THE EFFECTS OF SYSTEM CONFIGURATION ON PRODUCTIVITY AND AVAILABILITY

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

LeRoy Zimmerman

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Operations Research

Georgia Institute of Technology

September, 1974

Copyright, September 1974

THE EFFECTS OF SYSTEM CONFIGURATION ON

PRODUCTIVITY AND AVAILABILITY

Approved:

Stuart J. Deutsch, Chairman

William W. Hines

Lynwood A. Johnson

John T. Berry

Date approved by Chairman: _____

ACKNOWLEDGMENTS

I take this opportunity to express my thanks and appreciation to Dr. Stuart Jay Deutsch for his continuous assistance, ingenuity, and motivation which enabled this research to be completed. My thanks are also extended to Dr.'s Hines and Berry for their guidance and assistance during the writing phase of the research effort. Thanks are also extended to Dr. Lynwood A. Johnson.

A very special "Thanks" to my wife for her assistance in preparing drafts and acting as my grammarian. Her patience and understanding were very instrumental in enabling this work to be completed. To my wife--Alice, and my son--Rodney, I affectionately dedicate this thesis.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS ii

LIST OF TABLES vii

LIST OF ILLUSTRATIONS viii

SUMMARY. x

Chapter

I. INTRODUCTION 1

1.1 Background 1
1.2 Research Objectives 2
1.3 Overview of the Research. 3

II. LITERATURE SURVEY. 6

2.1 System Reliability. 7

2.1.1 Definition of System Reliability. 7
2.1.2 Measuring System Reliability 8

2.2 System Availability 9

2.2.1 Terminology. 9
2.2.2 Instantaneous Failure Rate. 10

2.3 Availability of a Maintained System 11

2.3.1 Distributional Properties of Failure Mechanisms 12
2.3.2 Fail and Fix Policy 13
2.3.3 Periodic Policy 15

2.4 Productivity Versus Reliability and Availability. . 17

2.4.1 Mutsenek and Lobzov 18
2.4.2 Swanson Demand System 19

2.5 Availability and Productivity Models. 20

2.5.1 Applebaum's Model. 20
2.5.2 Buzacott-MTBF of Repairable Systems. . . . 23

Chapter Page

2.5.3 Models with Buffers 27

2.5.3.1 Barten 28
2.5.3.2 Techniques of Law et al. . . . 31

2.6 Availability of a Complex System 34

2.6.1 Description of a Subsystem. 35
2.6.2 A Closed Form Solution 36
2.6.3 Complex Systems--A Need for Simulation. . 41

2.7 Summary 41

III. BUILDING A GPSS MODEL FOR A COMPLEX LINE MANUFACTURING SYSTEM. 43

3.1 Complex Systems' Synthesis 44

3.1.1 Module Complexity. 44
3.1.2 A Modular Approach 44

3.2 Module Size Capability 48
3.3 Design of An Example Module. 51

3.3.1 Description of the Example Module . . . 51
3.3.2 Subsystem Configuration. 55
3.3.3 Module Operating Characteristics. . . . 57

3.4 The GPSS Program--A Scenario 58

3.4.1 Unit Production Routine. 58
3.4.2 Dummy Storage 61
3.4.3 Record of Production Time 62
3.4.4 Exit Routine 63
3.4.5 Failure Routines 63
3.4.6 Repair Routine. 66
3.4.7 Routine to Record System Availability . . 67

3.5 Verification of the Simulation Program 67

3.5.1 Production Flow Checks 68
3.5.2 Failure and Repair Routines 70
3.5.3 Record of System Down 71
3.5.4 Combining All Routines 72

3.6 Model Validation 73

3.6.1 Comparisons to Closed Form Results . . . 73

Chapter Page

3.6.1.1 Collapsing the Module to Independence 73
3.6.1.2 Removal of the Auxiliary Variables. 74

3.6.2 Comparison of Results 75

3.7 Modular Control Variables 80

3.7.1 In-Process Buffers 80
3.7.2 Regular and/or Periodic Maintenance . . . 82
3.7.3 Failure Rate 82
3.7.4 Production Rate 83
3.7.5 Repair Rate 83
3.7.6 Preloading. 83
3.7.7 Load Sharing 84
3.7.8 Equipment Sharing 84
3.7.9 Subsystem Down Criteria 85
3.7.10 System Down Criteria 85

3.8 Modular Responses. 86

3.8.1 Productivity 86
3.8.2 Availability 87

IV. THE EFFECTS OF THE MODULE CONFIGURATION ON PRODUCTIVITY AND AVAILABILITY 88

4.1 A 2^{27-22}_{III} Fractional Factorial Design 89

4.1.1 Levels of the Variables 89
4.1.2 Experimental Design. 91

4.2 The First Experiment. 94

4.2.1 The Responses. 94

4.2.1.1 Average Productivity and Availability 94
4.2.1.2 Temporal Productivity and Availability 96

4.2.2 The Analysis 99

4.2.2.1 Estimate of Contrast 99
4.2.2.2 A Scenario of Explanation. . . . 102

4.3 The Second Experiment 105

4.3.1 The Responses. 107

Chapter Page

4.3.2 Unaliased Estimate of Main Effects 107

4.3.2.1 On Productivity 107
4.3.2.2 On Availability 110

4.4 The Third Experiment 114
4.5 A Model of Productivity and Availability 115

4.5.1 Significance of Main Effects 119

4.5.1.1 ANOVA on Estimates for Productivity. 120
4.5.1.2 ANOVA on Estimates for Availability. 123

4.5.2 Maximizing Productivity and Availability . . 126

4.5.2.1 Steepest Ascent 126
4.5.2.2 Branch and Bound Mixed-Integer Programming. 132

4.6 The Transient and Steady-State Effects 135

4.6.1 A View Through Batch Size 135
4.6.2 On Productivity and Availability. 143

4.7 Module Output to Module Input 144
4.8 Comparison of Applebaum's and Stimulation Results . 149

V. CONCLUSIONS AND RECOMMENDATIONS 152

5.1 Conclusions 152

5.1.1 Model to Study Complex Systems 152
5.1.2 The Experiments 153
5.1.3 Maximizing Productivity and Availability . . 155
5.1.4 Transient and Steady-State Effects 156

5.2 Recommendations. 157

APPENDICES

A. COMPUTER PROGRAM 159

B. PROGRAM SYMBOLOGY. 179

C. NOTES TO THE USER. 183

BIBLIOGRAPHY 187

LIST OF TABLES

Table		Page
1.	Results of Validation Runs	78
2.	Listing of Variables and the Levels	90
3.	Design Matrix for 2^{27-22} Design (1st 32 Runs)	93
4.	Responses from First Fraction	95
5.	Main Effects and Aliased Two Factor Interactions	103
6.	Design Matrix for Second Fraction	106
7.	Responses from Second Fraction	108
8.	Results After Separating the Main Effects and the Aliased Two Factor Interactions for Productivity	109
9.	Results After Separating the Main Effects and the Aliased Two Factor Interactions for Availability	111
10.	Comparison of Top Ranked Variables for Productivity and Availability	113
11.	Responses from the Third Fraction	116
12.	Estimates of Variable 26 and Its Two Factor Interactions for Productivity	119
13.	ANOVA on Variables for Productivity	121
14.	ANOVA on Variables for Availability	124
15.	Path of Steepest Ascent for Productivity	129
16.	Path of Steepest Ascent for Availability	131
17.	Results from BABMIP Algorithm	134

LIST OF ILLUSTRATIONS

Figure		Page
1.	System Modeled by ARINC for Periodic Maintenance	16
2.	Systems Conforming to Applebaum's Model	21
3.	Example of a System Fitting Buzacott's Model	24
4.	System with In-Process Storage in a Sequential Process	29
5.	Physical Layout of a Manufacturing System	32
6.	Module Representing a Small System.	46
7.	Module as Part of a Larger System	47
8.	Limiting Configuration of Module for Sequential Operations.	49
9.	Parallel Configuration Within a Module	50
10.	Combination Module Configuration	52
11.	Module Diagram	53
12.	Subsystem Layout.	56
13.	Line Balance Concept	60
14.	Link-up Routine	64
15.	Validation Flow Chart	79
16.	Alternative Buffer Layouts	81
17.	Production Interval--Set 1, Run 4	97
18.	Production Interval--Set 1, Run 16.	98
19.	Average Production Interval--Batch Size 8	137
20.	Average Production Interval--Batch Size 25	138
21.	Average Production Interval--Batch Size 50	139
22.	Average Production Interval--Batch Size 75	140

Figure Page

23. Batch Size Constraints 142

24. Module Interfacing 145

25. Production Interval--First and Second Modules 148

SUMMARY

This thesis presents a technique for analyzing complex line manufacturing systems. The systems are defined as complex as a result of either their sizes or interdependencies of operations. In this research, the analyses are directed toward: (1) providing the effects of various configurations and operating policies upon a system's productivity and availability, (2) maximizing productivity and availability, (3) investigating the effects of transient- and steady-states upon batch size, and (4) a comparison of numerical estimates of availability with and without the assumption of independent subsystems.

The problem is attacked by the use of a modular systems approach. A GPSS program was written to accommodate modules ranging in size from a small system to a portion of a very large system. The program was validated by collapsing the module for comparison to an analytical model. Module size is limited to any configuration of subsystems, such that, the total number of in-process buffers and units within the subsystems does not exceed 65. The variables of the module include: (1) production time, (2) failure rate, (3) repair rate, (4) repair teams, (5) periodic maintenance, (6) load sharing, (7) equipment sharing, (8) preloading, (9) in-process buffers, (10) subsystem configuration, and (11) system configuration.

An example module was used to conduct experiments, using design of experiment techniques. The experiments were fractions of a 2^{27} design. A total of 96 simulation experiments were conducted.

Analyses of variance on the main effects showed the variables which have significant effect upon productivity and availability for the example module. Linear additive models were built for the responses, productivity and availability, as functions of the system configuration variables. Using these models, directional configuration alternatives for maximization of productivity and availability were obtained by steepest ascent. Further, a branch and bound mixed-integer programming algorithm was used to obtain the optimal levels of configuration variables for maximizing productivity subject to an availability constraint. From the experiments, the configuration constraints which limited batch size were identified.

It has been found that often the numerical estimates of availability of a system, which has dependent subsystems, are the same as when subsystems are assumed independent. However, that is not always the case. Therefore, although the estimates may be the same, the reasons they are at that level may be totally different.

The module designed in this research is sufficiently flexible to accommodate a wide range of other specific systems. Thus it serves as a tool for systems' design, evaluation of alternative systems configurations, and systems analysis, the identification of leverage points in a specific system, for the maximization of productivity and availability.

CHAPTER I

INTRODUCTION

1.1 Background

The state of the Nation's economy indicates a definite need for improvement in industrial productivity. From 1947 to 1973, the annual average rate of productivity in private business rose 3.1 percent. The period, 1973 to the middle of 1974, showed hardly any increase in output per man-hour (3). Thus productivity has not sufficiently increased to balance the rise in labor cost. Consequently, the prices of commodities have increased, leading to inflation. One alternative for curving the inflationary tendencies is having manufacturers improve productivity and reliability.

Line manufacturing systems, as a result of technological advances, have become modern complex systems. The complexity stems from growth in size and sophistication. Systems, once having independent subsystems, now have functionally dependent subsystems. Further, line manufacturing has increased complexity with such features as equipment sharing, modular subassembly configurations, in-process storage, periodic maintenance, preloading, load sharing, and others in different assorted combinations.

The resulting complex system configurations do not conform to system models previously reported (1, 7). These models have typically assumed independence of subsystems for analytical simplicity. Further, they do not account for the simultaneous existence of other operating

policies such as: (1) preloading, (2) in-process buffers, (3) equipment sharing, or (4) preventive or periodic maintenance policies.

Even in a simple system which has subsystems in series, the subsystems are not independent. If one subsystem fails then the remaining subsystems can be backed up with production or the condition can exist that others idle awaiting parts from the failed subsystem. In either case the others cannot fail since they are idle and thus a dependency is created.

Further models suggested for use to optimize systems have been limited to analyzing a specific phase of operation within the system. Consequently, not all the configuration constraints are considered simultaneously (4, 17, 28, 32). For example, computer simulation was used to optimize buffer size. The simulation did not consider varying the levels of any other of the system's features. Thus, the simulation results only provide a conditional optimal solution.

The analytics to develop a closed form solution to the problems of availability or productivity for complex systems can rapidly become very cumbersome as the complexity of the system increases. To overcome that handicap and at the same time assess productivity and system availability, computer simulation models have been used (4, 20). In many cases, the computer simulation models were designed for a specific system. Consequently, many of the viable configuration alternatives which can be included in a complex system were not capable of being investigated.

1.2 Research Objectives

The objectives of this research are fourfold. (1) The first

object is to design a model with sufficient flexibility to accommodate complex systems with any combination of the following configuration constraints: (a) failure rate, (b) repair rate, (c) production rate, (d) buffers, (e) equipment sharing, (f) preloading, (g) load sharing, (h) repair teams, (i) periodic maintenance, (j) subsystem configuration, and (k) system configuration. (2) The second objective is to prioritize the configuration constraints by ranking their effects on productivity and availability. (3) The third objective is to explore the simultaneous maximization of productivity and availability. Also, as part of the third objective is the identification of any commonality of systems configuration constraints, to gain insight into the functional relationship between productivity and availability. (4) The fourth objective is to establish the effects transient and steady-state have on productivity and availability.

1.3 Overview of the Research

The literature survey of the previously mentioned models and other findings on system reliability and productivity are presented in Chapter II. The model by Applebaum (1) was of particular interest because it is not as restrictive as some others. The computer simulation model by Law, et al. (20), also received greater attention because it deals with optimal buffer capacity, although it is restricted to a specific system. In addition to existing models for productivity and availability, Chapter II also includes an attempt at a closed form solution for availability of a complex module. The attempt encounters a very difficult problem and illustrates a need to use computer simulation to study the problem of system availability and productivity

of complex systems.

Chapter III primarily discusses the construction and operations of the stochastic model, which is in the form of a computer program. The model is presented using a time increment approach of following the sequence of events through the system in the order they occur. In addition to the time increment approach, the model is presented in the form of a scenario for the computer program. The latter is a discussion based on the program. The procedures and results of program verification and model validation are also included in Chapter III. Validation is followed by a complete description of the control variables and the modular responses.

Chapter IV delineates the design of experiment techniques used, followed by the responses and analyses for the first experiment. As a result of performing the second experiment, the unaliased estimates of main effects were obtained. Using the responses from the first, second, and a third experiment, the significance of main effects was ascertained for the model. The model was used to maximize productivity and availability using the method of steepest ascent. Maximization of the model was also done using a branch and bound mixed-integer programming algorithm.

Section 4.5 of Chapter IV discusses the effects that transient- and steady-state phenomenon have on productivity and availability with a view at its effects on batch size. The last section compares the results of Applebaum's model to those of the model under different given conditions.

Chapter V contains the conclusions resulting from the experi-

mentation and analyses based on the responses. In addition, the inferences which can be drawn from the projection of system productivity when the responses from one module serve as input to the next module are also in this chapter. The second half of Chapter V contains the recommendations based upon the findings of this research and additional questions which need to be investigated.

CHAPTER II

LITERATURE SURVEY

The literature in this chapter discusses the reliability and productivity of line manufacturing systems. Of particular interest are those systems with dependent subsystems. The literature survey includes cases where dependent subsystems are modeled as independent and modeled as dependent.

Section 2.1 presents some concepts of system reliability as they apply to manufacturing systems. Section 2.2 defines those terms used in describing system availability. Section 2.3 discusses distributional properties of failure mechanisms and two maintenance policies: (1) fail and fix, and (2) periodic.

The literature which simultaneously discusses productivity and reliability are in Section 2.4. In Section 2.5 four models for finding system availability are presented. Two of the models address system availability assuming subsystem independence. The third and fourth models are used to determine the effect that in-process buffers have upon system availability.

Section 2.6 contains an illustration of a subsystem with load sharing and the capability of changing the mean time to repair based on the number of units operational. After attempting to find a closed form solution for subsystem availability, the need for simulation is established to accommodate this problem and others of greater magnitude.

2.1 System Reliability

The concept of system reliability is not unique among engineers. To develop the meaning as it applies to this research, a system is defined as " . . . a collection of regularly interacting or interdependent components . . . acting as a unit in carrying out an implicitly or explicitly defined mission" (14). Systems can range from simple to complex with numerous possibilities in between. A system near the upper end of the spectrum--complex--is composed of subsystems which themselves may be classified as systems. The term subystem refers to " . . . a work station, several work stations, or a conglomerate of automation lineage which contributes to the unreliability of the line" (20).

2.1.1 Definition of System Reliability

Knowing the reliability of a system provides management with a probabilistic likelihood that the system is operational during any given time interval to determine the expected rate of production. Reliability is defined as " . . . that characteristic of an item expressed by the probability that it will perform its required functions in the desired manner under all relevant conditions and on the occasions or during the time intervals when it is required so to perform" (13). Knight, Smith, and Aeronautical Radio Incorporated Research Corporation, ARINC, define reliability as " . . . the probability the product will give satisfactory performance for a given period of time when used in the manner and for the purpose intended" (3, 19, 28). Caplen defines reliability as a ratio " . . . the number surviving at instant t to the number at start when t = 0, when there is a large

number of items on test" (9).

Reliability, as shown in Caplen's definition, is a function of a time interval. This time interval in manufacturing systems is the length of a normal production run (20). By combining the definitions for system and reliability, a cumbersome meaning for system reliability results. That definition reads . . . the probability that a collection of regularly interacting or interdependent components, acting as a unit in carrying out an implicitly or explicitly defined mission will give satisfactory performance for a given period of time when used in the manner and for the purpose intended. However, a more concise definition is presented by Law, et al. " . . . the probability of continuously producing products at a predesigned rate during normal production runs" (20). The term production run used in reference to a system is the interval from start-up to shutdown.

2.1.2 Measuring System Reliability

When measuring system reliability, there is a degree of subjectivity in defining reliability. For example, there is no way to pinpoint "satisfactory performance." In an attempt to minimize the subjectivity of "satisfactory performance," manufacturers of commercial products and the military often establish lower bounds for when a product is considered satisfactory. To illustrate the inefficiency of such a lower bound, consider a 10 watt transmitter, the product, which has a 9.9 watt output; 10 watts is the lower bound, therefore, 9.9 watts is unsatisfactory. However, 10.1 is satisfactory. For the actual user, the system may very well be satisfactory at 9.9. And he is in fact the individual using the equipment for its intended purpose

(19). A comparable illustration of this is taken from a production line. For one individual, satisfactory performance may be producing x items per hour, while for another it is the products produced having zero defects, and for yet another individual satisfactory performance may be producing x products with not more than 5 percent being defective. To counter this possibility of uncertainty, Lloyd and Lipow define reliability as " . . . the probability of a successful operation of a device in the manner and under the conditions of intended customer use" (21).

2.2 System Availability

The terms used to identify system availability have not been consistently applied in the literature. The first subsection, therefore, is used to establish the meaning of these terms as utilized in all consequent sections.

2.2.1 Terminology

The first term discussed is system availability. ARINC and Buzacott describe system availability or efficiency as:

> . . . the probability that the system is operating satisfactorily at any point in time when used under stated conditions, where the total time considered includes operating time, active repair time, administrative time, and logistic time (2,8).

Further, Buzacott also quantifies system availability as " . . . the proportion of time in some long interval the system is working, measured by the ratio of mean up time, MUT, to the mean time between failures" (7).

The length of time a subsystem is operational over the time period of last failure to next failure is called up time. In a

stochastic process, the up time is generated by a random variable. The expected value of the random variable for up time is known as mean up time, E(U) or MUT. When the time to repair is generated by a random variable, the expected value of the random variable is known as mean down time, E(D) or MDT. The sum of MUT plus MDT is called mean time between failure, MTBF.

2.2.2 Instantaneous Failure Rate

The distribution of the random variables for time to failure has been described by many distributions. They include the exponential, normal, log normal, Weibull, and gamma. The exponential distribution has been used in cases where all the failures common to the equipment during the stage of "break-in" or "early age" are assumed to have expired. In addition, the exponential distribution provides simplicity in determining the failure rate.

The failure rate over an interval is expressed as

$$FR(t_1,t_2) = \left[\frac{R(t_1) - R(t_2)}{R(t_1)}\right]\left[\frac{1}{(t_2 - t_1)}\right] ,$$

where $R(t) = \text{Prob}[T > t]$ and T is defined as time to failure. As the interval between t_1 and t_2 approaches zero, the closer the relationship for failure rate tends to an instantaneous failure rate. Thus, instantaneous failure rate, also known as the hazard, is the limiting value as Δt approaches zero. The hazard function is defined as

$$h(t) = \frac{f(t)}{R(t)} ,$$

where f(t) is the time to failure density.

In the case of the exponential failure law,

$$f(t) = \lambda e^{-\lambda t} \quad ; \quad t > 0$$

$$0 \quad ; \quad \text{otherwise},$$

$$R(t) = e^{-\lambda t} \quad ; \quad t > 0$$

$$0 \quad ; \quad \text{otherwise},$$

and

$$h(t) = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \; ; \; t > 0$$

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

Therefore, the instantaneous failure rate or hazard is λ, a constant. As a result, the hazard function is uniform and implies that if a component is functioning at time t, it is as good as new (17).

2.3 Availability of a Maintained System

Once failure has occurred, the maintainability of the subsystem is needed in assessing system availability. The availability of a system which is not maintained is simply the ratio of mean time to first failure to total run time. If the system is expected to have its first failure before the desired production level is attained, some form of maintenance is required to meet the production requirement.

Maintainability is defined as " . . . the probability that a failed system is restored to operable condition in a specified downtime" (2). Smith and Thomasson extend their definition ot maintainability to " . . . the probability that a device that has failed will be restored to operational effectiveness within a given period of time

when the maintenance action is performed in accordance with prescribed procedures" (28, 30).

The maintenance can be performed before a failure occurs, as well as after. Of the numerous maintenance policies, two are most often used. First, and frequently the case, is repair only when a failure occurs. The second, preventive maintenance policy, may be based on equipment age or based on a time cycle. In the case of age, equipment would be replaced at certain fixed times--known as periodic maintenance; the time cycle usually is marginal checking (10).

2.3.1 Distributional Properties of Failure Mechanisms

The distribution of operating and downtimes is commonly accepted as exponential (1). Law, et al., in their simulation of an actual production system, acknowledge that downtimes can be closely approximated by a negative exponential (20). Drenick, in his presentation of mathematical aspects of reliability gives the typical failure laws as exponential, normal, gamma, and the failure rate under the policy that a failed item is replaced or repaired to a like new condition (10). The inputs to maintainability are time to repair and the repair rate, M(t). Such that,

$$M(t) = 1 - e^{-ut}, \text{ where } u = \frac{1}{MDT} \text{ (27).}$$

In the works by Applebaum, Buzacott, et al., it is assumed that failures of subsystems are independent. However, when one subsystem fails, causing the system to fail, other stations become idle; consequently, they cannot fail. Therefore, subsystem failures are, more

often than not, dependent and the product law of availability is invalid (8). The criteria used for labeling subsystems as independently operating are: (1) stations have independent failures, and (2) stations have adequate facilities to initiate the repair of any subsystem when it fails (7).

The advantage of assuming independence of subsystems is--the systems MTBF can be computed in terms of the MTBF and availability of the individual subsystems (7). Buzacott, like Applebaum, conceded that the major limitation of his procedures is that frequently the assumption of independent failures and repairs is not valid (7, 1).

2.3.2 Fail and Fix Policy

The policy of maintenance immediately following failure is addressed by Epstein and Hasford (11). The paper derives formulae for evaluating reliability of three different, two-unit, redundant systems. The three systems are: (1) the units have a constant repair rate and only one unit operates at a time; (2) the units have a constant repair rate and both units operate whenever possible; and (3) the units have a constant repair time and only one unit operates at a time. The second system--constant repair rate and both units operating simultaneously--is the case of primary interest to this research, since the systems being studied have all units active, unless they are non-operational. Thus only the second system is discussed.

In this formulation, failure rate in λ and the up time is a random variable x whose probability density function, p.d.f., $f(x;\lambda)$, is

$$f(x;\lambda) = \lambda e^{-\lambda x}; \text{ for } x \geq 0 \text{ and } \lambda \geq 0$$

$$= 0 \quad ; \text{otherwise.}$$

The length of repair is also a variable, Y, whose P.d.f. is

$$f(y;u) = ue^{-uy}; \text{ for } y \geq 0 \text{ and } u \geq 0$$

$$= 0 \quad ; \text{otherwise.}$$

For a two-unit redundant system, difference equations are used to obtain a measure of reliability. The difference equations, upon differentiation, give rise to the differential equations:

$$P_2'(t) = -2\lambda P_2(t) + uP_1(t)$$

$$P_1'(t) = 2\lambda P_2(t) - (\lambda + u)P_1(t)$$

$$P_0'(t) = \lambda P_1(t) \quad , \quad \text{where}$$

State 2: both units operating,

State 1: one unit down and the other operating, and

State 0: both units down.

Thus, after evaluation of the differential equations where system availability, A_s, is defined as $1 - P_0(t)$. Therefore,

$$A_s = 1 - (1 - e^{-\lambda t})^2$$

$$= 2e^{-\lambda t} - e^{-2\lambda t},$$

and the expected length of time to reach state 0 is $3\lambda + u/2\lambda^2$. If there

can be no repairs, u = 0, the expected time to reach state 0 is $3\lambda/2\lambda^2$ or $3/2\lambda$, and the ratio $3\lambda + u/2\lambda^2$ to $3/2\lambda$ is $1 + u/3\lambda$, the measure of improvement due to repair.

2.3.3 Periodic Policy

ARINC presents a formulation for the mean life of a redundant system with a periodic maintenance policy. Figure 1 is an example of the type system which fits the model. Each of the paths represents a sequence of operations being performed. The paths are redundant. Periodic maintenance is scheduled for every T hours of operation starting at t = 0. Each part is checked and any failure is replaced by a new and statistically identical component. Thus, the system is restored to new condition after each maintenance action.

The derivation of the reliability function for a time period of t hours has

$$t = jT + \tau \quad j = 0,1,2 \ldots ; \ 0 \leq \tau \leq T.$$

When τ equals zero the expected period of continuous operation is some multiple of T. Thus,

$$R_T(t = T) = R(T) \ , \text{ for } j = 1$$

and

$$R_T(t = 2T) = R(T)^2 \ , \text{ for } j = 2.$$

Then, for cases where $0 < \tau < T$, the general case of the reliability function becomes,

$$R_T(t = jT + \tau) = [R(T)]^j \ R(\tau) \quad j = 0,1,2 \ldots$$

0 T

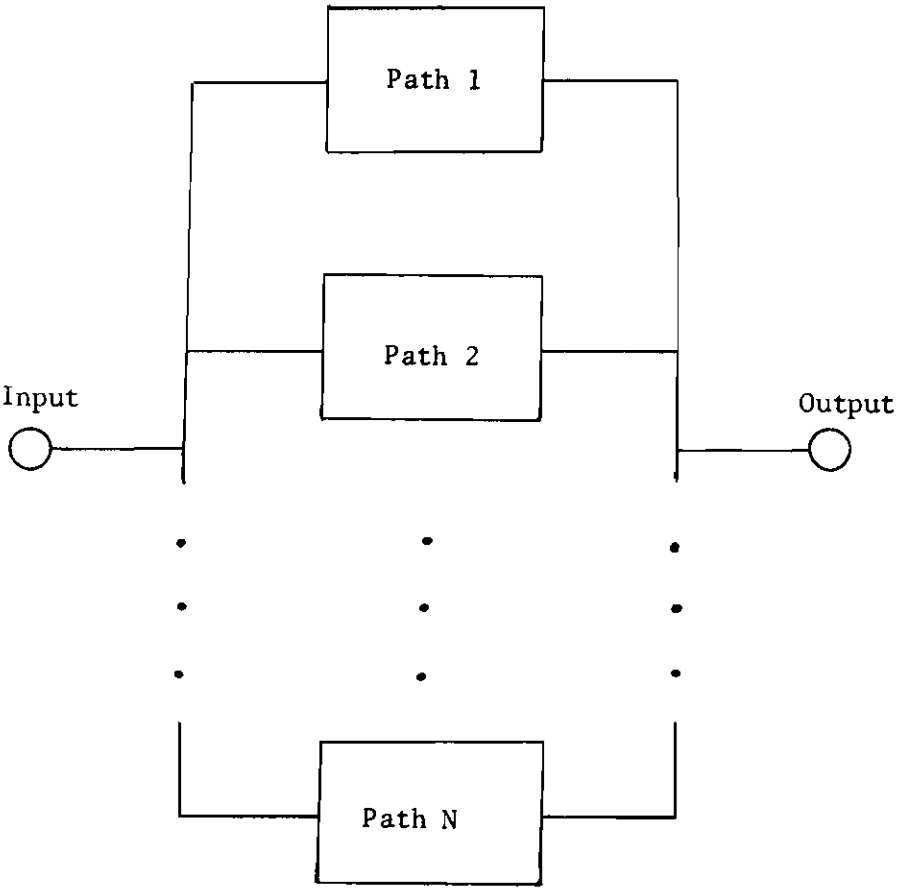


Figure 1. System Modeled by ARINC for Periodic Maintenance

The mean life of the system, θ_s, is computed by summing or integrating the reliability function over the limits 0 to T, where dt = d . Thus

$$\theta_s = \sum_{j=0}^{\infty} \int_0^T R_T(t)d\tau = \sum_{j=0}^{\infty} \int_0^T [R(T)]^j R(\tau)d\tau.$$

Simplified,

$$\theta_s = \frac{\int_0^T R(\tau)d\tau}{1 - R(T)} \qquad (2).$$

Therefore, in comparison to no maintenance, the mean life tends to increase as T decreases. However, this procedure lacks the consideration of replacement time. The expected economic gain or loss is not considered either.

