management*, VEM-28n3 (August, 1981), pp. 62-70.

GIAMT76 Giammo, T., "Validation of a Computer Performance Model of the Exponential Queueing Network Family," *Proceedings of the Int. Symposium on Computer Performance Modelling*, Measurement, and Evaluation, Harvard University, (1976).

GORDW67 Gordon, W. J., and G. F. Newell, "Closed Queueing Systems with Exponential Servers," *Operations Research*, 15 (1967), pp. 254-265.

GRAVS81 Graves, S. C., "Review of Production Scheduling," *Operations Research*, XXIX, 4 (1981), pp. 646-675.

GREEJ65 Greene, J. H., *Production Control, Systems and Decisions*, Illinois: R. D. Irwin Inc., 1965.

GUDNC74 Gudnason and E. N. Corlett, (ed.) *Development of Production Systems*, London: Taylor and Francis, 1974.

HARRJ73 Harrington, J., Computer Integrated Manufacturing, New York: Industrial Press Inc., 1973.

HAYSP82 Hays, P. and E. W. Zimmers Jr., "A Guide for Application and Justification of a Flexible Manufacturing System," MAPEC(The Manufacturing Productivity Education Committee), January, 1982.

HELGW61 Helgeson, W. B. and Birnie, D. P., "Assembly Line Balancing Using Ranked Positional Weight Technique," J. of Industrial Engineering, XII, 6 (1961), pp. 394-398.

HILLF80 Hillier, F. S. and G. J. Lieberman, Introduction to Operations Research, San Francisco: Holden-Day, Inc., 1973.

HORDA81 Hordijk, A. and Nico Van Dijk, "Network of Queues with Blocking," 8th Intern'l Symposium on Computer Perf Modelling, Measurement and Eval," Amsterdam: North Holland Publishing Co., 1981.

HOREY78 Horev, Y., N. H. Cook, and J. E. Ward, "Discrete Simulation of Flexible Manufacturing Systems," Report No. ESL-FR-834-4, M.I.T., (1978).

HOWAR66 Howard, R. A., "Decision Analysis : Applied Decision Theory," Proceedings of the 4th International Conference on Operations Research, pp. 55- 71, Wiley, 1966.

HUGHP73 Hughes, P. H. and G. Moe, "A Structural Approach to Computer Performane Analysis," Proceedings AFIPS National Computer Conference, Vol. 42, AFIPS Press, Montvale, N.J., pp. 109- 119.

HUTCG73 Hutchinson, G. K. and B. E. Wynne, "A Flexible Manufacturing System," J. of Industrial Engineering. December 1973, pp. 10- 17.

HUTCG76 Hutchinson, G. E., "Production Capacity: CAM vs Transfer Lines," J. of Industrial Engineering, September 1976, pp. 30- 35.

HUTCG84 Hutchinson, G. E., and J. R. Holland, "The Economic Value of Flexible Automation," Journal of Manufacturing Systems, Vol. 1, No. 2, pp. 215 - 227.

JACKJ57 Jackson, J. R., "Networks of Waiting Lines," *Operations Research*, V,(1957), pp. 518- 521.

JACKJ63 Jackson, J. R., "Jobshop Like Queueing Systems," *Management Science*, X, 10 (1963), pp. 131- 142.

JENSP80 Jensen, P. A. and J. W. Barnes, *Network Flow Programming*, New York: John Wiley & Sons Inc., 1980.

JEWEW67 Jewell, W., "A Simple Proof of $L= \lambda W$," *Operations Research*, XV, (1967), pp. 1109- 1116.

JOHNL74 Jonhson, L. A. and D. C. Montgomery, *Operations Research in Production Planning, Scheduling, and Inventory Control*, New York: John Wiley & Sons, 1974.

KANEJ81 Kanet, J. J., "Minimizing Variation of Flow Time in Single Machine Systems," *Management Science*, XXVII, 12 (1981), pp. 1453- 1464.

KELLF79 Kelly, F., *Reversibility and Stochastic Networks*, New York: John Wiley & Sons, 1979.

KELLF82 Kelly, F., "Networks of Quasi- Reversible Nodes," *Applied Probability- Computer Science: The Interface, Volume I*, Boston: Birkhauser (1982).

KILBM61 Kilbridge M. D., and L. Wester, "A Heuristic Method of Assembly Line Balancing," *J. of Industrial Engineering*, XII, 4 (1961), pp. 292- 298.

KIMEJ78 Kimemia, J. and S. B. Gordon, "Multicommodity Network Flow Optimization in Flexible Manufacturing Systems," Report No. ESL-FR-834-2, M.I.T.,(Sept., 1978).

KLAHT83 Klahorst, H. T., "How to Plan Your FMS," *Manufacturing Engineering*, Vol. 91, No. 3, pp. 52- 54.

KLEIJ74 Kleijnen J. P. C., *Statistical Techniques in Simulation*, New York: Marcel Dekker, Inc., 1974.

KLEIL75 Kleinrock, L., *Queueing Systems, Volume I: Theory*, New York: John Wiley and Sons Inc., 1975.

KLEIL76 Kleinrock, L., *Queueing Systems, Volume II: Computer Applications*, New York: John Wiley and Sons Inc., 1976.

LAWAV77 Law, A. M., "Confidence Intervals in Discrete Event Simulation: A Comparison of Replication and Batch Means", *Naval Research Logistics Quarterly*, XIV, (1977), pp. 667- 678.

LEIMF81 Leimkuhler, F. F., *Economic Analysis of CMS*, NSF Grant No. APR74 15256 XXI, February 1981.

LENZJ77 Lenz, J. E. and J. J. Talavage, *Generalized Computerized Manufacturing Systems Simulator (GCMS)*, NSF Grant No. APR74 15256 VII, August 1977.

LEWIW80 Lewis, W. C., M. M. Barash and J. J. Solberg, *Data Flow CMS Control System Architecture*, NSF Grant No. APR74 15256 XVII, December 1980.

LIPSL77 Lipsky, L. and J. D. Church, "Application of a Queueing Network Model for a Computer System,' *Computing Survey*, IX, (1977), pp. 205- 221.

LITTJ61 Little, J., "A Proof of the Queueing Formula L= ¼W," *Operations Research*, IX, (1961), pp. 383- 387.

LUCER57 Luce, R. D., and H. Raiffa, *Games and Decision*, Wiley, 1957.

LUKEH72 Luke, H. D., *Automation for Productivity*, New York: John Wiley and Sons Inc., 1972.

MAGAE72 Magad, E. L. "Cooperative Manufacturing Resource," *J. of Industrial Engineering*, IV, 1 (1972), pp. 36- 40.

MANSE64 Mansoor, E. M., "Assembly Line Balancing- An Improvement on the RPW Technique," *J. of Industrial Engineering*, XV, 2 (1964), pp. 73- 77.

MASTA70 Mastor, A. A., "On Exper'l Investigation and Comparative Eval of Prod'n Line Balancing Tech," *Management Science*, XVI, 11 (1970), pp. 728- 746.

MAXWW70 Maxwell, W., "On the Generality of the Equation L= λW," *Operations Research*, XVIII, (1970), pp. 172- 174.

MAYER76	Mayer, R. J., and J. J. Talavage, Simulation of a Computerized Manufacturing System, NSF Grant No. APR74 15256 IV, November 1976.
MCKEJ82	McKenna, J. , D. Mitra, and K. G. Ramakrishna, "A Class of Closed Markovian Queueing Networks," Applied Probability- Computer Science: The Interface, Volume II, Boston: Birkhauser (1982), pp. 453- 456.
MERCM77	Merchant, M. E., "The Inexorable Push for Automated Production," Production Engineering, January 1977, pp. 44- 49.
MERCM78	Merchant, M. E., "Computer Advancing World Manufacturing Technology," Modern Machine Shop, June 1978, pp. 123- 131.
MODMH82	Anonymous, "Flexible Manufacturing System -- Their Tremendous Potential," Modern Material Handling, Sept. 1982.
MUELH81	Mueller- Merbach, Heiner, "Heuristics and Their Design: A Survey," European Journal of Operations Research, VIII, 1 (1981), pp. 1- 23.
MULLB81	Muller, B., "Decomp Meth in the Const'n and Numerical Sol'n of Queueing Network Models," 8th Inter'l Symposium on Computer Perf Model, Measurement and Evaluation Amsterdam: North Holland Publishing Co., 1981, pp. 99- 112.
NEWEG71	Newell, G., Applications of Queueing Theory, London: Chapman and Hall, 1971.
NIENO75	Nie, N., C. H. Hull, J. G. Jenkins, K. Steinbrenner, and D. H. Bent, Statistical Package for the Social Sciences, New York: McGraw-Hill Book Co., 1975.
PADBM83	Padberg, M., H. Crowder, and E. L. Johnson, "Solving Large-Scale Zero-One Linear Programming Problems," Operations Research, Vol. 31, No. 5, September-October 1983.
PLOSG67	Plossl, G. W. and O. W. Wright, Production and Inventory Control, Englewood Cliffs, New Jersey: Prentice-Hall, 1967.

PONDJ77 Pond, J. B. "On the Road to CAD/CAM," *Iron Age*, March/April 1977, pp. 37-44.

PRENT74 Prenting, T. O. and N. T. Thomopoulus, *Humanism and Technology in Assembly Line System*, Rochelle Park, New York: Spartan Books, Hayden Book Co., 1974.

PRITA79 Pritsker, A. A. B. and C. D. Pegden, *Introduction to Simulation and SLAM*, New York: John Wiley & Sons, 1979.

RAIFH68 *Decision Analysis : Introductory Lectures on Choices Under Uncertainty*, Addison - Wesley, 1968.

REICE57 Reich, E., "Waiting Time When Queues Are in Tandem", *The Annals of Mathematical Statistics*, XXVIII,(1957), pp. 768- 773.