2.4 Productivity Versus Reliability and Availability

Productivity is viewed as the level of output of satisfactory products. Products which do not meet production specifications are faulty and as such are rejects. Although the faulty product is an output from the system, it is not a countable item for production level. For that reason, the term satisfactory is used to describe the output included in system productivity. Productivity can be defined as " . . . the product of system utilization times the theoretical output" (23). The above definition assumes faultless output. When faulty output is not included in the system's production, productivity is decreased by the expected faulty output.

2.4.1 Mutsenek and Lobzov

The work of Mutsenek and Lobzov is a pioneering effort to correlate system reliability and productivity of automatic assembly machines. Prior literature has the production level equal to output, which included both good and faulty items. Thus when the faulty items are subtracted, system reliability is not directly related to productivity. Mutsenek and Lobzov established the basic standards to correlate reliability and productivity by computing the expected quantity of defective production.

One of the first standards from Mutsenek and Lobzov is the adoption of a faultless probability P(t). The production level can be such that all output is faultless, in which case P(t) = 1. With automatic assembly P(t) is likely to be less than 1. The probability of a defect can be expressed as

$$q(t) = \lim_{\substack{\Delta t \to \\ N_0 \to \infty}} \frac{\sum_{i=1}^{\frac{t}{\Delta t}} n_i}{N_0},$$

where t is the time over which the defects are estimated; Δt is the assigned reference interval of time; N_0 is the number of working cycles for the production time; and n is the number of operations in the process.

The other unknown to determine productivity, besides the probability of defects, is the coefficient of machine utilization, η_{ut}. The reliability of the machine is expressed as η, such that

$$\eta = \eta d \prod_{1}^{K} \eta_i \quad ,$$

where η_d is the reliability coefficient of the driving mechanism and control system; K is the number of stations, and η_i is the reliability of the ith station. Then η_{ut} is expressed as,

$$\eta_{ut} = \frac{T_a - T_b - T_c}{T_a} \eta \quad ,$$

where T_a is the time during which the machine is under observation, T_b is administrative downtime, and T_c is set-up and maintenance downtime.

Thus the reliability of the system not only depends on the operational state of the machines, but also on their accuracy. When faulty output is received, the time required to overcome the causes are attributed to set-up time if they occur at the beginning of the operation. Otherwise, that time counts as downtime for preventive maintenance or repairs. Thus as concluded by Mutsenek and Lobzov, "The reliability of each station of the machine and the whole machine, and consequently its productivity, can be improved by reducing the lost time due to defects and also be reducing the probability of their occurrence" (23).

2.4.2 Swanson Demand System

Swanson's system combines the high cycle rate of a synchronous assembly system with the uptime characteristics of a non-synchronous system (29). A synchronous system infers a fixed index type of assembly system employing mechanically controlled machine dynamics. Heginbotham and Rippon define a fixed index type machine as one in

which, " . . . the work carriers index from workstation to workstation, dwelling for a fixed period while automatic workheads perform a sequence of tasks" (15). In a system of this sort with automatic assembly, when a workhead fails, indexing fails to take place until an operator clears the failure.

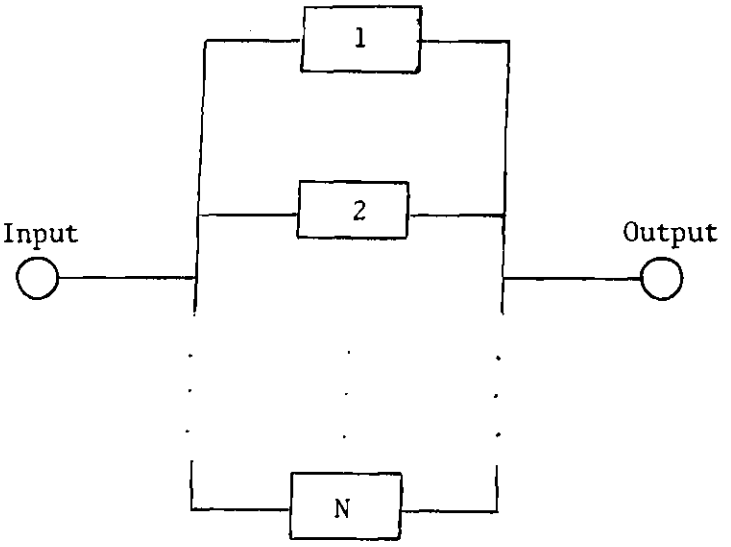
The non-synchronous characteristics of the system are (1) the system contains one or more demand modules and (2) the less efficiently performed operations are done in supply modules. The use of modules in this manner increases system productivity because only faultless items are output from the system. System reliability is increased because the modules can continue to operate when another module is out, and, the failure is quickly isolated.

2.5 Availability and Productivity Models

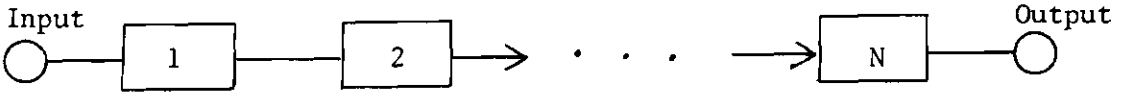
The models in this section include analytical and simulation approaches employed in assessing system availability. All the analytical models assume independence of subsystems. The third model, a computer simulation, investigates system availability and searches for optional buffer size of a given system.

2.5.1 Applebaum's Model

The model developed by Applebaum is capable of predicging system availability of complex systems with mutually independent subsystems. Figure 2 is an example of the type systems which conform to Applebaum's model. The two layouts have units parallel and in series. The term complex refers to systems which are large as a result of the number of subsystems.



A. Subsystems in Parallel Layout



B. Subsystems in Series Layout

Figure 2. Systems Conforming to Applebaum's Model

The constraints on the model are: (1) it is applicable only to steady-state phenomenon, and (2) mutual independence of subsystems is assumed throughout. The assumption of mutual independence stipulates the following: (1) there are no common time-varying environmental factors that influence either failure or repair, and (2) there is a repair team available for each subsystem. With the above stipulations, it is Applebaum's contention that system availability can be obtained without knowing the failure or repair time distributions of the subsystems (1).

In his formulation, the basic subsystem parameters are

$$\bar{u}_i = \frac{1}{\lambda_i} = \text{MUT of the ith subsystem,}$$

$$\bar{d}_i = \frac{1}{\mu_i} = \text{MDT of the ith subsystem}$$

and

$$\bar{a}_i = \frac{\bar{u}_i}{\bar{u}_i + \bar{d}_i} = \text{steady-state availability of the ith subsystem.}$$

The state of the system, $S(t_e)$ is defined as 1 or 0 is the system is "up" or "down," respectively. The subscript e indicates t is sufficiently large and that the system is in steady-state. $S(t_e)$ represents the external or overall state of the system. The internal state of the system, $\underline{S}(t_e)$, is the set of all possible combinations of subsystem states. For example, given a system with five subsystems, two of which must be "up" for the system to be "up," the internal state contains all the combinations of subsystems "up" and "down." $S(t_e)$, however, only contains those combinations which cause the system to change state.

Thus, the combinations containing one subsystem "up" or "down" are not included in $S(t_e)$.

The probability that $S(t_e) = \underline{x}$ is denoted by $A(\underline{x})$, where $\underline{x}$ is any given internal state. Then based on the independence of subsystems,

$$A(\underline{x}) = \prod_{i=1}^{N} (x_i a_i + \hat{x}_i \hat{a}_i),$$

and

$$A_S = \sum_{\underline{x} \in U} A(\underline{x}),$$

where U is the set of all combinations of internal states for which $S(t_e) = 1$.

The system MUT, $\overline{U}_s$, is

$$\overline{U}_s = \frac{1}{\lambda_s} = \frac{\sum_{\underline{x} \in U} A(\underline{x})}{\sum_{x \in \tau} \{(A(\underline{x}) \sum_{i \in \alpha} (x_i \lambda_i + \hat{x}_i u_i)\}},$$

where τ is a subset of U such that $S(t_e)$ has direct access to a "down" state, D. The term α is the critical set of subsystems in τ such that if the state of one element in α changes, then $\underline{s}(t_e)$ goes to D.

2.5.2 Buzacott-MTBF of Repairable Systems

Buzacott takes a very unique approach to finding the MTBF of a repairable system. Figure 3 is an example of the type system for which this approach is applicable. The subsystems do not necessarily conform to any given pattern of parallel or series. As shown, the first flowline has two subsystems in series, while the third flowline has a combination of series and parallel subsystems. The system does not

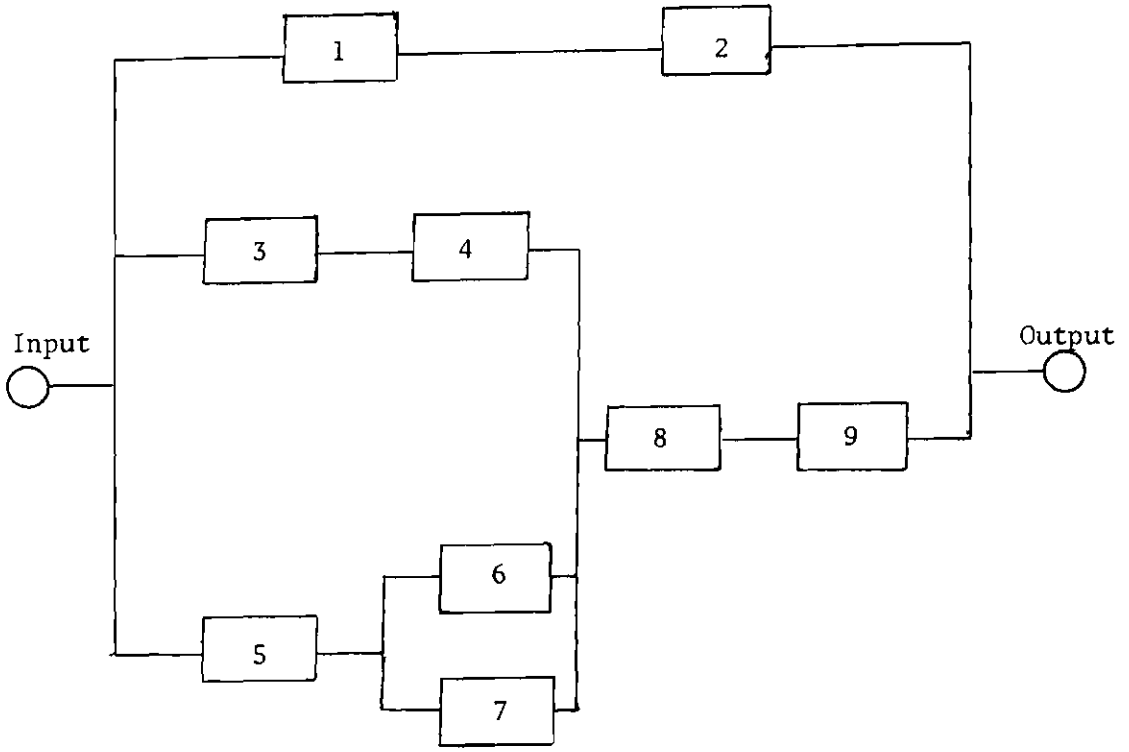


Figure 3. Example of a System Fitting Buzacott's Model

have a buffer capability. It is an improvement over Applebaum's model, since the system can now have any combination of parallel and series subsystems.

These systems are assumed to be made up of independently operating subsystems. The availability of the ith unit is

$$A_I = \frac{U_I}{T_I} ,$$

where U_I is MUT and T_I is MTBF.

The following formulae serve as a base for determining the availability of a system by successive reduction of the block diagram. For X and Y in series

$$A_{xy} = A_x A_y = \frac{U_x}{T_x} \cdot \frac{U_y}{T_y}$$

When X and Y are in parallel

$$A_{xy} = 1 - \bar{A}_x \bar{A}_y = 1 - \frac{D_x}{T_x} \cdot \frac{D_y}{T_y} ,$$

where D_i is MDT of the ith subsystem.

The general procedures for reducing the block diagram are:

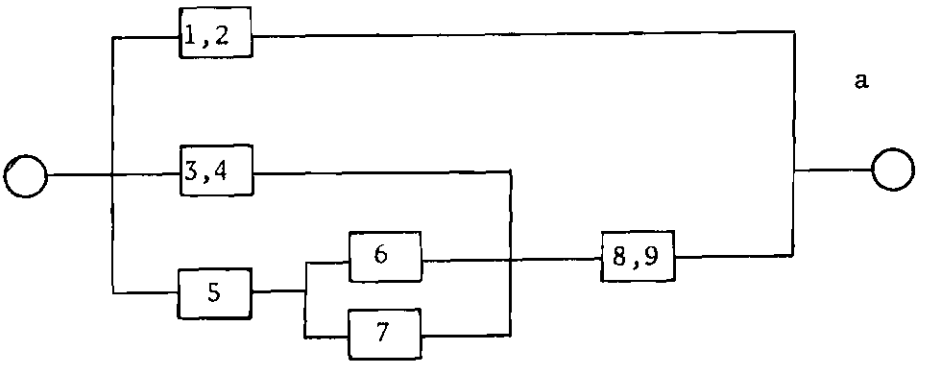
(1) Blocks in series between nodes are reduced to an equivalent block.

(2) Simple parallel or r/n blocks from the resulting diagram of (1) are reduced to an equivalent block.

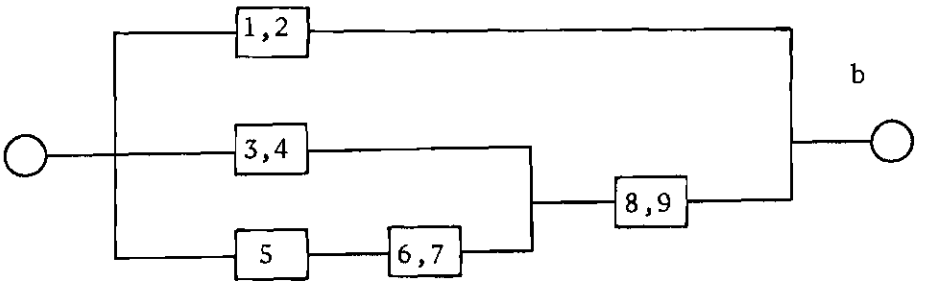
(3) Repeat (1).

(4) Repeat (2).

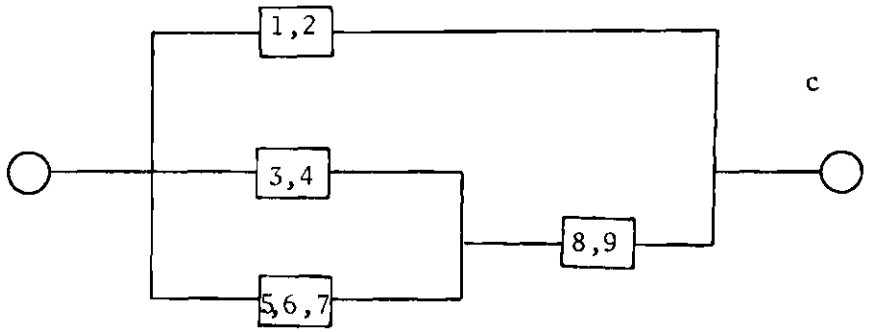
This is continued until the diagram is reduced to one block or no further reductions are possible using the above procedure. Using the system described in Figure 3, the above procedure is used to reduce the system. Step (1) series 1, 2 and 3, 4 and 8, 9 are reduced to (a)



Step (2) parallel 6, 7 are reduced to (b)



Step (3) series 5, 6, 7 are reduced to (c)



Step (4) parallel 34, 567 reduced.

Step (5) series 34567, 89 reduced.

Step (6) parallel 12, 3456789 reduced.

At this point Buzacott refers to Applebaum's model for a closed form solution (7).

2.5.3 Models with Buffers

Buffers are appealing features to a productive system. Buzacott points out that the use of buffers improves the system's productivity because when one subsystem or station fails, other stations continue and the system is able to continue production until the buffer is empty (8).

The term in-process storage or buffers refers to " . . . temporary storage devices suitably located at junctions among the work stations or subsystems to reduce the production time which would otherwise be lost due to machine breakdowns, operate delay, defective stock,

or other possible causes" (20).

Systems in a configuration with subsystems in series and infinite in-process storage do not have idle time caused by the inability of a subsystem to pass a finished item. When there are no buffers, a failure in a forward subsystem will cause a back-up in the system if other subsystems are prepared to pass an item to the downed subsystem. Under these circumstances the former situation--infinite in-process storage--is preferable. However, at the other end of the spectrum is no buffer storage.

2.5.3.1 Barten. Barten also uses the technique of computer simulation for analyzing the in-process storage requirements between any number of successive operations with flexibility regarding physical limitations or differences in operation times. As shown in Figure 4, the subsystems must all be in series and there can only be one sequence of operations. However, that sequence can have n operations and the buffer capacities are computed for each individual buffer.

To measure total production time for item i at station n,

$$T_{n,k} = R_{n,i} + D_{n,i} \qquad 2\text{-}1$$

where R = the independent production rate and D = the delay time spent by that operation awaiting an item.

However,

$$D_{n,i} = T_{n-1,i} - R_{n,i-1} \qquad 2\text{-}2$$

The only way a delay occurs is that the buffer between n-1 and n is

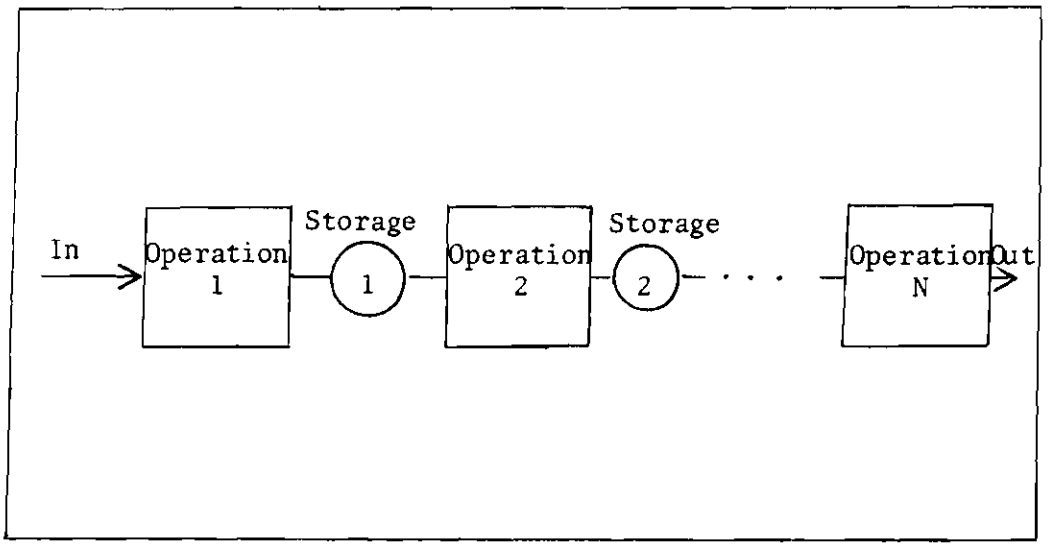


Figure 4. System with In-Process Storage in a Sequential Process

empty. Therefore,

$$D_{n,i} = [T_{n-1,i} - R_{n,i-1}]P(E^{i}_{n-1}) \qquad 2\text{-}3$$

where $P(E^{i}_{n-1})$ is interpreted to be the probability that the last storage (n-1) is empty at time i when operation n is ready to begin work on item i. Continuing the iterative equations,

$$T_{n-1,i} - R_{n-1,i} + D_{n-1,i} \qquad 2\text{-}4$$

$$D_{n,i} = [(R_{n-1,i} + D_{n-1,i}) - R_{n,i-1}]P(E^{i}_{n-1}) \qquad 2\text{-}4a$$

$$D_{n-1,i} = (T_{n-2,i} - R_{n-1,i-1})P(E^{i}_{n-2}) \qquad 2\text{-}5$$

Substituting 2-5 in 2-4,

$$D_{n,i} = [R_{n-1,i} + (T_{n-2,i} - R_{n-1,i-1})P(E^{i}_{n-2}) - R_{n,i-1}]P(E^{i}_{n-1}) \qquad 2\text{-}6$$

Continuing this procedure for n operations in the queue

$$D_{n,i} = [\sum_{j=1}^{n-1} R_{n-j,i} \prod_{L=1}^{j} P(E^{i}_{n-L})][1 - P(E^{i}_{n-(L+1)})]$$

$$- R_{n,i-1} P(E^{i}_{n-1}) \qquad 2\text{-}7$$

The mean output time of the sequence of operation is

$$\overline{T}_n = \overline{R}_n = \frac{1}{m} \sum_{i=1}^{m} D_{n,i}$$

This process establishes what the needs of the system are

without regard to economic considerations (4)-

2.5.3.2 Techniques of Law et al. Law, Baxter, and Massara, in their analysis of a manufacturing system with in-process buffers, take the simplistic approach to defining system availability,

$$R = \frac{\text{Average Yield (PCS/hr)}}{\text{Predesigned Rate (PCS/hr)}} .$$

As shown in Figure 5, the system under analysis has n+1 subsystems. The first n subsystems are in a parallel configuration. Each subsystem has a buffer before its output reaches the assembly line. The assembly line is considered the n+1 subsystem of the system. The measure of availability of a piece of equipment is

$$r = 1 - \frac{(\Sigma d)_e}{T}$$

where $(\Sigma d)_e$ = total downtime due to the equipment, T = total manned production time, and availability of non-equipment factors is defined as

$$r = 1 - \frac{(\Sigma d)_{ne}}{T}$$

where $(\Sigma d)_{ne}$ = total downtime due to non-equipment factors.
Thus, system availability, without buffers and assuming independent series arrangement, is identified as R_0, with

$$R_0 = r_1 r_2 r_3 \ldots r_n,$$

where r_i represents the availability of the ith subsystem (20).

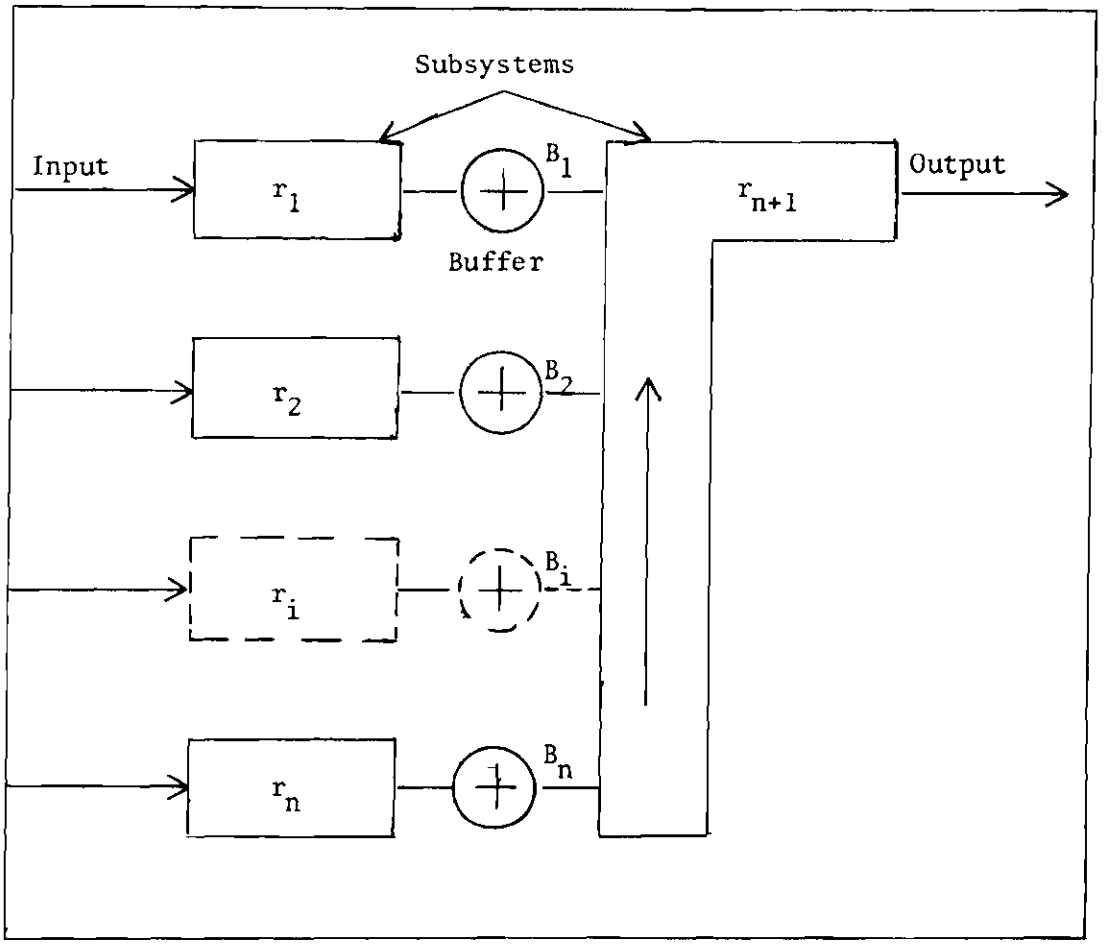


Figure 5. Physical Layout of a Manufacturing System

Although the analytical approach assumes a series arrangement, the subsystems in the schematic are parallel.

Availability, when buffers are inexhaustible, becomes

$$R = \text{Min}[r_1, r_2, \ldots, r_n].$$

Since the optimal buffer capacity lies between these two conditions, availability of a two-buffer system can be expressed as

$$R = \tfrac{1}{2}\{[(1-B_1) + (1-B_2)](r_1 r_2 r_3) + [B_1 + B_2](rm)\}$$

where B_1, B_2 = effectiveness of buffers 1 and 2, respectively,

$$0 \leq B_1 \, , \, B_2 \leq 1$$

and

$$r_m = \text{Min}[r_1, r_2 r_3]$$

The generalization for the n - buffer system yields

$$R = 1/n\left[\left(n - \sum_{i=1}^{n} B_i\right)\left(\prod_{i=1}^{n+1} r_i\right) + \left(\sum_{i=1}^{n} B_i\right)(r_m)\right]$$

where

$$r_m = \text{Min}[r_1, r_2, \ldots, r_{n+1}]$$

$$n \geq 1, \quad 0 \leq B_i \leq 1, \quad i = 1, 2, \ldots, n \quad .$$

Then

$$B_i = \frac{R_{(i)} - \prod_{j=1}^{n+1} r_j}{r_{m(i)} - \prod_{j=1}^{n+1} r_j}$$

where

$$r_{m(i)} = \text{Min}[r_i, \prod_{\substack{j=1 \\ j \neq i}}^{n+1} r_j] \ .$$

To measure the percentage of improvement in system availability (ΔR),

$$\Delta R = \frac{R - \prod_{i=1}^{n+1} r_i}{\prod_{i=1}^{n+1} r_i} \times 100$$

The potential system availability improvement is greatest, then, when there is a large number of subsystems or the availability of each subsystem is low (20).

2.6 Availability of a Complex System

Although some of the models presented in the earlier sections are labeled as complex, they fail to investigate combinations of important features simultaneously. For example, those models with buffers do not include load sharing nor periodic maintenance and vice versa.

An example of a system with load sharing is now expanded to include the ability to change the mean time to repair during a

production run. The closed form solution for this system with only that one addition will be seen to be very cumbersome. It should be noted that the closed form solution does not include other necessary system operating features such as buffers, equipment sharing, and periodic maintenance.

2.6.1 Description of a Subsystem

The subsystem described in this section imposes two conditions causing dependency. Each unit in the subsystem is identical. Identical units mean: (1) the production rate is the same for each unit, where the variance is very small, thus we consider production time to be a constant; (2) the time to failure density is exponential and the same for each unit; and (3) the time to repair density is exponential and identical for each unit.

The units operate on a load sharing basis. Given a subsystem that has a production rate L, then the load or the production rate for each unit, when all units are operative, is L/n, where n is the number of units in the subsystem. If one or more of the units fails, the load of the remaining units is increased to maintain a constant production rate. However, this increase in production rate on a unit causes its failure rate to increase. This situation triggers dependence of failure rate.

At the subsystem level in this model there is one repair team. Because of the speciality of repair, the team only works on one unit at a time. The time to repair is dependent on the number of machines that are in a failed state, i.e., if one machine is non-operational it is repaired at some mean rate μ_0; however, if two machines are non-

operational, the first unit is repaired at a mean rate μ_1 where $\mu_1 < \mu_0$. That same procedure holds for cases up to three units in for repair, where $\mu_0 > \mu_1 > \mu_2 = \mu_i$, $\forall\, i > 2$. As long as at least one machine is operational, then the subsystem is operational.

2.6.2 A Closed Form Solution

A closed form solution yields the percentage of time the subsystem is operational. The first step is to formulate the transition matrix for the subsystem. For this illustration the subsystem has three units.

Let State 0 = all units operational

State 1 = 1 unit non-operational

State 2 = 2 units non-operational

State 3 = all units non-operational

Thus State 3 represents the condition of the subsystem being down or non-productive.

	0	1	2	3
0	$1-3\lambda_0\Delta t$	$3\lambda_0\Delta t$	0	0
1	$\mu_0\Delta t$	$1-2\lambda_1\Delta t-\mu_0\Delta t$	$2\lambda_1\Delta t$	0
2	0	$\mu_1\Delta t$	$1-\lambda_2\Delta t-\mu_1\Delta t$	$\lambda_2\Delta t$
3	0	0	$\mu_2\Delta t$	$1-\mu_2\Delta t$

Solution:

Step 1.

P_0	P_1	P_2	P_3
1	1	1	1
$-3\lambda_0$	μ_0		
$3\lambda_0$	$-2\lambda_1 - \mu_0$	μ_1	
	$2\lambda_1$	$-\mu_1 - \lambda_2$	μ_2
		λ_2	$-\mu_2$

Step 2.

P_0	P_1	P_2	P_3
1	1	1	1
$-3\lambda_0$	μ_0		
	$-2\lambda_1$	μ_1	
		$-\lambda_2$	μ_2

Step 3.

$$\mu_0 P_1 = 3\lambda_0 P_0$$

$$\mu_1 P_2 = 2\lambda_1 P_1$$

$$\mu_2 P_3 = \lambda_2 P_2$$

Since, $$P_3 = P_3$$

then,

$$P_2 = \frac{\mu_2}{\lambda_2} P_3, \quad P_1 = \frac{\mu_1}{2\lambda_1}\left(\frac{\mu_2}{\lambda_2}\right) P_3, \quad P_0 = \frac{\mu_0}{3\lambda_0}\left(\frac{\mu_2}{\lambda_2}\right)\left(\frac{\mu_1}{2\lambda_1}\right) P_3 \quad .$$

But,

$$P_0 + P_1 + P_2 + P_3 = 1,$$

therefore,

$$P_3 = \frac{1}{1 + \frac{\mu_2}{\lambda_2} + \frac{\mu_1\mu_2}{2\lambda_1\lambda_2} + \frac{\mu_0\mu_1\mu_2}{6\lambda_0\lambda_1\lambda_2}} \quad .$$

P_3 represents the percentage of the time the system will be down, which represents non-availability, $\hat{A}$. Thus, availability, A, is represented as,

$$A = 1-\hat{A} = 1-P_3 = \frac{\frac{\mu_2}{\lambda_2} + \frac{\mu_1\mu_2}{2\lambda_1\lambda_2} + \frac{\mu_0\mu_1\mu_2}{6\lambda_0\lambda_1\lambda_2}}{1 + \frac{\mu_2}{\lambda_2} + \frac{\mu_1\mu_2}{2\lambda_1\lambda_2} + \frac{\mu_0\mu_1\mu_2}{6\lambda_0\lambda_1\lambda_2}} \quad .$$

The above methodology is useful if the system is relatively simple and the interest in availability is limited to steady-state phenomenon. To overcome the limitation of being restricted to steady-state conditions, differential equations can be used to find the availability at any given time t. The transition matrix is basically the same

	0	1	2	3
0	$1-3\lambda_0\Delta t$	$3\lambda_0\Delta t$	0	0
1	$\mu_0\Delta t$	$1-2\lambda_1\Delta t-\mu_0\Delta t$	$2\lambda_1\Delta t$	0
2	0	$\mu_1\Delta t$	$1-\lambda_2\Delta t-\mu_1\Delta t$	$\lambda_2\Delta t$
3	0	0	$\mu_2\Delta t$	$1-\mu_2\Delta t$

Then the differential equations are

$$P_0'(t) = -3\lambda_0 P_0(t) + \mu_0 P_1(t)$$

$$P_1'(t) = 3\lambda_0 P_0(t) - (2\lambda_1 + \mu_0)P_1(t) + \mu_1 P_2(t)$$

$$P_2'(t) = 2\ {}_1P_1(t) - (\lambda_2 + \mu_1)P_2(t) + \mu_2 P_3(t)$$

where $P_2(t) = 1 - P_0(t) - P_1(t) - P_2(t)$.

Using Laplace Transforms, we have

$$L\ P_0(t) = P_0(s),$$

$$L\ P_1(t) = P_1(s), \text{ and}$$

$$L\ P_2(t) = P_2(s).$$

The initial conditions for the system are

$$P_0(0) = 1,\ P_1(0) = P_2(0) = P_3(0) = 0.$$

Solution: let $\lambda_0 = \lambda$, $\lambda_1 = k\lambda$, $\lambda_2 = \ell\lambda$, and

let $\mu_0 = \mu$, $\mu_1 = \sigma\mu$, and $\mu_2 = h\mu$.

Then

$$P_0(s) = \frac{1 + \mu P_1(s)}{s + 3\lambda}$$

$$P_2(s) = \frac{(s^2 + 3\lambda s + 2k\lambda s + \mu s)\ P_1(s)}{g\mu}\ , \text{ and}$$

$$P_1(s) = 3gh\lambda\mu^2/(s^4+6\lambda s^3+2k\lambda s^3+\mu s^3+9\lambda s^2+6k\lambda s^2+3\lambda\mu s^2$$

$$+ 4gk\lambda\mu s^2-6gk\lambda^2\mu s+3\ell\lambda^2\mu s^3+g\mu s^4+6g\lambda\mu s^3+2gk\lambda\mu s^3$$

$$+g\mu^2 s^3+9g\lambda\mu s^2+3g\lambda\mu^2 s^2+gh\mu^3 s+gh\mu^2 s^2+3gh\ell\mu^2 s$$

$$+h\mu s^4+6h\lambda\mu s^3+2hk\lambda\mu s^3+h\mu^2 s^3+9h\lambda\mu s^2+6hk\lambda\mu s^2$$

$$+3h\lambda\mu^2 s^2)$$

At this point it becomes obvious that finding a closed form solution to this problem is not easy. In addition this formulation is for a subsystem with only three units. When the number of units increases the complexity of the problem multiplies.

This serves as one illustration of a system for which availability is not easily computed. Other characteristics of a system which add to the complexity of a system are (1) equipment sharing, (2) preventive maintenance and regular or "fix as it fails" repair facilities, and (3) buffers. Equipment sharing by subsystems will cause a back up on each subsystem using that piece of equipment whenever it fails and is ready to be used by the awaiting subsystems. When the system utilizes both "fix as it fails" repair techniques and preventive or periodic maintenance, there will be times when the subsystem would normally be available, but it is out for the length of time it takes to perform the periodic maintenance.

2.6.3 Complex Systems--A Need for Simulation

The subsystem described in the previous two sections illustrates a need for the use of some method or technique which will include the conditions causing subsystems to be dependent. In addition to load sharing, there are many other features which contribute to the complexity of a system. Among these are the inclusion of buffers and preloading the system. These and other features of a complex system become variables in the model of this study. A complete description of all variables is in Section 3.7.

Searches for a technique that will accommodate the dependent features lead to computer simulation. Although simulation does not yield a closed form solution, it will provide a realistic representation of the model. Thus providing insight to the expected performance of a system under a given set of circumstances.

2.7 Summary

All the models reviewed in this chapter assume independence of subsystems. Applebaum and Buzacott, however, acknowledge that this assumption is primarily for mathematical simplicity and takes away from the accuracy with which the system is represented.

The material included in this chapter generally serves as background for the thesis. Mutsenek and Lobzov researched the problem of improving productivity by improving reliability of automatic assembly machines (23). However, their study is restricted to increasing machine reliability. The system developed by Swanson contributes to the idea of using the module approach to study large complex systems

(29). Although Swanson has his system set up using nodules, there is no way to efficiently expand his system or modify it. Applebaum's model can evaluate systems with independent subsystems during steady-state. Buzacott's approach can accommodate non-redundant subsystems in addition to the system described by Applebaum. The models of Barten and Law et al. add the feature of buffers to the model used by Buzacott.

None of the above mentioned works include: (1) equipment sharing, (2) preloading, (3) periodic and "fail and fix" maintenance policies simultaneously. Neither is a relationship between reliability and productivity addressed in the models of Applebaum, Buzacott, Barten, or Law et al.

A subsystem is extracted from a load sharing system which illustrates how the magnitude of a closed form solution becomes very cumbersome. By merely changing the mean time to repair based on the number of units in for repair, the problem becomes unreasonable.

Consider adding to that problem other variables, such as: (1) equipment sharing, (2) periodic maintenance, and (3) buffers, and the problem grows even more. The magnitude of the problem is increased when (1) the additional variables are included, and (2) the subsystem is added to the system. Simulation is used to analyze the system which contains subsystems of the above described magnitude.

CHAPTER III

BUILDING A GPSS MODEL FOR A COMPLEX LINE MANUFACTURING SYSTEM

After determining a need for simulation, a scan of computer languages resulted in a choice of General Processing System Simulation, GPSS. The use of GPSS minimizes the amount of programming effort and is appropriate for the type problem that will be modeled.

Section 3.1 describes the classification of complex systems, and how the modular approach is used to study complex systems. A GPSS program, listed in Appendix A, can conform to numerous module configurations. The maximum module sizes are delineated in Section 3.2. A description and operating characteristics of the example module, which is used throughout the remainder of this study, are presented in Section 3.3.

Section 3.4 contains a scenario for the GPSS program. The scenario is subdivided into the different functioning features of the system. As with any simulation program, the program must be verified and validated. The procedures and results of verification and validation appear in Sections 3.5 and 3.6, respectively.

In Section 3.7 a complete description of the variables which constrain the module configuration are given. Section 3.8 details the model responses.

3.1 Complex Systems' Synthesis

Systems can be classified as complex as a result of their size or the interdependencies between elements. In either case, the complex system can be studied by subdividing the system and studying its parts using a modular approach. The degree of difficulty is lessened when complexity results from size alone. A complex system may have as few as one subsystem or as many as desired by the user. There is no established dividing line for simple and complex systems. However, reference to a complex system in this study does not refer to any system less complex than those described by the models in the literature.

3.1.1 Module Complexity

When the size of the system is the determining factor, the number of modules increases, but the module is less complex to model and verify. The module then closely approximates a simple system.

However, when complexity results from the integration of numerous operations, the module itself is likely to be complex. It is more difficult to build a modular model. For example, if several operations are performed at the same workstation and the system consists of multiple workstations, the module must insure an accurate portrayal of the integrated operations which take place at a given workstation. On the other hand, the number of modules can be increased when the operations are simple with a large number of workstations. A modular model of this type is easier to build and verify.

3.1.2 A Modular Approach

The modular approach to the problem dictates the need to design a system which is capable of portraying a simple yet sophisticated

system, Figure 6. The module in Figure 6 is a hypothetical example of how a module can be designed such that it represents a small system. The system in the figure has four subsystems with an in-process buffer for each subsystem. Although this system is labeled simple, it is complex when compared with others in the literature. However, it does not approach the size of a very large and intricate operating module. The flow enters the first subsystems, after processing the items flow to the buffer, and from there into the last subsystems for processing. The two flowlines are parallel; however, items from the last subsystems of each flowline must link-up before exiting the system.

As the complexity of the system increases as a result of its size, the small system in Figure 4 becomes a module of the larger system. Figure 7 is also a hypothetical example which illustrates how a larger system is subdivided into modules to simplify the system's analyses. This figure is not indicative of the maximum size of a system or module under study in this research. Each module is analyzed separately; however, the two modules are interfaced to get the output from the second module. Thus the activities of the entire system may be obtained after correlating the activities of the modules with regard to time.

For example, let the module represent a sequence of operations performed before a subsequent sequence is performed. Then the first sequence is observed over a given time interval (0,t), maintaining a record of times the item is ready to be transferred to the next sequence. At time t, the clock is reset to zero and the outputs of the first module serve as inputs to the second based on the times they

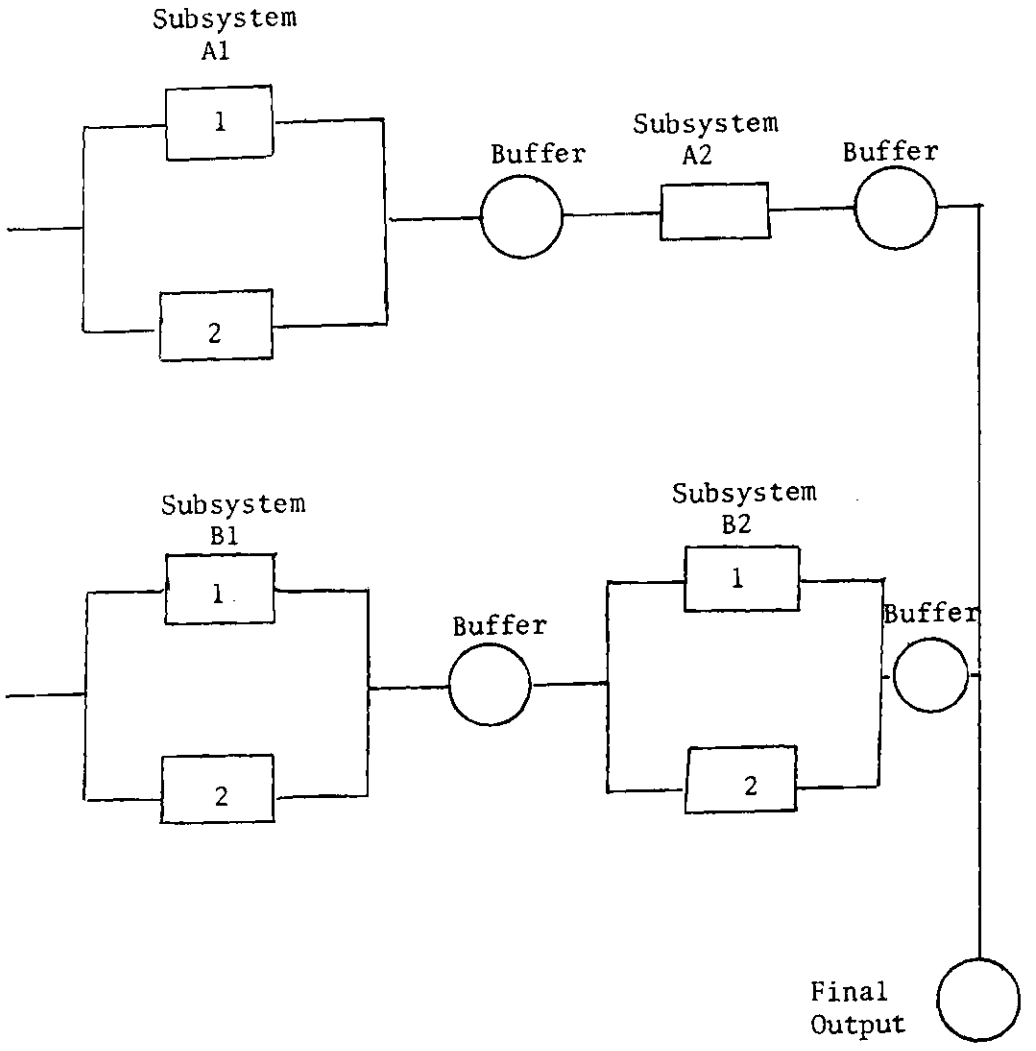


Figure 6. Module Representing a Small System

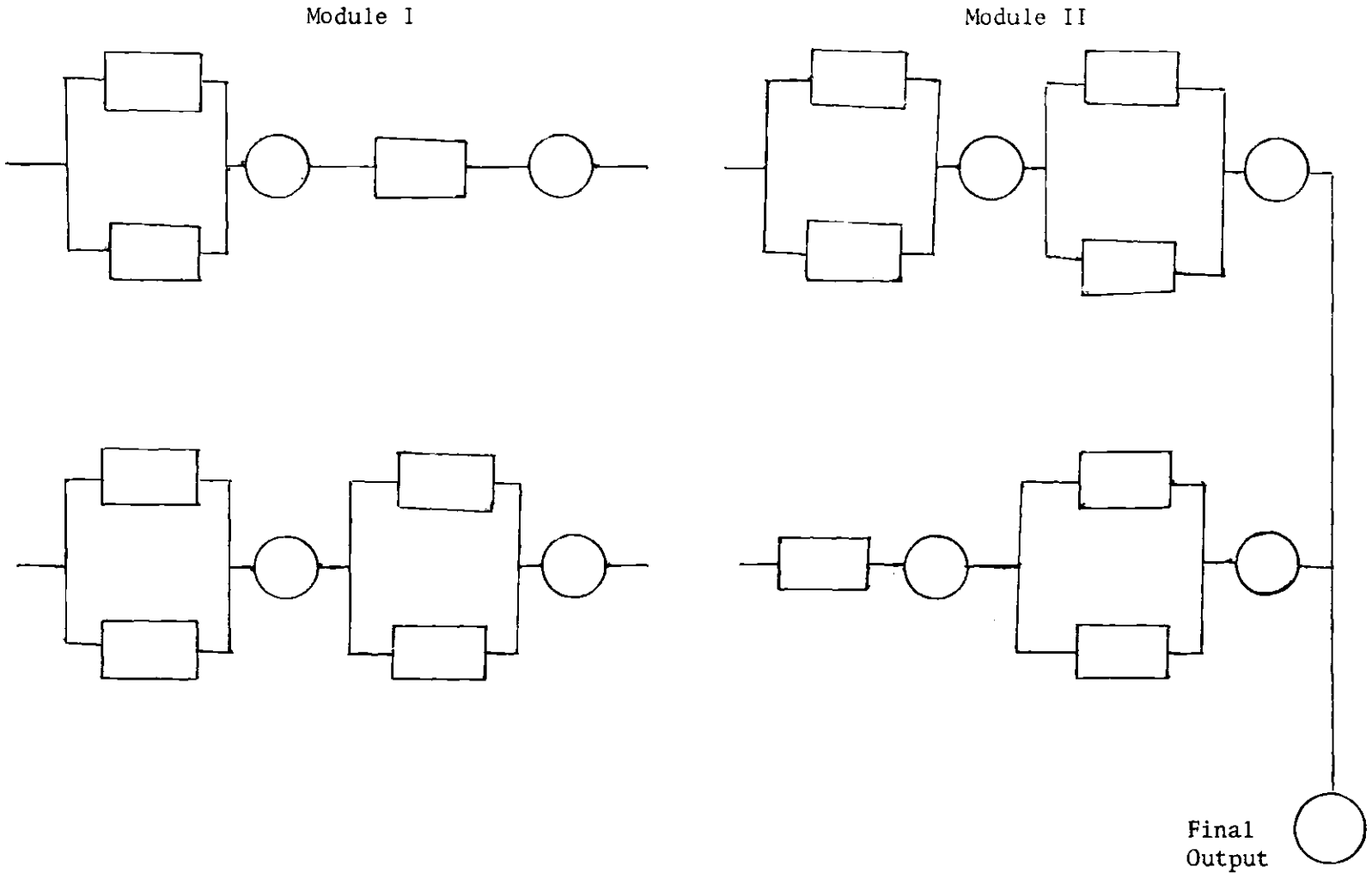


Figure 7. Module as Part of a Larger System

were output from the first. Therefore, the second module is also observed over the time interval (0,t). This process continues until each portion of the system has been included. The output from the last module becomes the output for the system.

3.2 Module Size Capability

As stated in Section 3.1, system complexity can result from the interdependencies between elements. When this situation is combined with the size factor forming a large complex system, the module must be large enough to represent the activities as they exist within the real system.

A GPSS program was written to accommodate large modules. The program listing, symbology, and notes to the users are in Appendices A, B, and C, respectively. The program is designed to model any module which has any combination of 65 units and buffers, or less. Appendix C contains the details on how to change configurations.

Figure 8 shows one limiting configuration of a module which can be accommodated using the modular designed program. The maximum number of sequential operations is 32 with 32 buffers. The program is not restricted to having the maximum number of units. The number of workstations is labeled i, where $1 \leq i \leq 32$. After each workstation is an in-process buffer. If the buffer capacity is zero, the buffer acts as a transfer point.

Figure 9 illustrates a second extreme configuration for the module at a limit, which can be accommodated by this study. In this layout the 64 subsystems are in a parallel configuration with one common buffer for the subsystems.

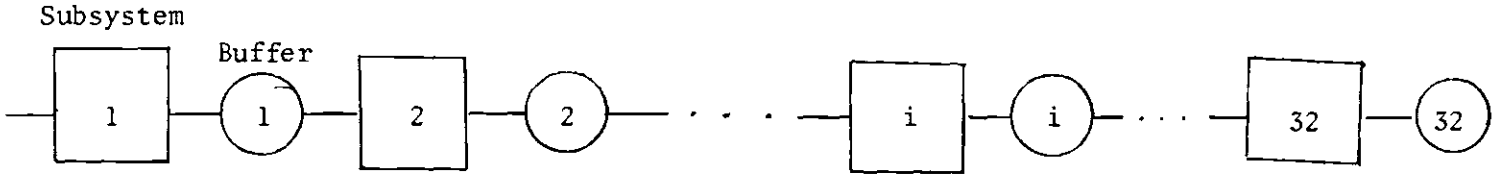


Figure 8. Limiting Configuration of Module for Sequential Operations

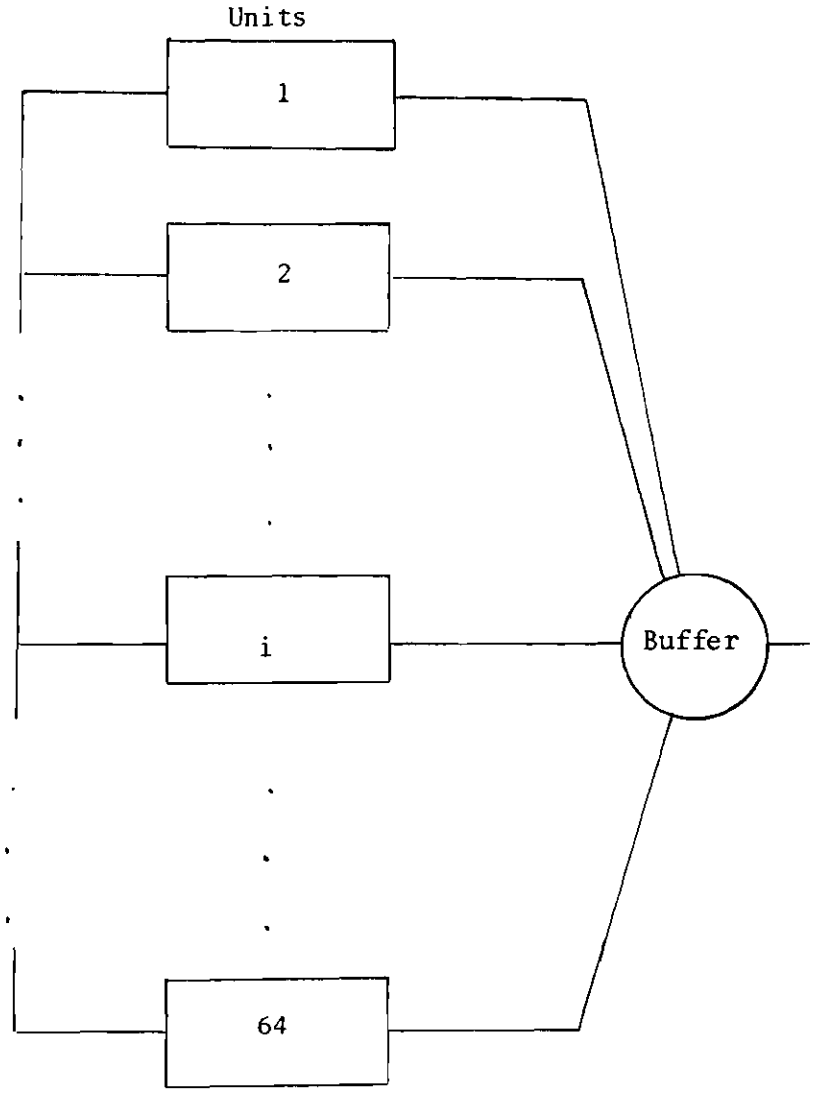


Figure 9. Parallel Configuration Within a Module

The third module configuration, Figure 10, is a combination of the previous two. This time the module has three flowlines with four subsystems in each flowline, again the units plus buffers equal the 65 limit. The subsystem consists of multiple units which are parallel and may or may not be redundant. In this figure, like the previous two, the total number of units and buffers cannot exceed 65. But any configuration with the sum of units and buffers less than or equal to 65 can be accommodated by the program.

3.3 Design of An Example Module

An example of a module is designed to illustrate the modular approach to analyzing a system. The module chosen for this illustration is not of the limiting size as described in the previous section. Rather, it is a simpler example with the capability of describing the activities of a system. The example module was chosen because it is comparable to models discussed in the literature, yet large enough to experience features not simultaneously considered in the literature.

The module described in this section will be used throughout the remainder of this study. This example neither illustrates the minimum nor maximum size for a module which can be handled by the GPSS program in Appendix A.