REIMM82 Reiman, M. I., "The Heavy Traffic Diffusion Approximation for Sojourn Times In Jackson Network," *Applied Probability- Computer Science: The Interface, Volume II*, Boston: Birkhauser (1982), pp. 409- 421.

REXRL73 Rexrode, L. O. and Marion Kosem, "Are You Really for DNC," *Automation*, May 1973, pp. 68- 75.

ROSEC78 Rose, C. A., "A Measurement Procedure for Queueing Network Models of Computer Systems," *Computing Survey*, Vol. X, pp. 263- 280 (1978).

RUNNJ78 Runner, J. A., *CAMSAM: A Simulation Analysis Model for Computerized Manufacturing Systems*," NSF Grant No. APR74 15256 XIII, December 1978.

SAMEC82 Samuelson, C. L. and W. G. Bulgren, "Product Form Solution for Queueing networks with Poisson Arrivals and General Servers," *Applied Probability- Computer Science: The Interface, Volume II*, Boston: Birkhauser (1982), pp. 499- 500.

SAUEC82 Sauer, C. H., "Computational Methods for Product Form Queueing Networks," *Applied Probability- Computer Science: The Interface, Volume I*, Boston: Birkhauser (1982), pp. 211- 216.

SAWEJ70 Sawyer, J. H. F., *Line Balancing*, London: The Machining Publishing Co., Ltd. (1970).

SCHAG78 Schaffer, G., "Computers in Manufacturing," *American Machinist*, April 1978, pp. 115- 130.

SCHMJ70 Schmidt, J. W. and R. E. Taylor, *Simulation and Analysis of Industrial Systems*, Homewood, Illinois: Richard D. Irwirn, (1970).

SCHMJ74 Schmidt, J. W., "Statistical Analysis of Simulation," *Proceedings of the Second Annual Systems Engineering Conference*, Minneapolis: November, 1974.

SCHWP77 Schweitzer, P. J. "Maximum Throughput in Finite Capacity Open Queueing Networks with Product- Form Solutions," *Management Science*, XXIV,(1977), pp. 217- 223.

SCHWH82 Schwetman, H., "Some Computational Aspects of Queueing Network Models," *Applied Probability-Computer Science: The Interface, Volume II*, Boston: Birkhauser (1982), pp. 135- 155.

SCHWP82 Schweitzer, P. J., "Bottleneck Determination in Network of Queues," *Applied Probability- Computer Science: The Interface, Volume II*, Boston: Birkhauser (1982), pp. 471- 485.

SECCO78 Secco-Suardo, G., "Optimization of a Closed Network of Queues," Report No. ESL-FR-834-3, Electronic System Laboratory, M.I.T., (Sept.,1978).

SHAPW77 Sharpe, W. I., Jr., "Assembly Line Balancing Techniques," *Society of Manufacturing Engineers*, Paper MS77-313. 1977.

SOLBJ76 Solberg, J. J., "Optimal Design and Control of Computerized Manufacturing Systems," *Proceedings, AIIE Systems Engineering Conference*, Boston, MA (1976).

SOLBJ77 Solberg, J. J., "A Mathematical Model of Computerized Manufacturing Systems," *Production and Industrial Systems*, Muramatsu R., and N. A. Dudley, (ed.), pp. 1265- 1278, London: Taylor and Francis, 1977.

SOLBJ79 Solberg, J. J., "Stochastic Modelling of Large Scale Transportation Networks," Report No. DOT-ATC-79-2, School of Industrial Engineering, Purdue University, Indiana (June,1979).

SOLBJ80 Solberg, J. J., *CAN-Q User's Guide*, NSF Grant No. APR74 15256 IX, July 1980.

STECK77 Stecke, K. E. and J. J. Solberg, *Scheduling of Operations in a Computerized Manufacturing System* NSF Grant No. APR74 15256 X, December 1977.

STECK81 Stecke, K. E. and J. J. Solberg, *The CMS Loading Problem*, NSF Grant No. APR74 15256 XX, February 1981.

STECK83 Stecke, K. E., "Non-linear Integer Production Planning Problems," *Management Science*, Vol. 29, No. 3, pp. 273- 288.

STIDS72 Stidham, S., "$L= \lambda W$, A Discounted Analogue and a New Proof," *Operations Research*, XX, (1972), pp. 1115- 1126.

STIDS72 Stidham, S., "A Last Word on $L= \lambda W$," *Operations Research*, XXII, (1974), pp. 417- 421.

TAGUG78 Taguchi, G., "Performance Analysis Design," *Production and Industrial Systems*, Muramatsu R., and N. A. Dudley, (ed.), pp. 1201- 1210, London: Taylor and Francis, 1977.

WARDJ78 Ward, J. E., "Numerical Experiment with a Closed Network of Queue Model," Report No. ESL-FR-834-8, M.I.T. (Sept., 1978).

WILDR72 Wild, R., *Mass Production Management*, London: John Wiley & Sons Inc., 1972.

The two page vita has been removed from the scanned document. Page 1 of 2

The two page vita has been removed from the scanned document. Page 2 of 2