3.3.1 Description of the Example Module

The module consists of two flowlines, each having two subsystems. The configuration of the subsystems will be discussed in Section 3.2.2. The module layout is illustrated in Figure 11. The production time, time to failure, and time to repair may be the same or different for

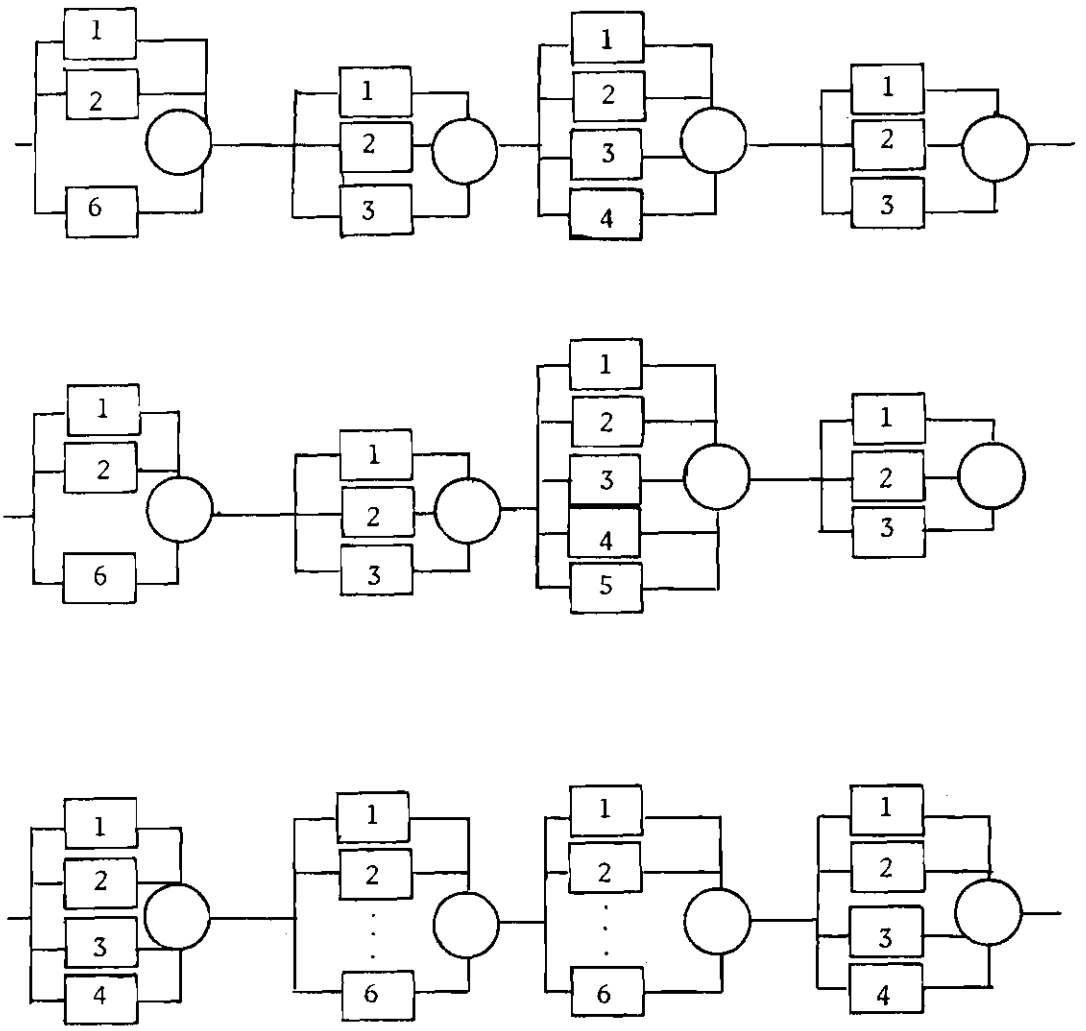


Figure 10. Combination Module Configuration

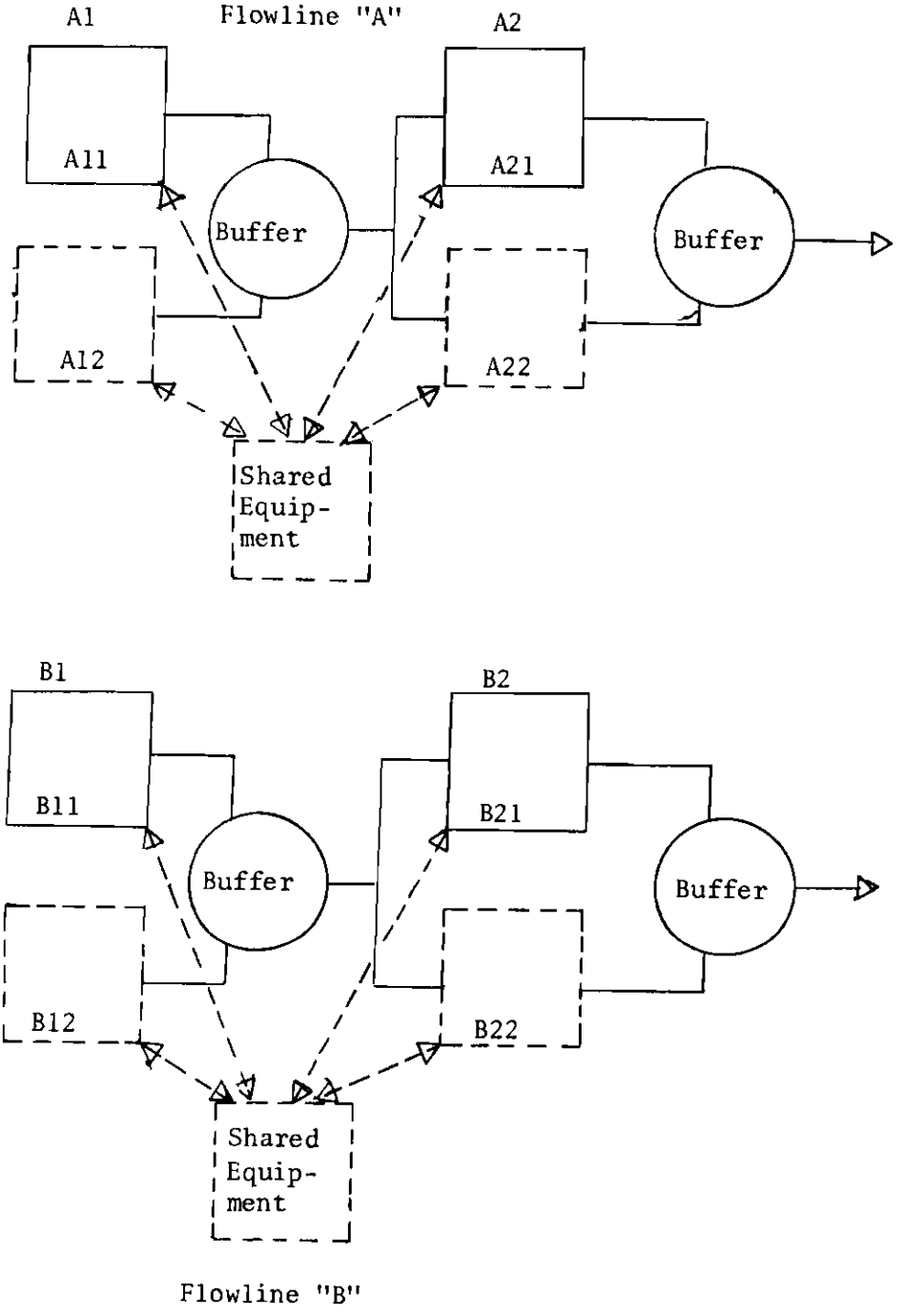


Figure 11. Module Diagram

each unit in any given subsystem. Time to failure and time to repair are assumed to be exponentially distributed, however, they may conform to other distributions as well. The two series are referred to as Flowlines "A" and "B". Therefore, the two subsystems in Flowline "A" are called A1 and A2 for the first and second, respectively. The same nomenclature applies to the subsystems in Flowline "B".

Each flowline has at its disposal one item of equipment which may be shared by the units of the two subsystems in that flowline. That piece of equipment may be utilized at any point during the production cycle. Although the arrows in Figure 11 indicate that each unit in a flowline is utilizing the shared equipment, the module's flexibility allow any combination of units to utilize the shared equipment.

The option of having periodic maintenance performed on the equipment of each subsystem is also available. The maintenance can be performed by the same crew that does the regular repair work or a special team may be brought in to perform periodic maintenance. The option is also available to operate the system without a capability to perform maintenance on a "fix as it fails" basis. To operate the system without periodic maintenance, the interval for periodic repair is set in excess of the production run time. Still another option is to have a repair team perform maintenance on more than one subsystem.

At the end of each subsystem is a buffer, into which items not immediately needed by the succeeding subsystem may be stored. The capacity of this buffer can be set by the user anywhere between 1 and ∞. When the capacity is 1, the buffer is merely acting as a transfer

point for items from units of one subsystem to the units of the next subsystem. The capacity of the buffers may differ within the module. For example, the buffer capacities for the four subsystems may be set as follows; subsystem A1-10, subsystem A2-0, subsystem B1-40, and subsystem B2-∞.

3.3.2 Subsystem Configuration

The subsystems, labeled A1, A2, etc., in Figure 11, may have one or more units. The units, as shown in Figure 12, are in a parallel configuration. It is not mandatory that the units have the same failure or repair rates.

The number of units in this model, m, is dependent upon the expected production time. Whenever the production rate for one unit is one-half the desired rate for the subsystem, there are two units in the subsystem. However, the model will conform to other configurations not dependent on the production time.

For the model, the number of units in each subsystem is equal to the number required for the entire process to conform to line balance. The term line balance here means the number of items produced by a given series of subsystems by time t can be equalled by any other series of subsystems, provided there are no failures in the interval (0,5).

Thus, the layout of subsystems is a control variable in the study. In addition to varying the layout of a subsystem, the number of subsystems within a flowline may also be varied. By varying the number of units and subsystems, the module is general enough to portray a wider variety of system layouts.

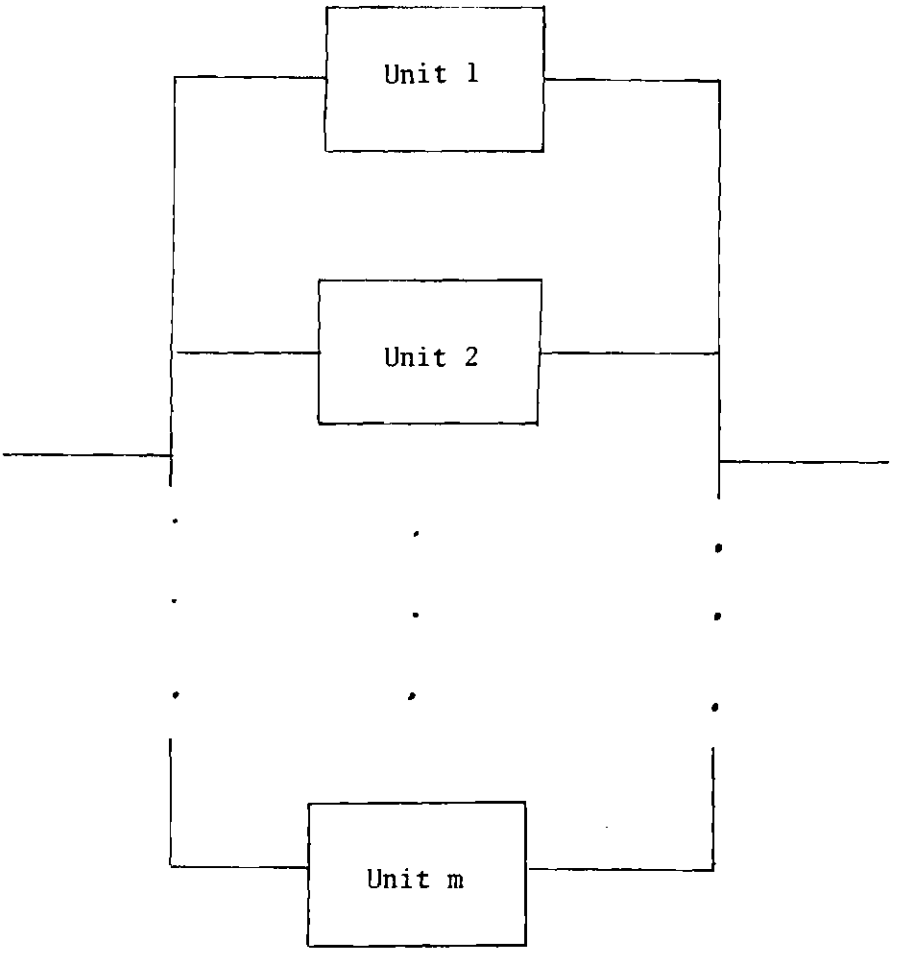


Figure 12. Subsystem Layout

3.3.3 Module Operating Characteristics

At time zero, each unit within the module is operational. The units in the first subsystem of each flowline begin production at start-up time. The production supply of raw materials is infinite, therefore, there is no waiting time for inputs associated with production for the first units in any flowline.

As each unit in the first subsystem of each flowline completes its production operation, the end product from that unit attempts to enter the buffer. Entry into the buffer is only denied if it is already filled to capacity. Whenever entry is denied, the item remains in the unit until entry into the buffer is allowed. The period of time that the item is awaiting entry into the buffer is idle time, because the unit cannot start on another item until the current item has reached the buffer. Idle time, resulting from denial of entry into the buffer, is diminished to zero when buffer capacity is infinite. All subsystems other than the first of each flowline must remain idle until an item arrives from the buffer.

After an item exits the last subsystem of a flowline and enters the buffer, it must await an item from the other flowline before it is permitted to leave the module. If this module is the last module, then this facilitates a link-up for flowlines. If this is not the last module the exit time is recorded and used as an entry time for the next module. This link-up feature is not a consideration in any of the analytical models. The computer simulation models fail to discuss any effects on overall production.

At time one the random variable generator for time to failure

is activated. Therefore, at time zero the probability that the system is operational is certainty. After each repair, the status of like units is checked and at that point the time to next failure is assigned. If an item is being produced when a failure occurs, production is interrupted until the unit is repaired. Upon completion of repair, production is resumed from the point of interruption.

3.4 The GPSS Program--A Scenario

This scenario is presenting the program, Appendix A, as it is read, rather than as following the system's operation on a time incremented basis. The first operation is setting up SAVEX cells. A SAVEX cell is a storage block for numeric values. These locations are used for initializing the variable values.

3.4.1 Unit Production Routine

The GENERATE statements at lines 238 and 240 serve a unique purpose. Their missions are to replace items in the buffers that attempt to go into units in A2 or B2 that are inactive during that run. If these units are active the item remains in the GENERATE block. A COMPARE statement follows the GENERATE block to check the unit's condition via a Boolean variable. The conditions stated in the Boolean variable which must be satisfied are (1) the facility in the inactive unit has attempted to accept the item, and (2) the unit is not active during this run.

The supply of raw material is assumed infinite. As a result, the raw material is generated to the initial units of a flowline immediately after it becomes inactive. Upon arrival of the raw material the facility is seized, thus denying its use by another item

until: (1) production is complete on that item, and (2) the item is accepted into another area. To keep records of the time length of production of items, as soon as the facility is seized the time is recorded. At this time a search is performed to find out if other units of the same subsystem are in a failed state. This information is needed if the system is load sharing. If the other units are not in a failed state, the transactions proceed with production at the normal rate. If other units are non-operational, a check is made to ascertain if the system is load sharing. At this point if the system is load sharing and other units are down, the production rate is increased by a factor of 1.5.

The length of time required for production is an assigned constant, the value based on the number of units in the subsystem, Figure 13. Under the concept of line balance, if the production rate is one per 100 minutes, then the subsystem has one unit if the production time is 100 minutes or two units if production time is 200 minutes, etc.

Upon completion of a portion of the required production time, the item goes to a facility which is shared by other units of the same flowline. While in the shared facility, this item continues to have control of the production unit. If there is no equipment being shared, then the facility is by-passed by assigning zero production time for that station.

Before the unit of the subsystem is released, a check is made to ascertain if there is an idle unit in the next subsystem; if not, is there a buffer and if there is--is it full? The item will retain

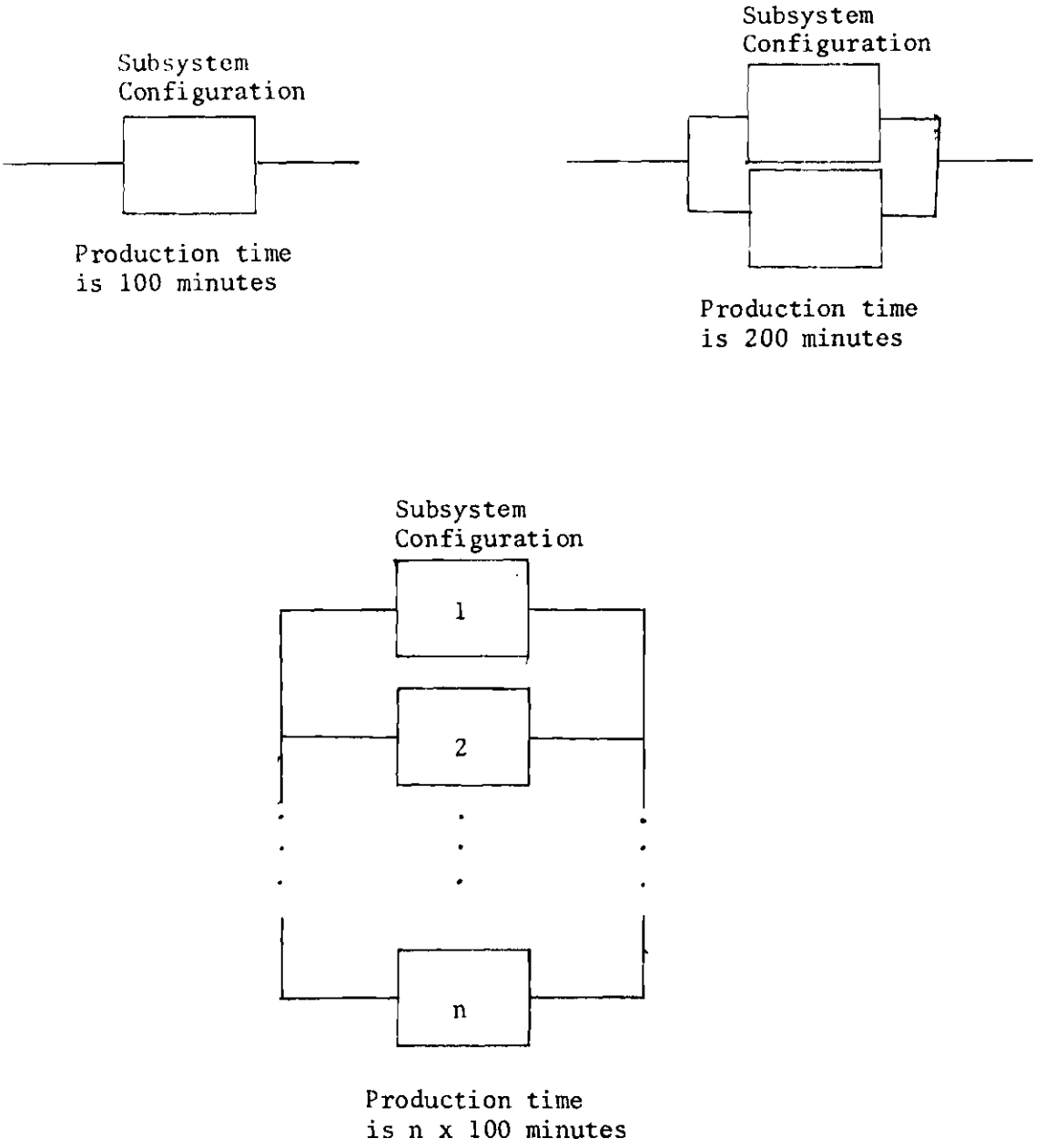


Figure 13. Line Balance Concept

control of the unit it is in if there is not an idle unit and (1) there is no buffer, or (2) the buffer is full.

3.4.2 Dummy Storage

In the language of GPSS on the UNIVAC 1108 in its configuration at Georgia Institute of Technology, the system overloads when there are too many active transactions in the system. For this program, overloading occurs when the sum of transactions in queues exceeds 65. To prevent the overloading condition from occurring, the first of two steps is taken at this point. As an item qualifies to release a unit after production, a COMPARE statement is entered. The question is asked, "Does the buffer currently have less than 5 items waiting to enter the next subsystem?" If the answer is yes, the item goes directly to the buffer; if no, the item bypasses the buffer, adds one to the dummy storage area, and exits the system. This dummy storage is considered a part of the buffer because whenever a check is made to see if the buffer is at capacity the queue and the dummy storage are summed to get the number in the buffer. The SAVEX H--is the name of the dummy storage. To know the total number of items that ever enter the dummy storage, the SAVEX PLUS--is used. (See Appendix B for program symbology.)

With the above procedure, the buffer capacity can be set at one. Then the module can have a combination of 72 units and buffers. Thus enabling the program to accommodate modules of the magnitudes in Figures 8, 9, and 10. In addition, when simulating a specific system, all the extraneous Boolean variables are deleted from the program. As more Boolean variables are deleted, the feasible number of active

transactions is increased. For example, the program in its test configuration has the Boolean variables needed to determine the system availability for all possible system and subsystem layouts. With a specific system, some of the layouts become infeasible.

To retrieve the items that enter the dummy storage, one transaction is generated whenever the number in the dummy storage SAVEX is greater than zero, the number of items in the queue is less than three, and the capacity of the buffer is greater than three. As soon as these conditions are met, items are fed into the queue. The number in the dummy is decreased by one, and one is added to the SAVEX, MINUS--, which records the number of items that exit the dummy.

The second step taken to prevent a system from overloading is to slow down buffer-fill when the system is preloaded and buffer capacity is high. This is accomplished by placing a time increment on filling the buffer. The time increment must be smaller than the production time for the succeeding subsystem.

The above procedure is the same for units A11 and B11. The only difference for A12 and B12 is that they have a COMPARE immediately after the facility is seized. That COMPARE asks the question, "Are there two or more units in that subsystem?" If the answer is yes, it begins to follow the above described routine; if no, the transaction goes no further, thus not allowing any activity to take place in that unit.

3.4.3 Record of Production Time

To complete the statistics of the length of time it takes the facility to complete production on a given item, the clock time is recorded just before the facility is released. A release of the

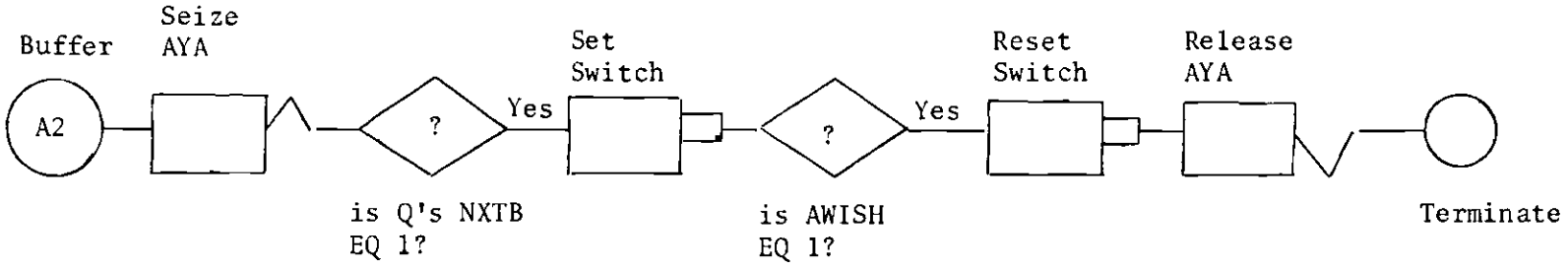
facility means either there is an idle unit or there is space in the buffer. In either case, when the item departs the unit from the first subsystem in a flowline, it will go to the area designated buffer. When the buffer capacity is zero, it passes immediately through that area into the next subsystem. This action in the program operates by having the transaction go to the QUEUE, A1A2 or B1B2, from subsystems A1 and B1, respectively. Following the QUEUE is a GO TO statement for the items to be directed to units in the next subsystem as they become idle.

3.4.4 Exit Routine

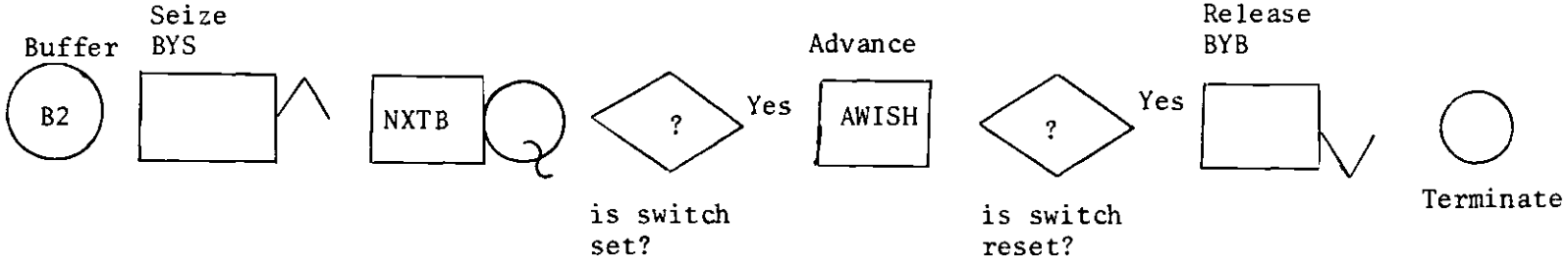
Once in the second subsystem of the flowline, the same procedure is followed as that in the previous subsystem until the item prepares to exit the unit. When the item enters the buffer following the last unit, it goes through a special routine. This is a link-up routine, Figure 14. As an item leaves the buffer, it seizes the link-up area for that flowline. In the case of items leaving the buffer for A2, QUEUE A2, it seizes AYA. Then a check is made to see if an item is available from B2 for link-up. To accomplish this, a false queue is compared to one. If there is an item present, then a switch is set to signal B2 that A2 also has one ready. The item in B2 then advances to a no-operation block. A2 then checks to see if B2 has arrived in that block. At that point, the switch is reset and the item from A2 exits the system. The item from B2 then checks to see if the switch has been r-set and if so, then exits the system.

3.4.5 Failure Routines

Generating failures and repairing units constitute the second



A. Link-up for Flowline A



B. Link-up for Flowline B

Figure 14. Link-up Routine

major routine for the program. To have a unit fail, a transaction is generated which produces the failure. The transaction first seizes the failure facility. This precludes any failures from occurring while the unit is down. A SAVEX is set, recording the time the transaction was generated. Next, two SAVEX cells are set. The first, SR-- represents the time the next failure will occur if the system is operating normally. The abnormal condition is accounted for in the second, SAVEX SL--.

An abnormal condition exists when the system is load sharing and at least one other unit in the subsystem is non-operational. A comparison is made with the Boolean Variable LO--, to ascertain if the abnormal conditions are in effect. The Boolean Variable LO-- equals one if the other unit is inoperative and the system is load sharing. Now the transaction advances to another Boolean Variable, RLP--, which equals one if the clocktime is greater than or equal to the time for the next failure or the unit goes down for periodic maintenance. In either case a check is made to see which is in fact the situation, by asking if the clocktime is greater than or equal to the time for the next failure. If the latter statement is not true, then the unit is in for periodic maintenance. Thus the transaction to generate the failure is routed to END--, where it waits until the unit returns from periodic repair; then it exits the system. If the former statement is true, the production in that unit is preempted.

A record of failures is maintained by recording the time the unit fails. A logic switch is set to notify other units and other sections of the program that the unit is inoperative. Now a decision

must be made on whether the unit will be repaired by the regular repair crew or the periodic maintenance team. First, the unit checks to see if there are any regular repair teams. If so, it enters a QUEUE which will permit it to go to the regular repair team if it is not time for periodic maintenance. However, if it is time for periodic repair, it will wait to be worked on by the periodic maintenance team. If there are no regular teams, the unit enters a QUEUE which directs each unit to wait until it is time for the next periodic service.

3.4.6 Repair Routine

When repaired by the periodic maintenance team, the time for its next periodic repair is scheduled as the unit is placed back into operation. Repairs performed by the regular repair team do not change the scheduling of periodic maintenance.

Upon completion of repairs, the crew performing the maintenance is released, the switch is reset to notify others the unit has returned to operation, and production is resumed at the same point it was interrupted by the failure.

A separate routine handles scheduling and performing periodic maintenance when the unit is operational at the time periodic maintenance is due. A transaction is generated, immediately seizing the facility handling periodic scheduling. The transaction is retained in a hold block the length of time equal to the interval between periodic repairs. Upon departure, a check is made to see if the unit is down and in regular repair. If yet, it waits until the unit comes out of regular repair then performs the periodic maintenance. If not, a check is made to see if it is down. A yes response means it is going to be

repaired by the periodic maintenance team. A no means it is operational, so the transaction waits until either it can seize the periodic team or the unit fails. In either case, the periodic maintenance will be performed. Should the crew be seized, the time is recorded and the maintenance is performed. Upon completion of repair, the next repair is scheduled and the unit is returned for production.

3.4.7 Routine to Record System Availability

A transaction is generated at time zero and seizes the compute routine. It attempts to enter the COMPARE block which is monitoring the system down criteria. When the criteria are met, the transaction enters and the time that the system went down is recorded and stored.

Next the transaction seeks entry into the COMPARE block which monitors when the system down criteria are no longer being met. Upon acceptance by the COMPARE block, the time is recorded and the length of time the system was down is recorded. The downtime is added to an accumulator for total downtime. The interval of the last time the system came up and the most recent failure is known as uptime. That difference is added to an accumulator for total uptime. A counter records the number of times the system failed during the production run. The transaction then releases the compute routine and immediately another transaction seizes the routine. This is a continuous process over the interval (0,t).

3.5 Verification of the Simulation Program

The problem of verifying a computer program for a simulation model is not a very well discussed topic. Although many authors

indicate it is a necessary step in simulation, few describe any techniques for doing so. Verification and validation are often thought to be one in the same, but that is farthest away from being the case.

Kiviat and Pritsker define verification as " . . . the process of establishing the precision with which the simulation program resembles the system under study" (27). In order to verify this simulation program, each process is taken individually for logical flow.

3.5.1 Production Flow Checks

First a unit of subsystem A1 is taken from series A, isolated, and tested for its production time, without failures. The supply of raw materials at the beginning of each series is infinite, therefore, as soon as the subsystem completes its work on one item another item is ready to enter the subsystem. That type of operation is applicable to the first subsystem in each series. Then the same process is checked in one unit from each of the other three subsystems.

Subsystems A1 and A2 are in series, meaning an item worked on in A1 proceeds to A2. The only way an item gets into A2 is that it comes from A1. Each unit in A2 can accommodate only one item at a time and there is a queue, queue A1A2, between A1 and A2. The capacity of the A1A2 queue ranges from 1 to ∞. As each item leaves a unit from A1 it goes to the A1A2 queue. If the queue is full it remains in the unit of A1 that worked on it, thus continuing its seizure of that facility. The facility is released when the queue is below capacity. The queue is only empty if a unit in A2 is idle or there are no items available from any unit in A1.

Therefore, the flow goes from stockpile into A1. Upon completion

by a unit in A1, the queue is checked to ascertain if it will accept the item; from there it goes to the first available unit in A2. Once an item is finished at A2, it must link-up with an item from B2.

The link-up of series A and B proves to be quite a challenge. The original attempt to perform that feat compares the facilities at A2 and B2 simultaneously, to see if one item is available from at least one unit in each subsystem. This procedure fails because one subsystem checks and when the conditions are met, that subsystem releases its item before the other subsystem can perform its check. Consequently, one subsystem is processing its items through while the other subsystem is only getting one item through. The next attempt is to label all the items with a mutual identifying feature by generating one item and continuously splitting it as needed by either series A or B. At the end of processing the split transactions match simultaneously, then release. The problem with that is, as soon as the split starts, it continues and overloads the system before any production is allowed to take place.

Then the decision is made to establish a link-up area, Figure 14. When an item from A2 enters the area, it checks to see if an item from B2 is waiting in a link-up queue. If so, it activates a logic switch which notifies product B that A is available for link-up. The product from B, if available, proceeds to the link-up area from the queue. The model allows instantaneous link-up. Then product A checks the link-up area to insure that B also arrives; if so, A deactivates the logic switch. Product B, meanwhile, checks if the logic switch is deactivated. That being accomplished A releases its position

in the link-up area and B releases its portion.

3.5.2 Failure and Repair Routines

After the production flow proves satisfactory, the failure and repair routines are checked. In the first attempts, the failure routine consists of generating a failure transaction for each unit of the system, holding it for a random period of time equivalent to a time to failure. At the end of that time the production routine is interrupted due to a failure. This technique proves unsatisfactory because the actual mean time to failure is much less than the value which is a part of the input. That routine ignores the fact that a unit should have a new time to failure assigned when it gets out of periodic maintenance.

To correct the oversight, a conditional hold is used for the length of time to next failure. As the failure transaction is generated, the time is recorded; added to that time is the length of time to the next failure. Then the transaction is placed in an ADVANCE block followed by a COMPARE statement. The comparison only allows the transaction to advance if (1) the time has elapsed for a failure to occur, or (2) production has been preempted for the unit to enter periodic maintenance. That satisfies the logic error associated with excess frequency of the units being non-operational. The same type conditional advance is used in the routine which generates periodic maintenance.

When a unit fails, it checks the clocktime to find out if it is time for periodic repair. The repair routine does not cover the situation of the time elapsing for periodic maintenance while the unit is in regular repair. Thus the units remain in a down status for the

duration of the run once the above mentioned condition occurs. The reason it remains down is that the unit is being placed in two locations at once and consequently, it cannot return because each location keeps checking the other. Statements are inserted to have the transaction in periodic maintenance check to see if the unit is in regular repair or in periodic maintenance after it fails. If it is found to be in regular repair then the transaction waits until the unit is released from regular repair to preempt the production routine. If the unit is found to be in periodic repair, then it goes to the end and terminates as soon as the unit leaves the maintenance area.

3.5.3 Record of System Down

The final operation to check is the routine which specifies when the system is considered down or returned to an operational level, based on the criteria set forth by the user. If the system has four subsystems, each subsystem has two units and the criteria for system down are any two subsystems down, then all combinations of two subsystems being down are checked. To further complicate matters some of the subsystems may have the criterion that if one unit is down, the subsystem is down; while the others may require that both units be down before the subsystem is down. All the possible combinations of criteria must be covered for the system to have the desired flexibility. The Boolean Variable permits multiple comparisons simultaneously, i.e., a Boolean Variable may read:

COMPARE (Unit A11 inop) and (Unit A12 inop).

Thus two statements can accommodate the situations of: (1) either unit

being inoperative by using "or", or (2) both units being down by using "and". This is done for each subsystem.

Consider system criteria of system down when two subsystems are down accompanied by (1) above for two of the subsystems, and (2) above for the other two. Without counting all the possibilities, a large number of COMPARE statements are required to account for all the combinations by units. At this point an experiment is performed by having a test program utilize a Boolean variable within a Boolean variable. There is no evidence that this is possible, but it seems logically feasible. The successful outcome of such an experiment reduces the required number of comparisons by more than half. By placing a Boolean variable within a Boolean variable, the capability now exists for satisfying the system's criteria in one statement, rather than as many as five statements, in some cases. Therefore, if the system requires that all subsystems be operational, the system criteria reads

COMPARE (BV for A1) OR (BV for A2) OR . . . .

This technique works; therefore, the comparisons are not made on the bases of combinations of units, but rather by subsystems.

3.5.4 Combining All Routines

As the last step, the production, failure, and repair routines are combined. As a check that the entire system is operating properly, the model is made deterministic by removing all random generators. The simulation is then run and the calculated results are obtained. When the random generators are replaced, the near same results are obtained.

3.6 Model Validation

One approach to model validation is to experiment with a list of all possible actions the system can take. This is a very costly and time consuming approach to the problem. In view of this, the approach used in this research is to validate a specific set of insights, not necessarily the generator of those insights.

Kiviat and Pritsker define validation as " . . . the process of establishing the accuracy with which the simulation program resembles the system under study" (27). In fulfillment of that definition, the preferred approach is to validate by comparing the simulation data with data from an actual system.

3.6.1 Comparisons to Closed Form Results

When there is no actual system available for comparison in validation, Maylor suggests two other avenues. The first is comparing replications by changing the random generators; the second is by comparing the simulation to a similar system for comparison (25).

The second approach is more applicable in this case; therefore, Applebaum's model is used for validation. In order to do this, all the dependent variables are removed from the model, thus assuming independence. The changes required to parallel Applebaum's model place the model variables into two categories to ascertain a comparable model: (1) collapsing and (2) removal.

3.6.1.1 Collapsing the Module to Independence. The first category contains all variables which were modified or collapsed. Among these are: (1) maintenance capability, (2) production rate, and (3) failure rate. The maintenance capability required for independence

entails having N repair facilities, where N represents the number of units. In the simulation model, the number of units in a subsystem can be greater than one, with one repair team for the subsystem. Applebaum's model does not accommodate multiple units in a subsystem with one repair team, nor does it accommodate a periodic maintenance policy. Periodic maintenance, therefore, is collapsed to an infinite interval. Thus, in this case, the maintenance capability must be expanded from the designed model to achieve independence.

Production rate must be collapsed. The time required to produce the items may be a constant or some distribution with varying times. In either case, for independence, the production rate must retain the same mean throughout the production run. Such is not the case in the designed model. The mean production rate varies with the number of units in for repair. To achieve independence, that varying feature, load sharing, is collapsed to a single case.

The third variable--time to failure--is also collapsed. The same explanation as used for production rate is applicable to the time to failure density, in that the designed model has the mean time to failure changing based on production rate. When the subsystems are independent, the mean of the time to failure density does not change. For the validation, time to failure, and time to repair are exponentially distributed.

3.6.1.2 Removal of the Auxiliary Variables. The second category contains those variables which are removed to achieve independence: (1) The capability of having series operations is removed. The instant subsystems are placed in series, independence is eliminated. If the

first subsystem fails, then the others must wait to continue. The only way to regain that independence with series is to have in-process buffers of infinite capacity and have them sufficiently stocked at time zero. (2) The capability of parallel operations of different types working together to produce a single final product is not available. The reason again is that this situation does not allow independence. One subsystem is dependent on the other for link-up of its components.

Independence is restored only if in-process storage is used with sufficiently large capacity that a component is always available when needed by the other subsystem. (3) The facility efficiency for short batch operations cannot be compared. Applebaum's model is applicable to steady-state only, and since short batch operations often do not reach steady-state, the efficiency cannot be determined using his techniques. (4) The need for in-process buffers is nonexistent. If the subsystems are independent, the situation cannot arise when one subsystem has to wait for another since that implies dependence. (5) Subsystems cannot share any facility, be it an operator or equipment. When the facility is in use by one subsystem, it is not available to any other. Thus, there is a dependence if that facility is shared. In addition to the removal of dependency from the model, the comparison with Applebaum could only be done for a preloaded system.

3.6.2 Comparison of Results

The inputs for both the example from Applebaum and the computer simulation are as follows:

number of subsystems = N = 3,

number of systems which must be operational for the system to

be operational is defined as M,

the mean time to failure is $\overline{u}$, where failure is exponentially distributed in minutes, and

the mean time to repair is $\overline{d}$, where repair time is exponentially distributed in minutes.

Using Applebaum's example, the subsystem of the research model can be considered a system and the data can be cross checked. Thus, when

$$\overline{u} = 10$$

$$\overline{d} = 8,$$

and $$M = 1$$

subsystem availability, a, is

$$a = \frac{\overline{u}}{\overline{u} + d} = \frac{10}{10 + 8} = .55556$$

unavailability,

$$\hat{a} = 1 - a = .44444.$$

The maintenance ratio, r, is

$$r = \frac{\overline{d}}{\overline{u}} = \frac{\lambda}{\mu} = \frac{\hat{a}}{a} = .8 \quad .$$

System availability as defined by Applebaum is

$$A_s = \sum_{K=M}^{N} \binom{N}{K} a^K \hat{a}^{N-K} \qquad 3\text{-}1$$

then,

$$A_s = .9122, \text{ and}$$

$$\hat{A}_s = 1 - A_s = .0878.$$

The system MUT is

$$\overline{U}_s = \frac{\sum_{K=M}^{N} \binom{N}{K} a^K \hat{a}^{N-K}}{\binom{N}{M} a^M \hat{a}^{N-M} M\lambda} \qquad 3\text{-}2$$

or,

$$\overline{U}_s = \frac{\overline{U} \sum_{K=0}^{N-M} \binom{N}{K} r^K}{M\, r^{N-M} \binom{N}{M}} \qquad 3\text{-}3$$

Thus from equation 3.2 or 3.3

$$\overline{U}_s = 27.71 \text{ minutes.}$$

To insure the computer simulation reaches steady-state conditions, the system runs for 4000 minutes. The results of the first two simulation runs, Table 1, show A_s over the interval (300,2000) is .9175; A_s over the latter half interval (2001,4000) is .9065. Both compare very closely with the theoretical A_s of .9122. System MUT, $\overline{U}_s$, over those intervals are 27.68 and 27.33, respectively. These values are also well within range of the theoretical $\overline{U}_s$ of 27.71. Table 2, also gives the results of other simulation runs made in comparison to the

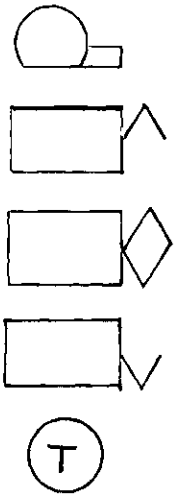
Table 1. Results of Validation Runs

N	M	$\bar{u}$	$\bar{d}$	A_s		$\bar{U}_s$	
				Applebaum	Simulation	Applebaum	Simulation
3	1	10	8	.9122	.9175	27.71	27.68
3	1	10	8	.9122	.9065	27.71	27.33
3	1	20	8	.9766	.9608	111.70	100.30
3	1	50	10	.9955	.9957	716.22	753.20
3	2	50	10	.9260	.9435	66.67	69.60

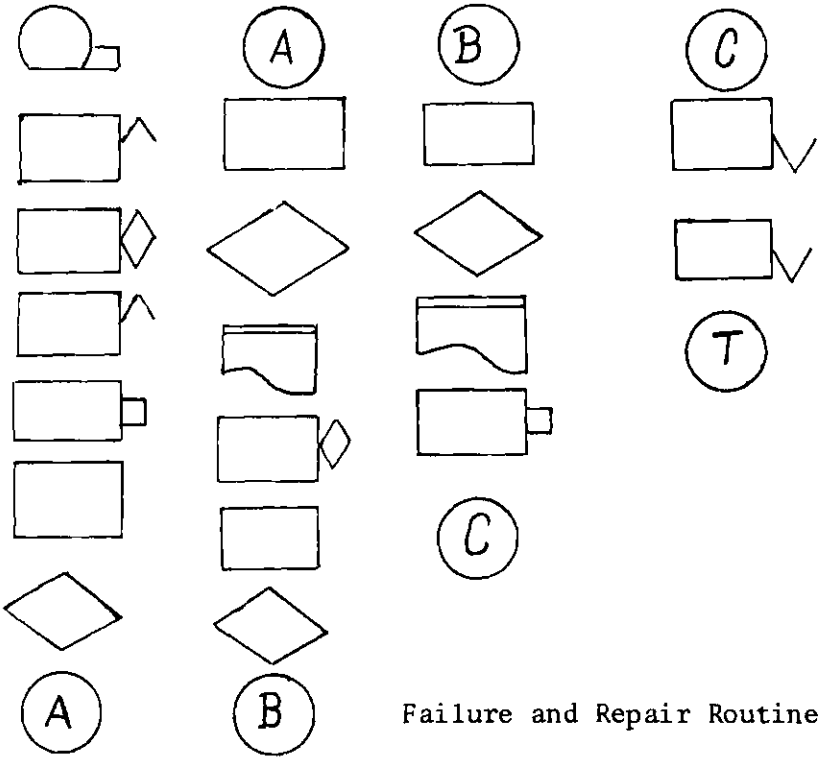
theoretical computations. The validation program is flow charted in Figure 15. This flow chart corresponds to N = 3 and M = 2. All runs after the first two cover the time interval (300,4000).

In addition to validating the logical flow of the program, the validity of the string of random numbers generated is also obtained in this phase. After getting values close to those using Applebaum's method, the random number generators are varied in several runs to establish a confidence interval on the results obtained. Using the input parameters from run 5 in Table 2, four additional runs are conducted. The results for availability are .9014, .9422, .9206, and .9392. Therefore, the expected value, using the input variables of run 5, is .9293 with a variance of .03.

Six additional runs are made to establish a confidence interval for the system's mean uptime, $\bar{U}_s$. For the run with a $\bar{U}_s$ value of 100.3 from the simulation, the variance is 16.2. The variances for the remaining runs with $\bar{U}_s$ of 753.2 and 69.6 are 57.3 and 8.8, respectively.



Production Routine



Failure and Repair Routine

Figure 15. Validation Flow Chart

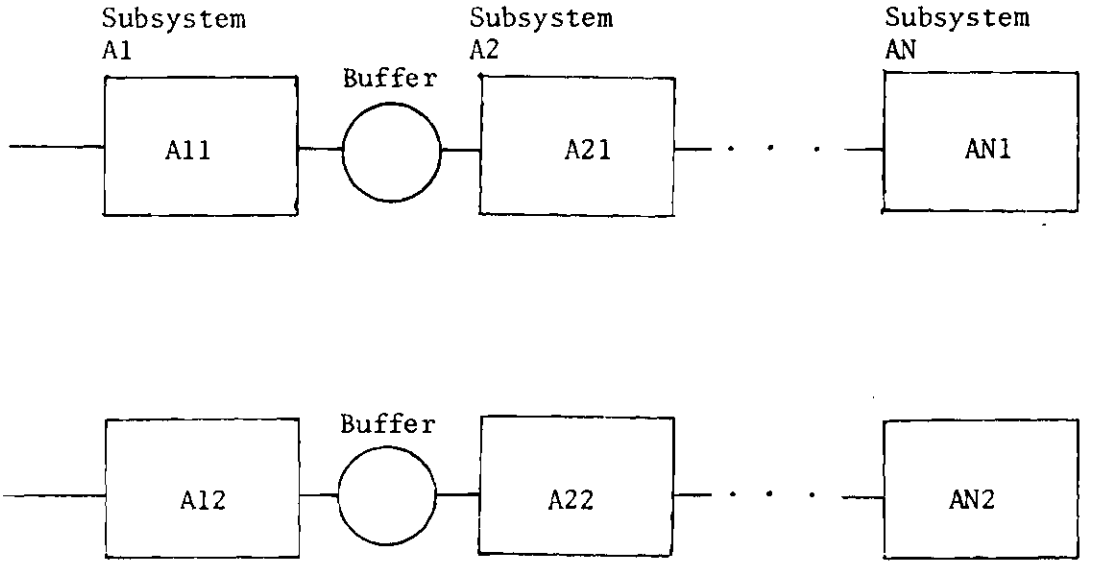
3.7 Modular Control Variables

The input parameters of this model include the variables discussed by Applebaum, Buzacott, Law and others mentioned in Chapter II (1, 10, 20, 32, 17, 23). The major difference between those models and this is the input variables in this model are not assumed to be independent. In addition to those variables, others such as equipment sharing and preloading are also included. The input variables, referred to as factors, are selected as a result of an effort to include numerous facets of a complex system. A description of the modules' capabilities are described below.

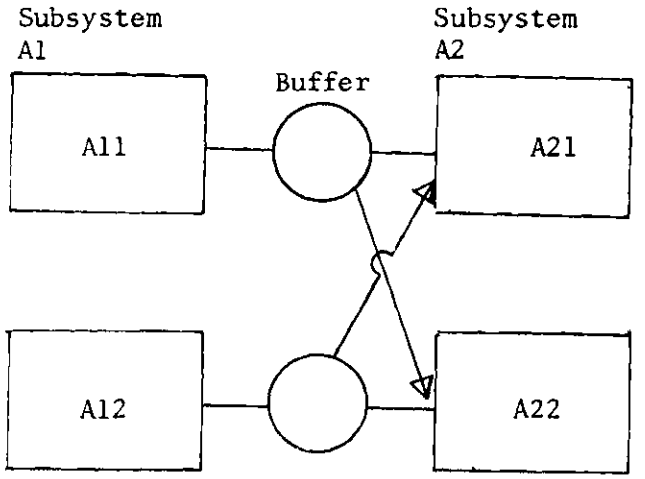
3.7.1 In-Process Buffers

The question of in-process buffers has been addressed previously and remains a feasible feature for production systems (4, 20, 32). The optimal capacity of an in-process buffer depends on several considerations of the system. The production rate and failure rate immediately crop up as major considerations. However, the factor of cost cannot be excluded. This research is designed to probe for the effect that buffer capacity will have upon the system, while simultaneously including the other variables in the system. As stated in Section 3.4, there is one buffer serving as a central dump for all units of a given subsystem. The units of the next subsystem of that flowline draw from that buffer.

The other alternative configurations, Figure 16, are: (1) a buffer for each unit, where $A_{i,j}$ supplies $A_{i+1,j}$, and (2) a buffer for each unit of the preceding subsystem and units of the succeeding subsystem draw randomly from those buffers. Neither of the last two



A. Unit $A_{i,j}$ Supplying $A_{i+1,j}$



B. Unit $A_{i,j}$ Supplying $A_{i+1,\text{all } j}$

Figure 16. Alternative Buffer Layouts

alternatives is considered in this research. However, the model can accommodate either configuration with minor adjustments. Some constraints which may require one of the latter configurations include cost, insufficient plant space, or plant layout.

3.7.2 Regular and/or Periodic Maintenance

The question is sometimes asked, "Which is better, having periodic maintenance or repair on a 'fix as it fails; basis?" The operating circumstances may require the continuous presence of a repair team. While on the other hand, the time to failure may be sufficiently large that it would be sufficient to shut down the system or parts thereof to perform periodic maintenance. Another possibility is that the repair team is cheaper if it comes for periodic maintenance rather than on a "fix as it fails" basis. The contingencies for maintenance policies include 'fix as it fails' repair, hereafter referred to as regular repair, and periodic maintenance.

3.7.3 Failure Rate

For every piece of equipment that operates, there is some probability that it will fail. That probability may lie anywhere between 0 and 1, inclusive. The sequence of failures over time form a distribution of failures. The interval--time placed into operation and failure--is known as time to failure (TTF). The mean of such a distribution is referred to as the mean time to failure (MTTF). The units of a subsystem have the same failure rate for this analysis. However, the module is capable of operating with different failure rates for units of the same subsystem. The subsystems may also have different failure rates. The time to failure distribution may be any

in the arsenal of GPSS.

3.7.4 Production Rate

The time required by a unit to complete its designed operation on an item, without interruption, is the production time. Production rate for a subsystem is the average output per unit of time. In this case the unit of time is 100 minutes. Thus if the production time is 100 minutes for the units of a subsystem, that subsystem requires and has only one unit. The production time of the subsystem, therefore, dictates the number of units the subsystem will have. This feature maintains the line balance throughout the system. Production rate, too, can be considered an exogenous variable. This variable is also subdivided to represent each subsystem separately.

3.7.5 Repair Rate

How fast a unit or subsystem is placed back into action is dependent upon the availability of a repair team and the repair rate. Repair rate, like failure and production rates, is subdivided to form four variables--one for each subsystem. The same rates apply to regular repair as apply to periodic maintenance. The time to repair density may be exponential or any other distribution that GPSS will accommodate.

3.7.6 Preloading

When a production run ends, one of two conditions exists. In the first condition, all items that are in the system complete processing and no new items enter. The second condition is such that at time n the system is shut down with items remaining in the system. The variable--preloading--nearly approximates the second condition. Instead of having the system left with items in the middle of production in a

unit, preloading means an item is available for processing by each unit in each subsystem.

3.7.7 Load Sharing

If the production rate is some set figure and there are two or more units in that subsystem, when one unit fails the production time of the other unit is decreased to maintain the desired production rate. A subsystem or system operating in this manner is said to be load sharing. This can also be stated as follows: given a prescribed load, L, for a subsystem with s units, the load carried by each unit when all units are operational is L/s; should r of those units fail, $s > r$, the load carried by the remaining units is $L/(s-r)$. When a module is load sharing and a subsystem has two or more units, the load sharing factor is τ, such that $1 \leq \tau < 2$. That means if one or more units fail, the remaining units can increase their production rate up to τ times the normal rate.

3.7.8 Equipment Sharing

Different subsystems sometimes have a part of the processing performed by another machine or they may share an operator. As an example, two or more subsystems may require a certain operation be done by the machinist, and the system may only have one machine shop. Sharing of an operator or equipment has been labelled equipment sharing. To portray this characteristic of a system, the model has flowline sharing. There are two flowlines, therefore, equipment sharing is represented as two variables, one per flowline. For the test module, the time spent at the shared facility comes at the end of the production process for that unit. However, the processing by the shared

facility can take place at any point in the production cycle by a given unit.

3.7.9 Subsystem Down Criteria

Each subsystem's down criteria are independent of the other. Therefore, again, this variable becomes four. These variables facilitate an assessment of effect that the subsystem's configuration has upon its availability. Since the module's subsystems have one or two units, the down criteria are restricted to one or two units being down before the subsystem is down.

3.7.10 System Down Criteria

The possible combinations for system down criteria are greatly increased over the possibilities for subsystem down criteria. The requirement for system down may be one subsystem down. In that case, there are four possibilities. Letting U represent the set of "up" conditions, U consists of {all units operational}. The set D, "down" conditions, consists of all combinations not in U. In addition to having four possibilities, the simulation must insure that if a subsystem goes down and at least one other subsystem is down, the system is not recorded as going down again. In the case where system down criterion is all subsystems non-operational,

$$D = \{\text{all subsystems down}\}$$

and

$$U = D' \quad .$$

This variable can be accommodated by the model in any configuration from one subsystem to all subsystems being down.

3.8 Modular Responses

The results or responses from the computer simulation runs are described as modular responses. Whenever the simulation represents a module of a larger system, the production output is fed into the next module. The modules are sequentially interfaced until the entire system is included.

Section 4.5 contains an illustration of how the responses from one module are utilized to project output from subsequent modules. If these responses are from the last module then the productivity of that module is the same as for the system.

3.8.1 Productivity

The number of items produced over a production run is the primary measure of productivity. The maximum output for the module is the run length divided by the minimum production time of the last subsystem. When maximizing productivity is mentioned with respect to this model, reference is being made to coming close to having the maximum output.

Each time an item enters a subsystem or departs a subsystem, the clocktime is recorded. The purpose for these statistics is to have inputs for the next module, such that, the arrivals appear at the time equated to departure from the preceding module. The model also provides the average production time for items from the module, thus providing average production time and a temporal account of productivity.

3.8.2 Availability

The second response generated is the availability of the system. The measure of availability is based on the criteria in effect during a given run. Therefore, in addition to being able to measure availability based on system output, a definite set of criteria provide the inputs for computing availability. The use of such criteria is very important because in complex systems of great magnitude, the output of a part of the system may not produce a countable number of items. And even if a large number of items is produced, that is not a completely accurate account of system availability. The two inputs generated by the program are mean uptime (MUT) and mean downtime (MDT). From these inputs the availability of the system can be computed.

In addition to the averaged computations, statistics are kept on each subsystem and the system, giving when they went down and the time they were returned to action. This temporal account of availability serves the same purpose as that for productivity.

CHAPTER IV

THE EFFECTS OF THE MODULE CONFIGURATION ON PRODUCTIVITY AND AVAILABILITY

The program described in Chapter III is used to analyze the effects of module configuration alternatives on system productivity and availability. The module configuration diagramed in Section 3.3 is used in the experimental analysis being comparable in size to the systems cited in the literature (1, 3, 6, 19, 28). In addition to having the same features as the systems in the literature, the experimental system contains: (1) equipment sharing, (2) preloading, and (3) periodic and regular repair considerations.

Inferences about the effects of different configurations and changing the operating conditions are obtained by performing simulation experiments. The experiments consist of simultaneously changing the levels of all 27 configuration constraints and observing the system's productivity and availability.

Section 4.1 gives the 27 variables for the module and the fractional factorial design used in the first experiment.

The responses and analyses from the first experiment are presented in Section 4.2. The establishment of a need for and the results of the second experiment are in Section 4.3. The combining of the first and second experiments to yield the unanalyzed estimates of the main effects are also contained in Section 4.3. Sections 4.4 and 4.5 discuss the third experiment, and the combined analysis of all

experiments, which include the delineation of significant effects and the maximization of productivity and availability by steepest ascent and branch and bound mixed-integer programming techniques, respectively. Section 4.6 presents the system transient and steady-state effects on productivity and availability viewed through batch size. The results of interfacing two modules are discussed in Section 4.7 and further comparisons of the simulation results to Applebaum's model are described in Section 4.8.

It should be noted that the inferences gleaned from the sequences of experiments are not intended to be directly translated for all system configurations. The same technique, however, may be applied using a modular approach with the GPSS module to facilitate other specific systems. On the other hand, the particular system being analyzed will keynote specific configuration constraints one might look to for leverage in improving a particular system's productivity and/or availability.

4.1 A 2^{27-22}_{III} Fractional Factorial Design

4.1.1 Levels of the Variables

Each of the variables is tested at two levels, as shown in Table 2. Some of the variables are qualitative because of their experimental levels, while the remaining are quantitative for the same reason. All the variables listed as qualitative indicate an absence or presence of certain test conditions. The quantitative variables represent a range for the degree influence exerted by the variable. The variables in Table 2, categorized as qualitative are: (1) buffer capacity, (2) load sharing, (3) subsystem down criteria, (4) system

Table 2. Listing of Variables and the Levels

Variables	Levels	
	High	Low
(1) Production Rate A1	100	200
(2) Production Rate A2	100	200
(3) Production Rate B1	100	200
(4) Production Rate B2	100	200
(5) MTTF A1	960	120
(6) MTTF A2	960	120
(7) MTTF B1	960	120
(8) MTTF B2	960	120
(9) MTTR A1	180	20
(10) MTTR A2	180	20
(11) MTTR B1	180	20
(12) MTTR B2	180	20
(13) Buffer Capacity		1
(14) Production Rate Criteria	Loadsharing = 1	No Loadsharing = 0
(15) Subsystem Down Criteria A1	All Down	One Down
(16) Subsystem Down Criteria A2	All Down	One Down
(17) Subsystem Down Criteria B1	All Down	One Down
(18) Subsystem Down Criteria B2	All Down	One Down
(19) System Down Criteria	All Down	One Down
(20) Periodic Repair A1	10005	240
(21) Periodic Repair A2	10005	240
(22) Periodic Repair B1	10005	240
(23) Periodic Repair B2	10005	240
(24) Equipment Sharing Flowline A	1	0
(25) Equipment Sharing Flowline B	1	0
(26) Regular Repair Teams	1 per Subsystem	0
(27) Preloading Policy	1 = Yes	0 = No

down criteria, (5) periodic repair, (6) equipment sharing, (7) regular repair teams, and (8) preloading policy. The remaining variables--production rate, mean time to failure, MTTF, and mean time to repair, MTTR--are quantitative.

4.1.2 Experimental Design

As previously stated, the model has 27 variables, each to be tested at two levels as discussed in Section 4.1.1. Thus, there are 2^{27} or more than 134 million possible combinations which can be tested. It is both economically and physically infeasible to conduct that many experiments. The necessity of a fractional factorial design is evident. However, with a fractional design, the ability to estimate the effects resulting from certain higher order interactions is lost. The criteria outlined by Box and Hunter (5) were used to choose an appropriate fraction. The first assessment is of the main effects and the aliased two-factor interactions. Therefore, the first fraction must yield at least as many experiments as there are variables. Since there are 27 variables, the minimum fraction is 2^5 or 32 experiments. The resulting design is a 2^{k-p} or 2^{27-22}, where k is the number of variables and p is the number of words in the defining relationship.

The design matrix for a 2^5 design for this experiment has five independent variables and the remaining 22 variables are represented by two, three, and four factor interactions. The resulting 22 words are listed in Equation 4-1 below. The layout of levels for each variable on each experiment is in Table 3. The runs coincide with the number of the experiment. The variable numbers are the same as those in Table 2.

6=1*2	I=1*2*6	
7=1*3	I=1*3*7	
8=1*4	I=1*4*8	
9=1*5	I=1*5*9	
10=2*3	I=2*3*10	
11=2*4	I=2*4*11	
12=2*5	I=2*5*12	
13=3*4	I=3*4*13	
14=3*5	I=3*5*14	
15=4*5	I=4*5*15	
16=1*2*3	I=1*2*3*16	
17=1*2*4	I=1*2*4*17	4-1
18=1*2*5	I=1*2*5*18	
19=1*3*5	I=1*3*5*19	
20=1*3*4	I=1*3*4*20	
21=1*4*5	I=1*4*5*21	
22=2*4*5	I=2*4*5*22	
23=2*3*4	I=2*3*4*23	
24=2*3*5	I=2*3*5*24	
25=3*4*5	I=3*4*5*25	
26=1*2*3*4	I=1*2*3*4*26	
27=1*2*3*5	I=1*2*3*5*27	

Table 3. Design Matrix for 2^{27-22} Design (1st 32 Runs)

Runs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	-	-	-	-	-	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	+	+
2	+	-	-	-	-	-	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-
3	-	+	-	-	-	-	+	+	+	-	-	-	+	+	+	+	+	+	-	-	-	+	+	+	-	-	-
4	+	+	-	-	-	+	-	-	-	-	-	-	+	+	+	-	-	-	+	+	+	+	+	+	-	+	+
5	-	-	+	-	-	+	-	+	+	-	+	+	-	-	+	+	-	-	+	+	-	-	+	+	+	-	-
6	+	-	+	-	-	-	+	-	-	-	+	+	-	-	+	-	+	+	-	-	+	-	+	+	+	+	+
7	-	+	+	-	-	-	-	+	+	+	-	-	-	-	+	-	+	+	+	+	-	+	-	-	+	+	+
8	+	+	+	-	-	+	+	-	-	+	-	-	-	-	+	+	-	-	-	-	+	+	-	-	+	-	-
9	-	-	-	+	-	+	+	-	+	+	-	+	-	+	-	-	+	-	-	+	+	+	+	-	+	-	+
10	+	-	-	+	-	-	-	+	-	+	-	+	-	+	-	+	-	+	+	-	-	+	+	-	+	+	-
11	-	+	-	+	-	-	+	-	+	-	+	-	-	+	-	+	-	+	-	+	+	-	-	+	+	+	-
12	+	+	-	+	-	+	-	+	-	-	+	-	-	+	-	-	+	-	+	-	-	-	-	+	+	-	+
13	-	-	+	+	-	+	-	-	+	-	-	+	+	-	-	+	+	-	+	-	+	+	-	+	-	+	-
14	+	-	+	+	-	-	+	+	-	-	-	+	+	-	-	-	-	+	-	+	-	+	-	+	-	-	+
15	-	+	+	+	-	-	-	-	+	+	+	-	+	-	-	-	-	+	+	-	+	-	+	-	-	-	+
16	+	+	+	+	-	+	+	+	-	+	+	-	+	-	-	+	+	-	-	+	-	-	+	-	-	+	-
17	-	-	-	-	+	+	+	+	-	+	+	-	+	-	-	-	-	+	+	-	+	+	-	+	+	+	-
18	+	-	-	-	+	-	-	-	+	+	+	-	+	-	-	+	+	-	-	+	-	+	-	+	+	-	+
19	-	+	-	-	+	-	+	+	-	-	-	+	+	-	-	+	+	-	+	-	+	-	+	-	+	-	+
20	+	+	-	-	+	+	-	-	+	-	-	+	+	-	-	-	-	+	-	+	-	-	+	-	+	+	-
21	-	-	+	-	+	+	-	+	-	-	+	-	-	+	-	+	-	+	-	+	+	+	+	-	-	-	+
22	+	-	+	-	+	-	+	-	+	-	+	-	-	+	-	-	+	-	+	-	-	+	+	-	-	+	-
23	-	+	+	-	+	-	-	+	-	+	-	+	-	+	-	-	+	-	-	+	+	-	-	+	-	+	-
24	+	+	+	-	+	+	+	-	+	+	-	+	-	+	-	+	-	+	+	-	-	-	-	+	-	-	+
25	-	-	-	+	+	+	+	-	-	+	-	-	-	-	+	-	+	+	+	+	-	-	+	+	-	-	-
26	+	-	-	+	+	-	-	+	+	+	-	-	-	-	+	+	-	-	-	-	+	-	+	+	-	+	+
27	-	+	-	+	+	-	+	-	-	-	+	+	-	-	+	+	-	-	+	+	-	+	-	-	-	+	+
28	+	+	-	+	+	+	-	+	+	-	+	+	-	-	+	-	+	+	-	-	+	+	-	-	-	-	-
29	-	-	+	+	+	+	-	-	-	-	-	-	+	+	+	+	+	+	-	-	-	-	-	-	+	+	+
30	+	-	+	+	+	-	+	+	+	-	-	-	+	+	+	-	-	-	+	+	+	-	-	-	+	-	-
31	-	+	+	+	+	-	-	-	-	+	+	+	+	+	+	-	-	-	-	-	-	+	+	+	+	-	-
32	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

4.2 The First Experiment

The layout of the design in Table 3 gave the level at which each variable was to be tested on each run in the first fraction. After running the 32 simulations, the output provides the responses as described in Section 3.8.

4.2.1 The Responses

4.2.1.1 Average Productivity and Availability. The output of productivity ranged from 0 to 89, while the maximum possible output was 99 over the time (0,10000) minutes. The average output for all experiments was 21 items of a possible maximum of 99 items over the production run. Since one objective is to maximize productivity, the variable levels of the run producing 89 items are examined. The variable levels of interest include: (1) the repair rate of each subsystem is at its low level; (2) the system is preloaded with infinite buffer capacity; (3) periodic maintenance is absent from all subsystems; (4) the failure rate of the second subsystem in the first flowline, A2, is the only subsystem with mean time to failure at high level; and (5) each subsystem in flowline B has two units.

Table 4 contains the responses from the first fraction. As that table is reviewed, it is important to remember that there is not a one to one relationship between the two responses. Therefore, there is nothing theoretically wrong with having a high production rate and low availability, or vice versa.

Those cases where the production level is below 8 are the results of the absence of repair teams and an infinite periodic maintenance interval in at least one subsystem. In addition, the

Table 4. Responses from First Fraction

Run Number	Production	Availability	Run Number	Production	Availability
1	56	100	17	77	100
2	0	0	18	12	0
3	2	0	19	1	99.8
4	89	100	20	18	22.9
5	0	98.2	21	0	0
6	26	14.2	22	47	99
7	14	99.7	23	20	.3
8	1	0	24	28	97
9	0	0	25	1	98.5
10	2	98.1	26	27	17
11	34	6.5	27	20	100
12	27	99.6	28	1	0
13	8	99.9	29	82	67
14	42	100	30	0	95
15	0	28	31	0	0
16	44	24	32	75	100

runs with production outputs between 1 and 8 were all preloaded.

Availability ranges from 0 to 100 percent. The response tends to be at the extremes in this fraction. However, in the 32 experiments the availability tends to both extremes equally.

In addition to the responses for the system, subsystem data are also present. This makes it possible to observe subsystem performance as well. It is important to have that information available to check the differences between the means of distributions that were input, and means resulting from the random numbers generated. There were no cases of large discrepancy greater than the variance established during the validation.

4.2.1.2 Temporal Productivity and Availability. Each simulation run contains the time series for completed subassemblies from the module. To describe the input scheme for the next module, the production times are graphically plotted. The x-axis represents the ith item produced by the system and the y-axis represents the rate of productivity associated with the ith item. The rate of productivity is the ratio of one item produced to the time interval between that item and the (i-1) item.

Figures 17 and 18 are temporal plots of runs 4 and 16 of the first experiment. These two figures are examples from the first 32 experimental runs. They are used to illustrate the temporal out over the production run. The time intervals for productivity in run 4 appear to establish some regularity after the twelfth item is produced. The regularity is easily identified in this run with a large production level, while the regularity in run 16 is not as easily identified.

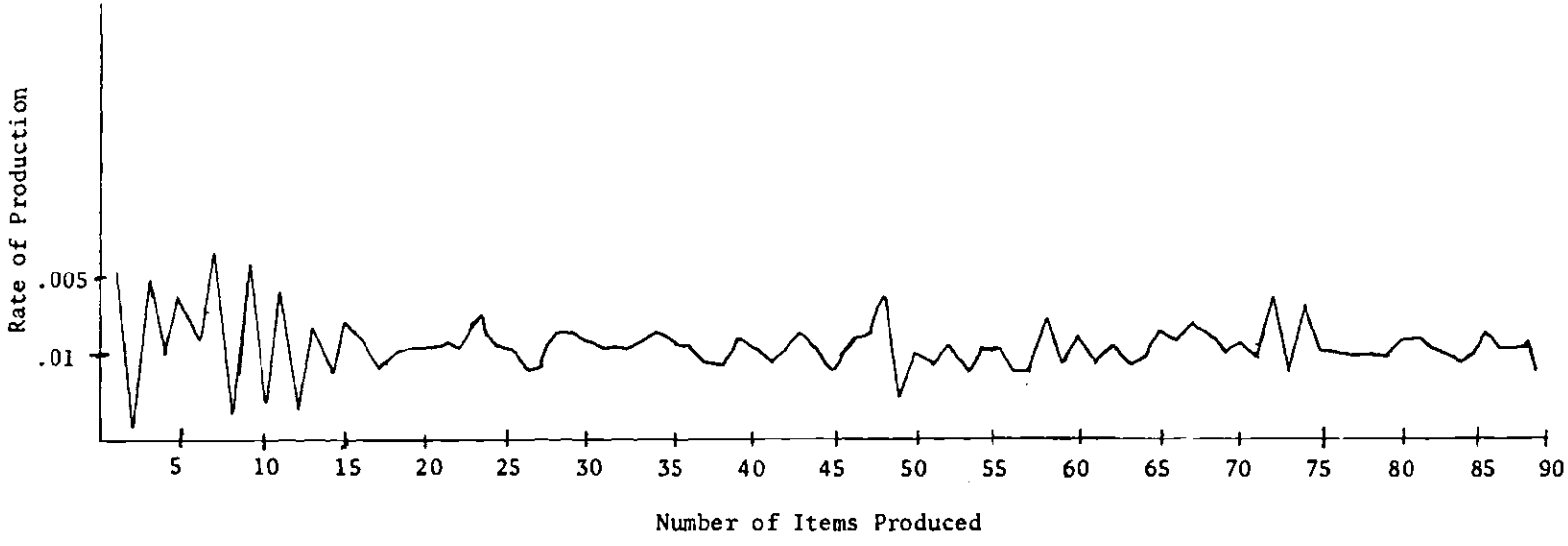


Figure 17. Production Interval--Set 1, Run 4

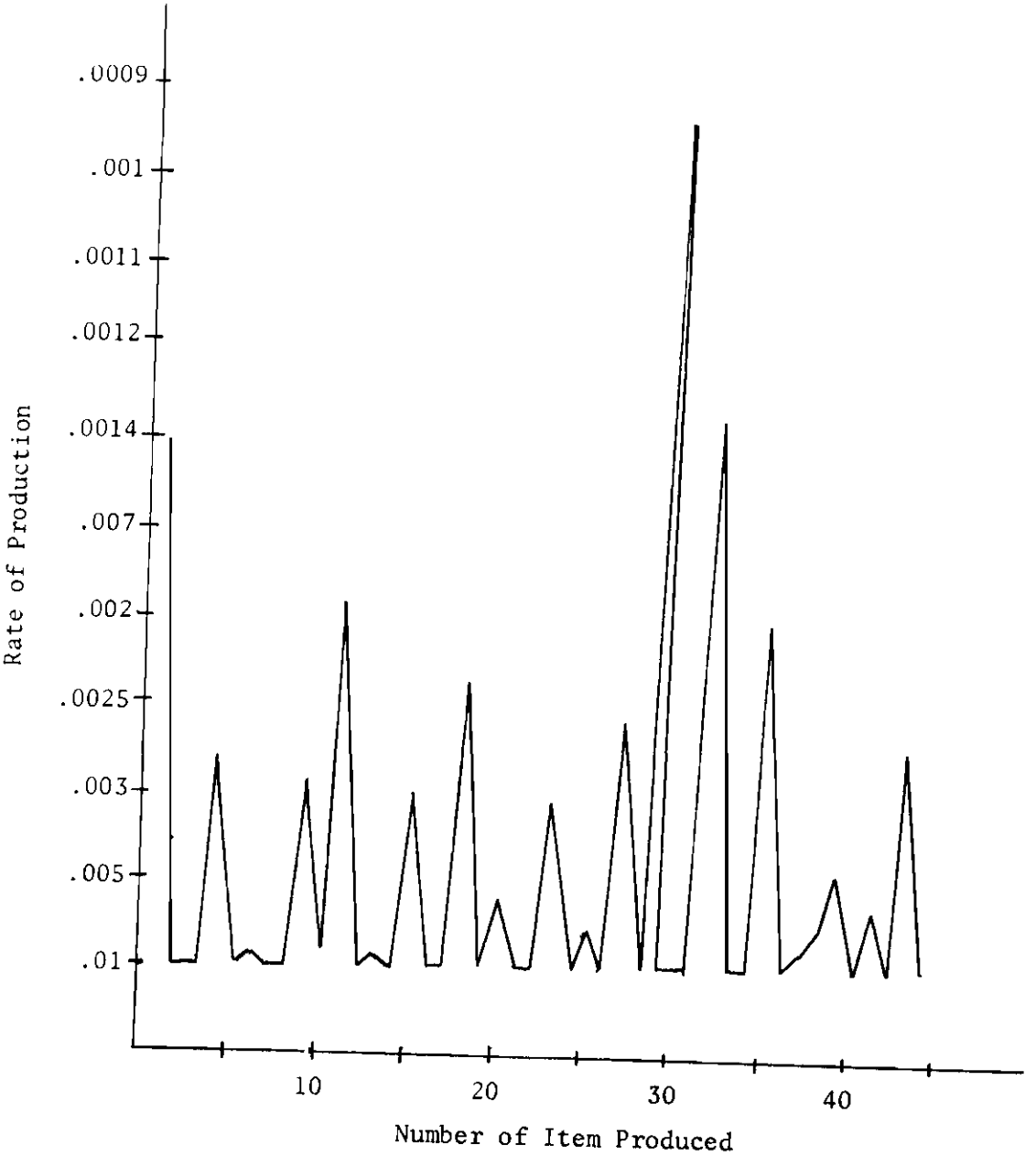


Figure 18. Production Interval--Set 1, Run 16

The time intervals as shown in Figures 17 and 18 serve as the intervals for inputs into the next module. These two series are used in Section 4.7 to present the responses of the second module.

4.2.2 The Analysis

As the responses were reviewed, some inferences could be observed almost immediately. The first observation was those runs with a production level of zero all had two things in common: (1) the systems were void of a regular repair capability, and (2) at least one or more subsystems did not have periodic maintenance performed during the production run.

There were four production runs with high levels of productivity. In all four cases, the system had regular repair teams and periodic maintenance was performed on a maximum of two subsystems. After reviewing the obvious results, other less than obvious results were sought.

4.2.2.1 Estimate of Contrast. Using the 22 defining words from Equation 1 and combining them, first two at a time, then three at a time, four at a time, . . . , and 22 at a time, the complete defining relationship is derived. Following that procedure yields over four million terms in the defining relationship. Grauer (12) developed a Fortran computer program which outputs the aliasing structure for designs with a maximum of 32 variable and 28 defining words. The defining relationship then gives rise to the total aliasing structure, Equation 4-2, assuming all three factor and higher interactions are negligible.

$$L_1 = 1 + 24*27 + 23*26 + 15*21 + 14*19 + 13*20 + 12*18 + 11*17 + 10*16 + 5*9 + 4*8 + 3*7 + 2*6$$

$$L_2 = 2 + 20*26 + 19*27 + 15*22 + 14*24 + 13*23 + 5*12 + 4*11 + 3*10 + 8*18 + 8*17 + 7*16 + 1*6$$

$$L_3 = 3 + 18*27 + 17*26 + 15*25 + 5*14 + 4*13 + 12*24 + 11*23 + 2*10 + 9*19 + 8*20 + 1*7 + 6*16$$

$$L_4 = 4 + 16*26 + 5*15 + 14*25 + 3*13 + 12*22 + 2*11 + 10*23 + 9*21 + 1*8 + 7*20 + 6*17$$

$$L_5 = 5 + 16*27 + 4*15 + 3*14 + 13*25 + 2*12 + 11*22 + 10*24 + 1*9 + 8*21 + 7*19 + 6*18$$

$$L_6 = 6 + 1*2 + 21*22 + 20*23 + 19*24 + 5*18 + 4*17 + 3*16 + 14*27 + 13*26 + 9*12 + 8*11 + 7*10$$

$$L_7 = 7 + 1*3 + 21*25 + 4*20 + 5*19 + 18*24 + 17*23 + 2*16 + 12*27 + 11*26 + 9*14 + 8*13 + 6*10$$

$$L_8 = 8 + 1*4 + 5*21 + 3*20 + 19*25 + 18*22 + 2*17 + 16*23 + 10*26 + 9*15 + 7*13 + 6*11$$

$$L_9 = 9 + 1*5 + 4*21 + 20*25 + 3*19 + 2*18 + 17*22 + 16*24 + 10*27 + 8*15 + 7*14 + 6*12$$

$$L_{10} = 10 + 2*3 + 5*24 + 4*23 + 22*25 + 18*19 + 17*20 + 1*16 + 12*14 + 11*13 + 9*27 + 8*26 + 6*7$$

$$L_{11} = 11 + 2*4 + 24*25 + 3*23 + 5*22 + 18*21 + 1*17 + 16*20 + 12*15 + 10*13 + 7*26 + 6*8$$

L_{12} = 12 + 2*5 + 3*24 + 23*25 + 4*22 + 1*18 + 17*21 + 16*19 + 11*15
+ 10*14 + 7*27 + 6*9

L_{13} = 13 + 3*4 + 5*25 + 2*23 + 22*24 + 1*20 + 19*21 + 16*17 + 14*15
+ 10*11 + 7*8 + 6*26

L_{14} = 14 + 3*5 + 4*25 + 2*24 + 22*23 + 20*21 + 1*19 + 16*18 + 13*15
+ 10*12 + 7*9 + 6*27

L_{15} = 15 + 4*5 + 26*27 + 3*25 + 23*24 + 2*22 + 1*21 + 19*20 + 17*18
+ 13*14 + 11*12 + 8*9

L_{16} = 16 + 5*27 + 4*26 + 14*18 + 13*17 + 12*19 + 11*20 + 1*10 + 9*24
+ 8*23 + 2*7 + 3*6

L_{17} = 17 + 3*26 + 25*27 + 15*18 + 13*16 + 13*21 + 1*11 + 10*20 + 9*22
+ 2*8 + 7*23 + 4*6

L_{18} = 18 + 3*27 + 25*26 + 15*17 + 14*16 + 1*12 + 11*21 + 10*19 + 2*9
+ 8*22 + 7*24 + 5*6

L_{19} = 19 + 2*27 + 22*26 + 15*20 + 1*14 + 13*21 + 12*16 + 10*18 + 3*9
+ 8*25 + 5*7 + 6*24

L_{20} = 20 + 2*26 + 22*27 + 15*19 + 14*21 + 1*13 + 11*26 + 10*17 + 9*25
+ 3*8 + 4*7 + 6*23

L_{21} = 21 + 24*26 + 23*27 + 1*15 + 14*20 + 13*19 + 12*17 + 11*8 + 4*9
+ 5*8 + 7*25 + 6*22

$$L_{22} = 22 + 20*27 + 19*26 + 2*15 + 14*23 + 13*24 + 4*12 + 5*11 + 10*25 + 9*17 + 8*18 + 6*21$$

$$L_{23} = 23 + 1*26 + 21*27 + 15*24 + 14*22 + 2*13 + 12*25 + 3*11 + 4*10 + 6*18 + 7*27 + 6*20$$

$$L_{24} = 24 + 1*27 + 21*26 + 15*23 + 2*14 + 13*22 + 3*12 + 11*25 + 5*10 + 9*16 + 7*18 + 6*19$$

$$L_{25} = 25 + 18*26 + 17*27 + 3*15 + 4*14 + 5*13 + 12*23 + 11*24 + 10*22 + 9*20 + 8*19 + 7*21$$

$$L_{26} = 26 + 1*23 + 21*24 + 2*20 + 19*22 + 18*25 + 3*17 + 4*16 + 15*27 + 8*10 + 7*11 + 6*13$$

$$L_{27} = 27 + 1*24 + 21*23 + 20*22 + 2*19 + 3*18 + 17*25 + 5*16 + 15*26 + 9*10 + 7*12 + 6*14 \qquad 4\text{-}2$$

Table 5 contains the effects for the contrasts listed in Equation 4-2. The effect is the ratio of the gradient and range of the test variable or contrast.

4.2.2.2 A Scenario of Explanation. In Table 5 note that the contrast for L_{26} has the greatest magnitude, 32.75 for productivity. That is at least two times greater than any other contrasts. The others which rank high in magnitude are 6, 13, 27, and 24, respectively.

The contrasts ranked highest under availability are 19, 21, 7, 8, and 27, respectively. The only common contrast among those ranked high for productivity and availability is L_{27}. Although L_7 is not among those ranked highest for productivity, its value should not be

Table 5. Main Effects and Aliased Two Factor Interactions

Contrasts	Response	
	Productivity	Availability
L_1	7.875	-0.0182
L_2	-0.375	-0.1319
L_3	1.250	0.0512
L_4	-1.750	0.0653
L_5	4.000	0.0164
L_6	16.250	0.1572
L_7	9.625	0.1883
L_8	1.375	0.1854
L_9	-9.812	0.0227
L_{10}	-2.500	-0.1484
L_{11}	4.625	-0.1398
L_{12}	-10.000	0.0588
L_{13}	16.125	0.0666
L_{14}	10.625	-0.0236
L_{15}	2.125	0.0184
L_{16}	-5.125	-0.0948
L_{17}	-2.125	-0.1017
L_{18}	3.000	-0.0618
L_{19}	1.500	0.6618*
L_{20}	1.000	-0.0453
L_{21}	-2.250	-0.2783
L_{22}	1.625	0.0192

Table 5. Continued

Contrasts	Response	
	Productivity	Availability
L_{23}	-5.625	-0.1046
L_{24}	11.375	0.0623
L_{25}	-1.000	0.0221
L_{26}	32.750*	0.0209
L_{27}	15.250	0.1737

overlooked. The same rationale applies to L_6 for availability.

The contrast exerting the least influence on productivity is variable 2, production time of the second subsystem in the first flowline, A2, and its aliased two factor interactions. Contrast L_1 exerts the least influence on availability.

It is not to be inferred or claimed at this point that any one or more variables exert more influence than the others. Neither is it to be implied that any given two factor interaction has more effect on productivity or availability than any other. None of these claims can be made because while the effect associated with one contrast in Table 5 is low, there could be two or more interactions or the main variable with large magnitudes. Since the contrast is a linear combination and effects may be negative or positive, one interaction could be large and positive added to an interaction which is large and negative, thus cancelling the large magnitudes and the contrast appears small.

4.3 The Second Experiment

The first experiment yielded the estimates of effects listed in Table 5. However, in an experiment with 27 variables, the number of two factor interactions is too great to make any definite conclusive statements on the factor or factors causing a given estimate to have a certain magnitude. To untangle the main effects requires a second fraction. The second fraction is a mirror image of the first fraction. The levels of the variables for the experiment were obtained by multiplying the entire matrix used for the first fraction by (-1) (see Table 6).

Table 6. Design Matrix for Second Fraction

Runs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	+	+	+	+	+	+	+	+	+	+	-	-
2	-	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	+	+	+	+	+	+
3	+	-	+	+	+	+	-	-	-	+	+	+	-	-	-	-	-	-	+	+	+	-	-	-	+	+	+
4	-	-	+	+	+	-	+	+	+	+	+	+	-	-	-	+	+	+	-	-	-	-	-	-	+	-	-
5	+	+	-	+	+	-	+	-	-	+	-	-	+	+	-	-	+	+	-	-	+	+	-	-	-	+	+
6	-	+	-	+	+	+	-	+	+	+	-	-	+	+	-	+	-	-	+	+	-	+	-	-	-	-	-
7	+	-	-	+	+	+	+	-	-	-	+	+	+	+	-	+	-	-	-	-	+	-	+	+	-	-	-
8	-	-	-	+	+	-	-	+	+	-	+	+	+	+	-	-	+	+	+	+	-	-	+	+	-	+	+
9	+	+	+	-	+	-	-	+	-	-	+	-	+	-	+	+	-	+	+	-	-	-	-	+	-	+	-
10	-	+	+	-	+	+	+	-	+	-	+	-	+	-	+	-	+	-	-	+	+	-	-	+	-	-	+
11	+	-	+	-	+	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	-	+	+	-	-	-	+
12	-	-	+	-	+	-	+	-	+	+	-	+	+	-	+	+	-	+	-	+	+	+	+	-	-	+	-
13	+	+	-	-	+	-	+	+	-	+	+	-	-	+	+	-	-	+	-	+	-	-	+	-	+	-	+
14	-	+	-	-	+	+	-	-	+	+	+	-	-	+	+	+	+	-	+	-	+	-	+	-	+	+	-
15	+	-	-	-	+	+	+	+	-	-	-	+	-	+	+	+	+	-	-	+	-	+	-	+	+	+	-
16	-	-	-	-	+	-	-	-	+	-	-	+	-	+	+	-	-	+	+	-	+	+	-	+	+	-	+
17	+	+	+	+	-	-	-	-	+	-	-	+	-	+	+	+	+	-	-	+	-	-	+	-	-	-	+
18	-	+	+	+	-	+	+	+	-	-	-	+	-	+	+	-	-	+	+	-	+	-	+	-	-	+	-
19	+	-	+	+	-	+	-	-	+	+	+	-	-	+	+	-	-	+	-	+	-	+	-	+	-	+	-
20	-	-	+	+	-	-	+	+	-	+	+	-	-	+	+	+	+	-	+	-	+	+	-	+	-	-	+
21	+	+	-	+	-	-	+	-	+	+	-	+	+	-	+	-	+	-	+	-	-	-	-	+	+	+	-
22	-	+	-	+	-	+	-	+	-	+	-	+	+	-	+	+	-	+	-	+	+	-	-	+	+	-	+
23	+	-	-	+	-	+	+	-	+	-	+	-	+	-	+	+	-	+	+	-	-	+	+	-	+	-	+
24	-	-	-	+	-	-	-	+	-	-	+	-	+	-	+	-	+	-	-	+	+	+	+	-	+	+	-
25	+	+	+	-	-	-	-	+	+	-	+	+	+	+	-	+	-	-	-	-	+	+	-	-	+	+	+
26	-	+	+	-	-	+	+	-	-	-	+	+	+	+	-	-	+	+	+	+	-	+	-	-	+	-	-
27	+	-	+	-	-	+	-	+	+	+	-	-	+	+	-	-	+	+	-	-	+	-	+	+	+	-	-
28	-	-	+	-	-	-	+	-	-	+	-	-	+	+	-	+	-	-	+	+	-	-	+	+	+	+	+
29	+	+	-	-	-	-	+	+	+	+	+	+	-	-	-	-	-	-	+	+	+	+	+	+	-	-	-
30	-	+	-	-	-	+	-	-	-	+	+	+	-	-	-	+	+	+	-	-	-	+	+	+	-	+	+
31	+	-	-	-	-	+	+	+	+	-	-	-	-	-	-	+	+	+	+	+	+	-	-	-	-	+	+
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

4.3.1 The Responses

The level of productivity ranges from 0 to 86, Table 7. Again, four computer simulated production runs yield high levels of productivity. The absence of regular repair teams and no periodic maintenance in at least one of the last subsystems of the two flowlines yield a zero or near zero response for productivity.

In the cases where system availability is zero, system down criterion is one subsystem down. Those runs with the criteria that all subsystems are down have high system availability, as a rule. The references to availability are dealing with the system availability resulting from the levels for subsystem and system down criteria.

4.3.2 Unaliased Estimate of Main Effects

The purpose for conducting the second experiment was to provide sufficient data points to isolate the extimates of main effects. The estimates in the first experiment corresponded to the aliases in Equation 2. The second experiment was the complete opposite of the first. Therefore, the aliasing structure is the same with one exception. The structure now becomes the main effect minus the two factor interactions, again assuming all three factor and higher interactions are negligible. By averaging the sum of the two corresponding estimates, the results are the unaliased estimates of the main effects.

4.3.2.1 On Productivity. Table 8 contains the main effects separated from the aliased two factor interactions for productivity. In the combined effects, Table 5, L_{26} has the greatest magnitude, 32.75. Now it becomes evident that variable 26--repair teams, 33.66--has some definite influence on the magnitude of L_{26}.

Table 7. Responses from Second Fraction

Run Number	Production	Percent Availability	Run Number	Production	Percent Availability
1	0	0	17	0	0
2	86	71.6	18	76	99.7
3	15	99	19	37	6
4	15	2	20	0	96
5	36	27	21	16	99.8
6	1	93	22	1	0
7	0	0	23	2	98.7
8	81	97	24	38	40
9	21	100	25	50	1.5
10	1	0	26	1	91
11	4	89.5	27	0	0
12	26	13	28	2	98.8
13	5	0	29	0	0
14	8	99	30	20	7
15	52	16	31	49	100
16	0	88	32	32	0

Table 8. Results After Separating the Main Effects and the Aliased Two Factor Interactions for Productivity

Variable	Main Effects	Aliased Two Factor Interactions
Repair Teams	33.656	-0.901
Repair Rate A2	-10.780	8.281
Failure Rate B2	9.594	-8.219
Preloading	8.469	6.781
Subsystem Down Criteria A2	-8.156	3.032
Failure Rate A2	7.719	8.531
Repair Rate B2	7.156	-2.844
Failure Rate B1	6.594	3.031
Loadsharing	5.968	4.656
Buffer Capacity	4.532	11.594
Equipment Sharing A	4.468	6.906
Subsystem Down Criteria B2	3.594	-0.594
Periodic Maintenance A2	-3.531	1.281
Equipment Sharing B	-3.469	2.469
Subsystem Down Criteria A1	-3.468	5.594
Production Time B2	3.219	-4.969
System Down Criteria	-3.031	4.531
Failure Rate A1	2.906	1.094
Repair Rate A1	-2812	-7.000
Periodic Maintenance B2	-2.218	-3.406
Subsystem Down Criteria B1	-2.156	0.031
Periodic Maintenance B1	1.718	-0.094
Periodic Maintenance A1	-1.344	2.344
Production Time A2	-1.094	0.719
Production Time A1	0.844	7.032
Production Time B1	0.469	0.781
Repair Rate B1	-0.344	4.969

However, the ranking does not continue exactly the same for main effects as it did in the combined effects. The other variables ranking high are (1) repair rate for A2, (2) failure rate of the second subsystem in the second flowline, B2, (3) preloading policy, (4) subsystem down criteria for A2, (5) failure rate of A2, and (6) repair rate for B2. The listings of variables ranked high and the contrasts for productivity have variables 6--failure rate of A2, and 25--preloading, in common. Note that the main effect from variable 13, buffer capacity, is less than half of what its magnitude was when included with its aliased two factor interactions. However, in column 2, the magnitude of aliased two factor interactions of variable 13, 11.6, is higher than any others in that column.

Referring to Equation 3-2, observe some of the interactions in L_{13} which could exert influence causing it to have the greatest magnitude of the two factor interactions. For this purpose, L_{13} is isolated below:

$$L_{13} = 13 + 3*4 + 5*25 + 2*23 + 22*24 + 1*20 + 19*21 + 16*17 + 14*15 + 10*11 + 7*8 + 6*26.$$

The interaction of variables 6, failure rate of A2, and 26, repair teams, is the only one where both variables appear in the listing of variables with high magnitude.

4.3.2.2 On Availability. The variable ranked highest for the estimates of main effects on availability is 19, system down criteria, Table 9. System down criteria represents changes in the system's configuration. The effects from variable 19, 60.52×10^{-2}, are nearly

Table 9. Results After Separating the Main Effects and the Aliased Two Factor Interactions for Availability

Variable	Main Effects	Effects from Aliased Two Factor Interactions
System Down Criteria	60.52	-0.20
Periodic Maintenance A2	-20.43	-7.40
Repair Teams	19.55	1.35
Subsystem Down Criteria B1	-14.61	4.43
Production Time A2	-14.54	1.36
Repair Rate A2	-12.84	-2.00
Failure Rate B1	12.70	6.13
Preloading	12.26	5.10
Production Time A1	-12.11	10.29
Periodic Maintenance B2	-11.64	1.18
Failure Rate A2	11.22	4.50
Subsystem Down Criteria A2	-10.48	1.00
Failure Rate B2	9.92	8.61
Subsystem Down Criteria B2	-8.55	2.37
Repair Rate B2	-8.28	20.04
Production Time B1	5.61	0.48
Repair Rate B1	-5.46	-8.52
Equipment Sharing A	-5.42	11.65
Buffer Capacity	5.36	1.30
Production Time B2	4.08	2.45
Periodic Maintenance B1	-3.94	5.68
Subsystem Down Criteria A1	2.72	-0.89
Repair Rate A1	-1.82	4.08
Periodic Maintenance A1	1.68	-6.22
Load Sharing	-1.42	-0.94
Failure Rate A1	-0.54	2.18
Equipment Sharing B	0.40	1.82

three times greater in magnitude than the effect corresponding to any other variable.

Based on the model and system configuration used in the experiment, system availability is increased if system down criteria is at its high level. System down driteria at a high level means a system configuration, such that all subsystems must fail before the system is down.

Table 9 also shows that on the average, it is better to have periodic maintenance, as indicated by a negative sign for the effects. Whereas that logic hampers productivity, it helps availability. Since the presence of repair teams is a variable, periodic repair can somewhat compensate for the absence of repair teams.

Variable 21, preventive maintenance interval for A2 ranks second followed by variable 26, repair teams. Thus, repair teams appears prominent in the ranking for both productivity and availability. Repair teams is followed on the list by 17--subsystem down criteria for B1, 2--production time for A2, 10--repair rate for A2, and 7--failure rate of B1. Of the first eight variables on either list, variables 26, 10, and 7 appear on each.

Although only eight variables are discussed as ranking high under availability, seven others are ranked very close to the variables ranked sixth, seventh, and eighth. From those additional seven, four rank in the top eight for productivity. Table 10 gives a comparison of the variables with the greater magnitudes for productivity and availability. For example, repair teams ranks high for both productivity and availability, while system down criteria only ranks high

Table 10. Comparison of Top Ranked Variables for Productivity and Availability

Variable	Productivity	Availability
Repair Teams (26)	Yes	Yes
System Down Criteria (19)	No	Yes
Repair Rate for A2 (10)	Yes	Yes
Failure Rate of B2 (8)	Yes	Yes
Preloading (27)	Yes	Yes
Subsystem Down Criteria A2 (16)	Yes	Yes
Failure Rate of A2 (6)	Yes	Yes
Repair Rate for B2 (12)	Yes	Yes
Failure Rate of B1 (7)	Yes	Yes
Preventive Maintenance for A2 (21)	No	Yes
Subsystem Down Criteria B1 (17)	No	Yes
Production Time A2 (2)	No	Yes
Production Time A1 (1)	No	Yes
Preventive Maintenance B2 (23)	No	Yes

for availability.

However, the table has nothing to do with the significance of the variables listed. It does show by the commonality of variables, which rank high in magnitude of effects. Note that all the variables listed rank high for availability, while less than half rank high for productivity. The size of this list changes when a measure of significance is assessed.

4.4 The Third Experiment

The main effects are listed by priority to indicate their dominance. However, some of the two factor interactions may possess greater dominance than some or all of the main effects. In an effort to minimize the number of two factor interactions, all the combinations of variables which cannot physically interact are eliminated. For instance, equipment sharing on flowline A does not interact with the repair rate of a subsystem on flowline B. As a result of the process of elimination, some contrast previously having twelve two factor interactions now have five, while in some contrasts no two factor interactions could be eliminated.

Variable 26 has the greatest main effect on productivity, therefore, observe the variables with which it interacts in the different contrasts. First considering the other variables which ranked high in productivity, L_8, L_{10}, L_{13}, L_4, and L_{15} have 26 interacting with 10, 8, 6, 16, and 27, respectively.

The magnitudes of the two factor interactions of the above mentioned contrasts, in nearly all cases, appear among those highly ranked.

The same type analogy was done with availability. This time

observe both the interactions involving 26 and 19. The contrast L_{12} has 16*19; L_{24} has 6*19 and 21*26; L_{21} has 10*19; L_8 has 10*26; and L_{11} has 7*26.

As a result of the prominence of variable 26 in the aliased two factor interactions with the greater magnitudes, the estimates of its two factor interactions are untangled. To untangle the estimates of variable 26 and its two factor interactions requires the third set of experiments. The design for the third experiment is the same as the first experiment with one exception--the levels for variable 26 are reversed in each run. The responses of the third experiment are listed in Table 11. Those runs in experiment one which had very high production levels now have much lower levels. For example, in the first experiment runs 4, 17, 29, and 32 have production levels of 89, 77, 82, and 75, respectively. While in the third experiment those runs have production levels of 0, 0, 46, and 1, respectively (Table 11).

4.5 A Model of Productivity and Availability

The first experiment is modeled for the response, y_i, with

$$y_i = \beta_0 + \sum_{i=1}^{27} \beta_i x_i + \sum_{\substack{i,j=1 \\ i \neq j}}^{27} \beta_{ij} x_i x_j + e_i \quad .$$

The terms β_i are the least square coefficients and represent the gradient per unit of design. The relationship between β_i and the effects, E_i is

$$E_i = 2\beta_i \quad , \quad i \neq 0$$

Table 11. Responses from the Third Fraction

Run Number	Productivity	Availability (in %)	Run Number	Productivity	Availability (in %)
1	62	1.96	17	0	99.14
2	29	98.10	18	2	0
3	12	28.77	19	83	99.97
4	0	0	20	0	0
5	28	99.12	21	51	47.8
6	0	0	22	0	55.60
7	2	96.44	23	0	0
8	73	44.21	24	15	98.80
9	34	6.25	25	9	100
10	0	86.38	26	0	0
11	0	0	27	3	90.20
12	11	99.85	28	33	14.72
13	0	41.19	29	46	16.94
14	33	19.98	30	52	99.93
15	43	96.73	31	0	0
16	1	0	32	1	0

and

$$E_i = \beta_i \quad , \quad i=0 \quad .$$

Given the responses from the first fraction, y_i, then β_0, the mean, and the term $(\beta_i + \beta_{ij})$ are computed.

The sum of the first and second fractions is one fraction of a 2^{27-21} design. The model of the fraction resulting from combining the first and second fractions:

$$y_i = \beta_0 + \sum_{i=1}^{27} \beta_i x_i + e_i \quad . \qquad 4\text{-}3$$

The sum of the first and third fractions give rise to the model

$$y_i = \beta_0 + \beta_{26} x_{26} + \sum_{\substack{i=1 \\ i\neq 26}}^{27} \beta_{26i} x_i x_{26} + e_i \quad . \qquad 4\text{-}4$$

The term $\sum_{\substack{i=1 \\ i\neq 26}}^{27} \beta_{i26} x_i x_{26} + e_i$ from Equation 4-4, is included in the e_i term of Equation 4-3. Thus upon completion of the third experiment, instead of having one fraction of a 2^{27-21}, the sum of the three experiments is 96 runs. Normally, the combination of two fractions, each from 2^{27-21} designs, gives rise to one fraction of a 2^{27-20} design. However, each of the fractions has one set of experiments in common.

This results in a new family, because the third fraction of a 2^{27-22} design is combined with the first and second fractions, a 2^{27-21} design. Box and Hunter address combining fractionals not of

the same family with the comment, "Possibilities arising from designs of this sort are presently being investigated" (4). Investigations have not yet revealed the solution to this problem.

This research does not have as one of its objectives to solve the problem of combining fractionals not of the same family. However, to overcome the problem in this case, three steps were taken to estimate the parameters for the postulated model

$$y_i = \beta_0 + \beta_{26}x_{26} + \sum_{\substack{i=1 \\ i\neq 26}}^{27} \beta_{i26}x_i x_{26} + e_i \quad . \qquad 4\text{-}5$$

As given information, we know that:

$$y_i - (\beta_0 + \sum_{i=1}^{27} \beta_i x_i) = \sum_{\substack{i=1 \\ i\neq 26}}^{27} \beta_{i26}x_i x_{26} + e_i \quad ,$$

where

$$y_i - (\beta_0 + \sum_{i=1}^{27} \beta_i x_i) = e_i \quad . \qquad 4\text{-}6$$

The e_i in Equation 3-6 is the same e_i from Equation 4-3.

Step 1: Calculate e_i's for expressions in the first fraction using:

$$\hat{y}_i = \hat{\beta}_0 + \sum_{i=1}^{27} \hat{\beta}i x_i$$

and

$$y_i - \hat{y}_i = e_i \quad .$$

Step 2: Using the same equations as Step 1, calculate e_i's for expressions in the third fraction.

Step 3: Using the e_i's from Steps 1 and 2, estimate the parameters in Equation 4-5.

The results of the above listed procedure are listed in Table 12 for productivity. The most outstanding interaction is repair teams with preventive maintenance for A2, whose effect is 9.14.

Table 12. Estimates of Variable 26 and Its Two Factor Interactions for Productivity

Interaction	Effect	Interaction	Effect
21*26	9.14	18*26	-1.10
23*26	3.64	3*26	1.00
24*26	-3.58	20*26	.96
7*26	2.62	2*26	.86
26	-2.53		
19*26	2.49	8*26	.65
13*26	2.44	26*27	-.64
6*26	2.35	16*26	-.60
4*26	-2.32	1*26	.40
25*26	1.98	11*26	.36
10*26	-1.48		
15*26	1.47		
22*26	1.35		
17*26	-1.11		

The second and third are the interactions of 26 with preventive maintenance for B2 and equipment sharing on flowline A, respectively. The magnitude of those effects are less than half the magnitude of 21*26.

4.5.1 Significance of Main Effects

To establish a variable as being significant an ANOVA is done and the computed statistic, $\hat{F}$ is compared to the test statistic, F, at

.05 and .1 α-levels. The observations from each of the three experiments are used affording a total of 96 degrees of freedom. The regression sum of squares, SSQ, is defined as:

$$SSQ = b'x'y,$$

where $$b = (x'x)^{-1}x'y \ .$$

There is one degree of freedom for each SSQ associated with each term in the model, except the variable 26, repair teams, which utilizes 2 degrees of freedom. The first estimate of effect for repair teams comes from the model in Equation 3-3. The coefficient for variable 26, after combining the first and third experiments, is small because the coefficient has already been subtracted out of the e_i term. The coefficient was first subtracted out when the main effects were untangled, accounting for one degree of freedom. For all practical purposes, the second estimate for the coefficient is for the most part associated with pure error.

4.5.1.1 ANOVA on Estimates for Productivity. The main effects appear in the same order on the ANOVA table, Table 13, as in Table 8. Using the F values from statistical tables, all the variables down to preloading are significant at the .05 level. The list extends down to and including repair rate of B2 at the .1 level. The interaction of repair teams and preventive maintenance for A2 is significant at the .05 level.

The designation of certain variables as being significant does not label the remaining variables as insignificant. However, those variables which are qualitative--indicating absence or presence--and

Table 13. ANOVA on Variables for Productivity

Sources	SSQ	df	MSSQ	$\hat{F}$
Repair Teams (26)	18035.2	2	9017.6	32.8*
Repair Rate of A2 (10)	1859.3	1	1859.3	6.8*
Failure Rate of B2 (8)	1472.7	1	1472.7	5.36*
Preloading Policy (27)	1147.6	1	1147.6	4.2*
Subsystem Down Criteria for A2 (16)	1064.3	1	1064.3	3.9**
Failure Rate of A1 (5)	953.3	1	953.3	3.47**
Repair Rate of B2 (12)	819.32	1	819.32	2.9**
Failure Rate of B1 (7)	695.7	1	695.7	2.53
Load Sharing Policy (14)	568.9	1	569.9	2.1
Buffer Capacity (13)	328.6	1	328.6	1.2
Equipment Sharing Flowline A (24)	319.4	1	319.4	1.2
Subsystem Down Criteria for B2 (18)	206.7	1	206.7	.8
Periodic Maintenance for A2 (21)	199.5	1	199.5	.7
Equipment Sharing Flowline B (25)	192.6	1	192.6	.7
Subsystem Down Criteria for A1 (15)	192.4	1	192.4	.7
Production Time for B2 (4)	165.8	1	165.8	.6
System Down Criteria (19)	147.0	1	147.0	.5
Failure Rate of A1 (5)	135.1	1	135.1	.5
Repair Rate of A1 (9)	126.5	1	126.5	.5
Periodic Maintenance for B2 (23)	78.7	1	78.7	.3
Subsystem Down Criteria for B1 (17)	74.4	1	74.4	.3
Periodic Maintenance for A2 (22)	47.2	1	47.2	.2
Periodic Maintenance for A1 (20)	28.9	1	28.9	.1
Production Time for A2 (2)	19.1	1	19.1	.07
Production Time for A1 (1)	11.2	1	11.2	.04
Production Time for B1 (3)	3.52	1	3.52	.01
Repair Rate of B1 (11)	1.9	1	1.9	.01
21*26	1336.6	1	1336.6	4.9*
23*26	212.2	1	212.2	.8
24*26	205.1	1	205.1	.7
7*26	112.4	1	112.4	.4
19*26	99.0	1	99.0	.4
13*26	95.3	1	95.3	.3
6*26	88.4	1	88.4	.3
4*26	86.1	1	86.1	.3
25*26	63.0	1	63.0	.2
10*26	35.0	1	35.0	.1
15*26	34.4	1	34.4	.1
22*26	29.2	1	29.2	.1
17*26	19.7	1	19.7	<.1
18*26	19.5	1	19.5	<.1
3*26	16.1	1	16.1	<.1
20*26	14.7	1	14.7	<.1

Table 13. Continued

Sources	SSQ	df	MSSQ	$\hat{F}$
2*26	11.8	1	11.8	<.1
8*26	6.8	1	6.8	<.1
26*27	6.5	1	6.5	<.1
16*26	5.7	1	5.7	<.1
1*26	2.6	1	2.6	<.1
11*26	2.1	1	2.1	<.1
Residual	12363.3	45	274.74	

Table Values: $F_{1,45,.05} = 4.06$, $F_{2,45,.05} = 3.2$, $F_{1,45,.1} = 2.8$

*Significant at .05 level
**Significant at .1 level

are not dominate will not likely play a major role. Those variables designated significant will not change to insignificant as the estimates of the other two factor interactions are untangled from the residual error. Thus, repair teams, the rate of repair on A2, failure rate of B2, and the preloading policy are definitely significant at an α-level .05. At an α-level of .1, subsystem down criteria for A2, failure rate of A1, and repair rate of B2 are significant.

Those variables with $\hat{F}$ values approaching the table value of F soon become significant as other two factor interactions with high valued effects are untangled from the residual. Others which show a probable sign that they will become significant are failure rate of B1 and the system's load sharing policy.

4.5.1.2 ANOVA on Estimates for Availability. The availability of the system has more variables appearing significant than productivity. The $\hat{F}$ value for system down criteria is 144.48 compared to the table's F value of 4.05 at an α-level of .05 (see Table 14). Twelve of the 27 variables are significant at the .05 α-level. Some of the variables of the 12 are especially noteworthy, among them are: (1) repair teams, (2) preloading policy, (3) all aspects of operation on A2, and (4) failure rate and periodic maintenance for B1. At the .1 level, failure rate, subsystem down criteria and repair rate, all for B2, become significant, thus raising the total to 14 of 27 variables being significant.

As was the case with productivity, the exclusion of any variables from being significant does not automatically designate those variables as insignificant. The same situation exists where the residual sum of

Table 14. ANOVA on Variables for Availability

Sources	SSQ	df	MSSQ	$\hat{F}$
System Down Criteria (19)	58602.7	1	58602.7	144.48
Periodic Maintenance for A2 (21)	6678.2	1	6678.2	16.46*
Repair Teams (26)	6106.3	2	3053.2	7.53*
Subsystem Down Criteria for B1 (17)	3415.2	1	3415.2	8.42*
Production Time for A2 (2)	3382.6	1	3382.6	8.34*
Repair Rate of A2 (10)	2637.9	1	2637.9	6.50*
Failure Rate of B1 (7)	2580.6	1	2580.6	6.36*
Preloading Policy (27)	2404.9	1	2404.9	5.93*
Production Time for A1 (1)	2346.4	1	2346.4	5.78*
Periodic Maintenance for B2 (23)	2167.8	1	2167.8	5.34*
Failure Rate of A2 (6)	2014.2	1	2014.2	4.97*
Subsystem Down Criteria for A2 (16)	1757.3	1	1757.3	4.33*
Failure Rate of B2 (8)	1574.5	1	1574.5	3.88**
Subsystem Down Criteria for Bs (18)	1169.6	1	1169.6	2.88**
Repair Rate of B2 (12)	1096.9	1	1096.9	2.7**
Production Time for B1 (3)	503.6	1	503.6	1.24
Repair Rate of B1 (11)	477.0	1	477.0	1.17
Equipment Sharing Flowline A (24)	470.0	1	470.0	1.16
Buffer Capacity (13)	459.7	1	459.7	1.13
Production Time for B2 (4)	266.3	1	266.3	.66
Periodic Maintenance for B1 (22)	248.4	1	248.4	.61
Subsystem Down Criteria for A1 (15)	118.4	1	118.4	..29
Repair Rate of A1 (9)	53.0	1	53.0	.13
Periodic Maintenance for A1 (20)	45.2	1	45.2	.11
Load Sharing Policy (14)	32.3	1	32.3	.08
Failure Rate of A1 (5)	4.7	1	4.7	<.1
Equipment Sharing Flowline B (25)	2.6	1	2.6	.1
23*26	723.6	1	723.6	1.78
6*26	411.6	1	411.6	1.01
22*26	319.1	1	319.1	.78
21*26	314.0	1	314.0	.77
16*26	290.4	1	290.4	.71
7*26	178.1	1	178.1	.43
19*26	155.7	1	155.7	.38
15*26	102.9	1	102.9	.25
24*26	61.7	1	61.7	.15
2*26	43.0	1	43.0	.10
1*26	42.8	1	42.8	.10
20*26	42.1	1	42.1	.10
17*26	38.9	1	38.9	.10
8*26	33.4	1	33.4	<.1
13*26	21.9	1	21.9	<.1
3*26	15.6	1	15.6	<.1

Table 14. Continued

Sources	SSQ	df	MSSQ	$\hat{F}$
10*26	13.2	1	13.2	<.1
18*26	8.6	1	8.6	<.1
4*26	4.9	1	4.9	<.1
25*26	2.9	1	2.9	<.1
Residual	19062.5	47	405.6	

Table Values: $F_{1,47,.05} = 4.05$, $F_{2,47,.05} = 3.2$, $F_{1,47,.1} = 2.8$

*Significant at .05 level
**Significant at .1 level

squares is large and contains the sum of squares for the remaining two factor interactions.

4.5.2 Maximizing Productivity and Availability

In this section, two techniques are used to maximize productivity and availability. The maximum level of one does not necessarily place the other at its maximum level. The first approach is steepest ascent; the second is branch and bound, using mixed-integer programming.

4.5.2.1 Steepest Ascent. The model to be maximized is a linear additive model,

$$y_i = \beta_0 + \sum_{i=1}^{27} \beta_i x_i \quad .$$

The model has 27 variables, 12 of which are quantitative and the remaining 15 are qualitative. The quantitative variables are: (1) four variables for subsystems' production times, (2) four variables for subsystems' failure rates, and (3) four variables for subsystems' repair rates. The qualitative variables are included in the steepest ascent at a constant level.

The production times, variables 1, 2, 3, and 4, are entered as quantitative variables although the high and low levels used in this experiment restrict it to two possible values. The time required to produce an item may vary, however, the system is designed on the concept of line balance. This constraint removes the possibility for these variables to be continuous. Therefore, production time must take on values which will yield an integer for the number of units in that subsystem.

The production variables were entered ignoring the constraint. From the maximum solution, the four variables are set to the nearest integer value.

The objective for productivity is to maximize production, where the maximum possible level is 99.

The model is now written:

$$\hat{y} = 21.4 + .4x_1 - .55x_2 + .24x_3 + 1.6x_4 + 1.45x_5 + 3.9x_6$$
$$+ 3.3x_7 + 4.8x_8 - 1.4x_9 - 5.4x_{10} - 1.7x_{11} - 3.6x_{12}$$
$$+ 2.3x_{13} + 3.0x_{14} - 1.7x_{15} - 4.1x_{16} - 1.1x_{17} + 1.8x_{18}$$
$$- 1.5x_{19} - .67x_{20} - 1.8x_{21} + .86x_{22} + 1.1x_{23} + 2.2x_{24}$$
$$- 1.7x_{25} + 16.8x_{26} + 4.24x_{27} \quad . \qquad 4\text{-}7$$

Now let

$$x_{15}, x_{16}, x_{17}, x_{19}, x_{20}, x_{21}, x_{25} = -1,$$

$$x_1, x_2, \ldots, x_{12} = 0,$$

and

$$x_{13}, x_{14}, x_{18}, x_{22}, x_{23}, x_{24}, x_{26}, x_{27} = 1 \quad .$$

Then

$$\hat{y} = 64.08 \ .$$

Table 15 shows the calculated yield at a given number of incremented units along the path to maximizing the model.

Although the calculated value exceeds the system capability, note that the levels in failure have been exceeded. In addition, recall that the first four variables can only be set at 100 or 200. Thus, accounting for the over value. The incremental units represent the points along the path of steepest ascent to maximizing productivity. The values under variablex x_1, x_2, x_3, and x_4, representing production times, are not values such that the number of units in a subsystem are integers. A fraction of a machine has no physical meaning. Therefore to substantiate the experimental results, the production times are set at the nearest value, in the direction to which the path is tending, such that production time yields an integer number of units.

The mean time to failure exceeds the test levels in three of the four subsystems at both the seventh and eighth unit increments. The high level for the test is 960 minutes, however, the mean time to failure for the second subsystem in the "B" flowline is 1340 minutes. The other subsystems exceeding the test level are A2 and B1 with values 1190 and 1090, respectively.

The configuration constraints are set at the levels specified at the seventh and eighth points along the path of steepest ascent. Computer simulation runs made at specified levels for seven and eight units verify the calculated system production levels of 97 and 99, respectively.

The objective for availability is to maximize availability very close or equal to 100 percent.

Table 15. Path of Steepest Ascent for Productivity

Path	Variable												Calculated Yield
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
Base	150	150	150	150	540	540	540	540	100	100	100	100	64.08
1 Unit	151	148.6	150.6	154	570	621	609	640	94	79	99	86	68.98
3 Units	153	145.8	152	162	631	784	746	840	83	86	98	57	78.89
5 Units	155	143.0	143	170	691	946	884	1040	72	7	96	28	90.69
7 Units	157	140.2	154.2	178	751	1108	1021	1240	61	0	94	20	96.18
8 Units	158	138.8	154.8	182	782	1190	1909	1340	66	0	93	6	102.06

The model for availability is:

$$y = 48.95 - 6.05x_1 - 7.27x_2 + 2.8x_3 + 2x_4 - .3x_5 + 5.6x_6$$
$$+ 6.4x_7 + 5.0x_8 - .9x_9 - 6.4x_{10} - 2.7x_{11} - 4.1x_{12} + 2.7x_{13}$$
$$- .7x_{14} + 1.4x_{15} - 5.2x_{16} - 7.3x_{17} - 4.3x_{18} + 30.3x_{19}$$
$$+ .8x_{20} - 10.2x_{21} - 2x_{22} - 5.8x_{23} - 2.7x_{24} + 2x_{25}$$
$$+ 9.8x_{26} + 6.1x_{27} \qquad 4\text{-}8$$

Now let

$$x_{14}, x_{23}, x_{24} = -1$$

$$x_1, x_2, \ldots, x_{12} = 0$$

all other x = 1

so $\hat{y} = 80.45.$

Thus the calculated system availability when x_1 through x_{12} are set at their base values is 80.45. Table 16 starts the iterative procedure at that point.

Although the values for production times, Table 16, are not values that yield integers when divided by 100, the levels used for a verification of calculated yield are those in the direction to which the production times tend. The values for all variables are within the range of the levels tested.

Table 16. Path of Steepest Ascent for Availability

Path	Variables												Calculated Yield
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	
Base	150	150	150	150	540	540	540	540	100	100	100	100	80.45
1 Unit	139	136	155	154	536	628	640	618	97	81	92	88	90.31
2 Units	127	1123	160	158	531	717	740	696	95	62	84	75	100.22

Using the results from Table 16, the system was simulated with the settings corresponding to the steepest ascent. The results of the computer simulation have system availability at 100 percent.

4.5.2.2 Branch and Bound Mixed-Integer Programming. The steepest ascent approach provides feasible solution for maximizing productivity. However, that solution is not necessarily the optimal solution. The optimal solution can have various cost constraints not considered by the steepest ascent. The solution from the branch and bound mixed-integer programming algorithm (BABMIP), does include the constraint considerations. The problem of maximizing productivity or availability is a combination of constrained integer valued variables--qualitative, and real valued variables--quantitative. The branch and bound mixed-integer programming algorithm provides an optimal solution.

The objective functions to be maximized for productivity and availability are the same as maximized using steepest ascent, Equations 4-7 and 4-8, respectively. The constraints on the real variables are such that they do not exceed physical limitations. For example, a repair rate of less than zero has no physical meaning. The same applies to production and failure rates.

In order to maximize productivity, using steepest ascent, the bounds on the variables are extended further from their base value than when the corresponding variables in the maximization of availability. Thus, in this model, productivity carries more weight than availability. Availability is used as a constraint in maximizing productivity with BABMIP. Availability is constrained at less than or equal to 100 percent.

The BABMIP algorithm does not accommodate the (-1,1) limitation of variable values without including zero. Therefore the low level, -1, is adjusted to zero for each of the qualitative variables. To adjust the value to zero, (1) each B_i is subtracted from the value of the objective function, and (2) the B_i's which are coefficients of the x_i's are multiplied by two. Thus if the value of x_i is 0 the value has already been included; if the value is one, then half is negated by the B_i already subtracted, thus only leaving the desired value.

As shown in Table 17, the value of the objective function is 99, the same as with steepest ascent. However, the values of x_i's are different. In the steepest ascent, each of the variables makes some positive contribution. However, the optimal solution from BABMIP shows it is not necessary to have a contribution from each configuration constraint. The production times change at each point along the path of steepest ascent. However, BABMIP has three of the subsystems making no contribution by changing their production times. Changes in configuration can be associated with cost. Thus, when cost is a consideration, it is usually desirable to maximize production at minimum cost.

To maximize productivity using the steepest ascent technique, each qualitative variable is at a level to make a positive contribution. The optimal solution using BABMIP has subsystem down criterion for A1, preventive maintenance policy for B2, and equipment sharing on flow-line A, making no contribution to maximizing productivity. The optimal solution lowers the mean time to failure from 1340 to 1190 for B2. The MTTF for A2 and B1 are also lower. The mean time to repair is lowered from 66 to 20 for A1; MTTR remains the same using BABMIP as

Table 17. Results from BABMIP Algorithm

Variable	Value	Variable	Value
x_1	0	x_{15}	0
x_2	0	x_{16}	1
x_3	0	x_{17}	1
x_4	.9	x_{18}	1
x_5	1.4	x_{19}	1
x_6	1.3	x_{20}	1
x_7	1	x_{21}	1
x_8	1.56	x_{22}	1
x_9	1	x_{23}	0
x_{10}	2	x_{24}	1
x_{11}	0	x_{25}	0
x_{12}	1.3	x_{26}	1
x_{13}	1	x_{27}	1
x_{14}	1		

it was using steepest ascent. The MTTR for B1 is 180 minutes, an increase of 87 over the level using steepest ascent.

4.6 The Transient and Steady-State Effects

The steady-state phenomenon of the reliability of a system is discussed, as pointed out in Chapter I, by nearly everyone working in the area of system reliability. The assumption that the system is operating under steady-state condition is one of the bases for the conclusions reached in previous studies. The systems which do not reach steady-state during a production run are not in the above category of effects during steady-state. In addition, given that the effects of the transient state are known, the planner has more upon which to base his decisions. Another facet of transient and steady-state is knowing which configuration constraints cause the transient phenomenon to be longer. The first look focuses on how the transient phenomenon affects the system based on batch size. That will be followed by the effects on productivity and availability.

4.6.1 A View Through Batch Size

Using the data from the first and second fractions, the time interval between production of each item is averaged. When all the runs are considered, the minimum production level is zero. A zero production level is present in 14 of the 64 experiments. The next step is to eliminate the conditions causing the zero production level. The absence of regular and periodic repair causes the system to remain down once it fails. Therefore, whenever failure occurs before one item is produced, the production level is zero. After eliminating the absence of both types of repair, the minimum batch size is eight.

From these, runs are sequentially eliminated such that batch size increases to the next higher minimum value.

Figures 19, 20, 21, and 22 graphically illustrate how the increase in batch size lessens the effect of the transient phenomenon on the rate of production. The x-axis indicates the ith item produced in a given batch size. The y-axis represents the ratio of one to the time interval between the (i-1) and the ith item. Although there may appear to be little numerical difference between .01 and .005, the difference represents 100 minutes production time. Each marked increment on the y-axis equals 100 minutes of production time.

Figure 19 consists of those runs with the production level greater than seven. The average time for the first item is 483 minutes. The time interval drops to a low of 252 minutes between the second and third items. Then the time continuously increases for the fourth, fifth, and sixth items, but declines for the seventh and eighth items. There is not a pattern established nor does the interval between items get closer. The irregularity and lack of a close interval are indicators that the transient phenomenon is present throughout.

Figure 20 represents the averaged time intervals of all runs with output greater than 24. Twenty-four of the 64 runs are in that category. The time for the first item is 379 minutes. The second and third items show decreases. The pattern is irregular up to the production of item 19. From 19 through 25, the interval of production is close and the pattern begins showing signs of stabilization. The pattern regularity and a near constant interval of production shows the process reaches steady-state on the production of item 19.

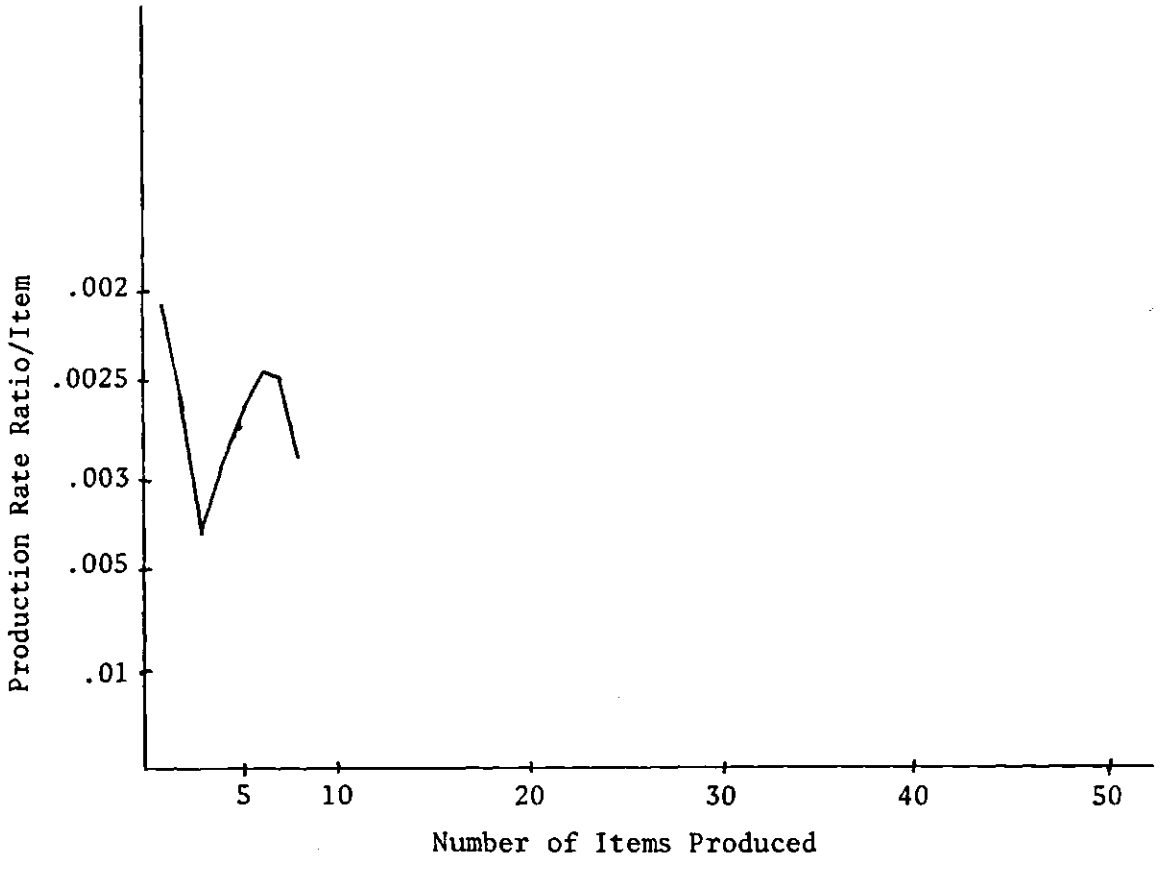


Figure 19. Average Production Interval--Batch Size 8

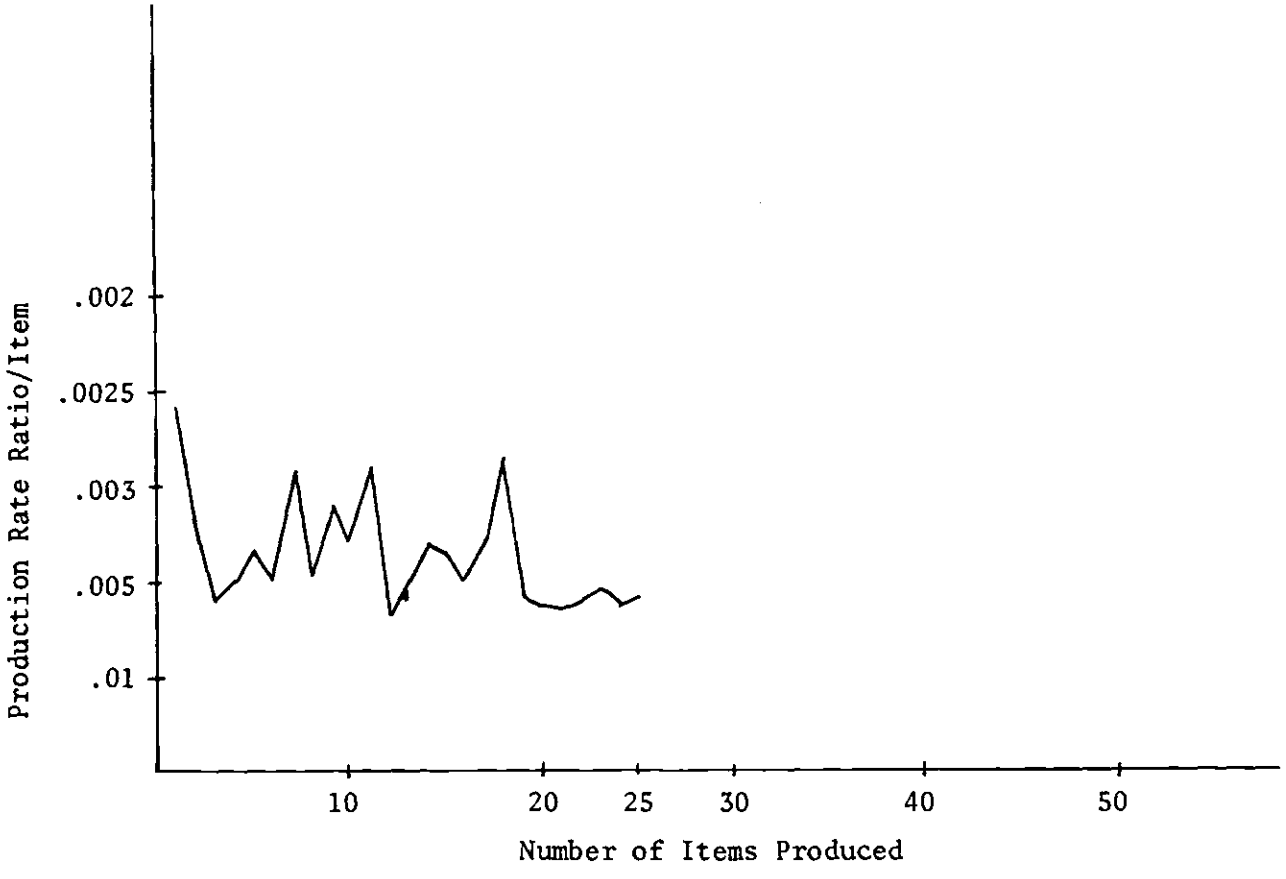


Figure 20. Average Production Interval--Batch Size 25

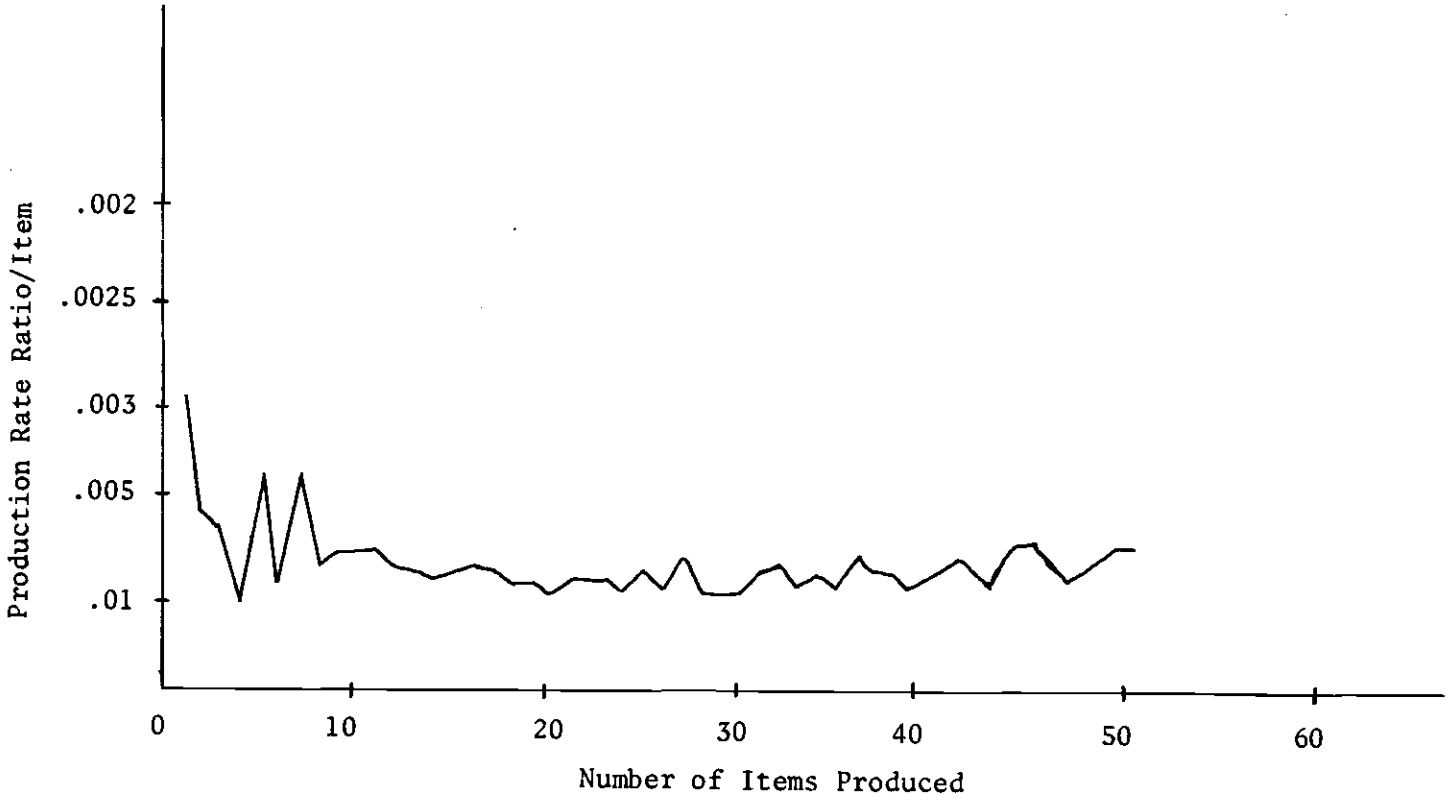


Figure 21. Average Production Interval--Batch Size 50

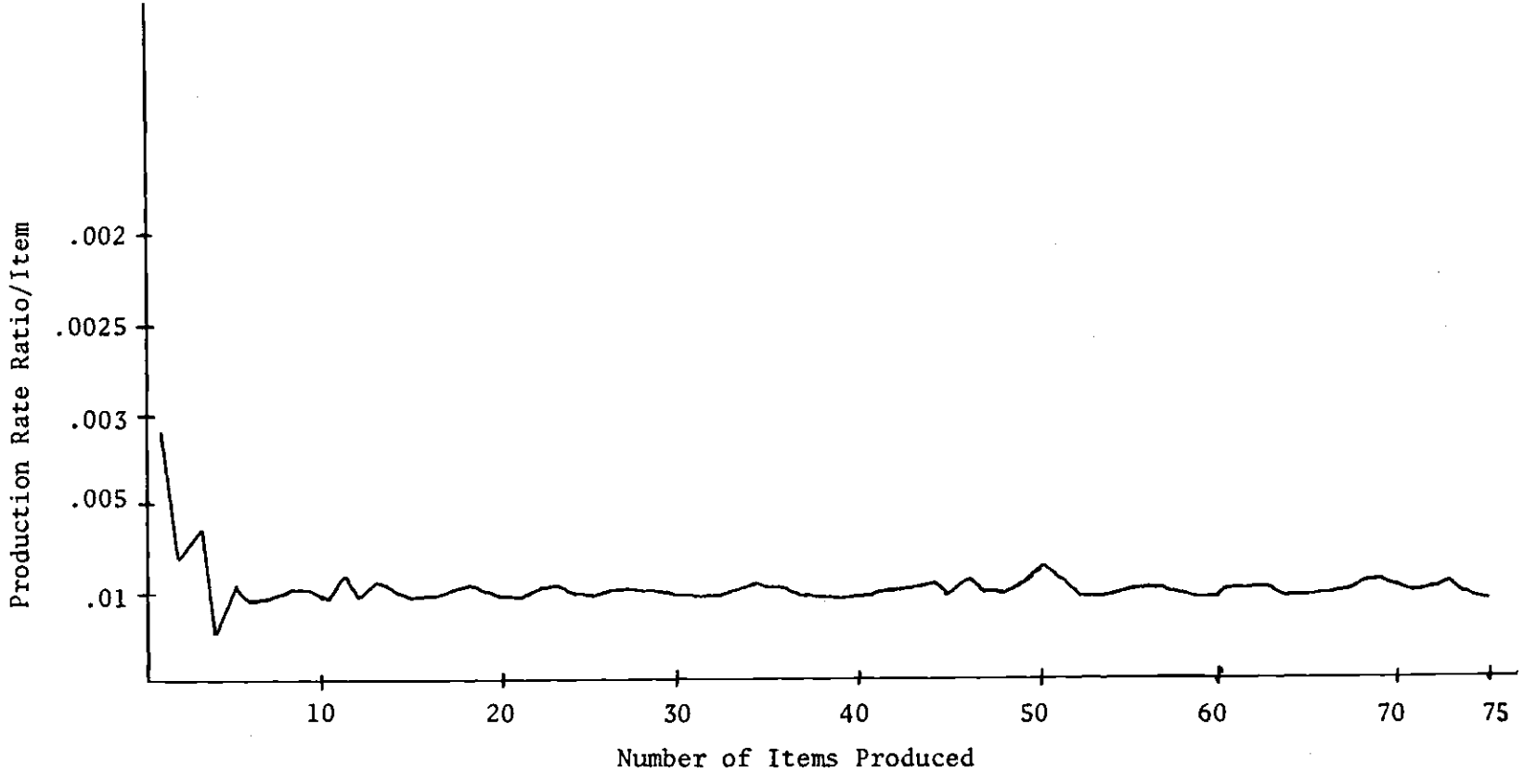


Figure 22. Average Production Interval--Batch Size 75

The production level for the runs included in Figure 21 is 50 items or more. The average time for the first item is 300 minutes. As the minimum batch size increases, the average time for the first item decreases. As in the previous two figures, the production interval decreases for the second and third items. With the minimum production level at 50, a regular pattern begins at the production of the eighth item. Steady-state is in effect eleven times sooner when the minimum batch size is 25.

In Figure 22 the batch size is 75. Again the production time decreases in comparison to the smaller batch sizes. The production time interval stabilizes after the fourth item. The production rate for maximum production is 100 minutes. At a production level of 75 the time interval begins to center around that time.

The differences in average production time on Figures 21 and 22 are also germane to the planner. For example, the production time for those items between 35 and 45 averages 140 minutes in Figure 12, while the average is 100 minutes for the same group of items in Figure 22. Although the runs for both groups have regular repair teams, the runs in Figure 9 are also preloaded with infinite buffer capacity.

Figure 23 shows the variables, which when present, allow the minimum batch size to be exceeded. Their absence in other cases is consistent with a less than minimum batch size. The minimum is the starting point for the line on which the variable appears. The variables listed act as constraints for the batch size to exceed the minimum levels. Each experiment is considered individually to establish which variables are constraining.

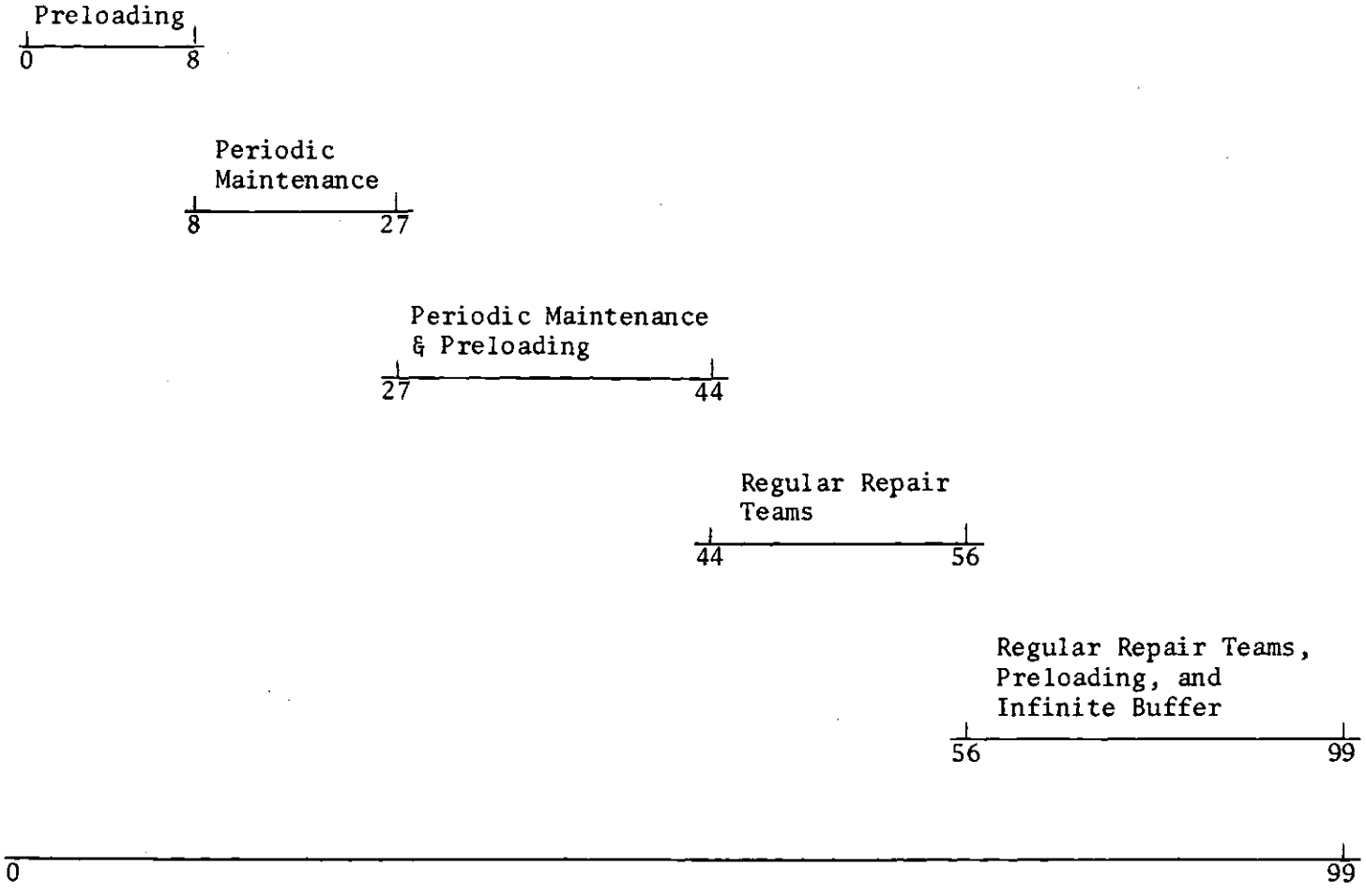


Figure 23. Batch Size Constraints

Those runs with a production level greater than zero and up to eight are preloaded. The level of the other variables vary during these runs. Whenever the batch size is greater than eight, but less than 27, periodic maintenance is consistently performed. The addition of preloading to periodic maintenance increases the range for production. The interval includes a low of 27 and a high of 44 items being produced. For batch sizes between one and 44, regular repair teams are not consistently absent or present. However, for all batch sizes greater than 44, repair teams are present. The production level ranges from 56 to 99. When the system is preloaded, buffer capacity is infinite and regular repair teams are present.

4.6.2 On Productivity and Availability

The production rate is irregular during the transient state. Examples of this can be seen in Figures 20, 21, and 22. In Figure 20, the production rate for the first item is 383 minutes; it drops to 180 for the third, then goes back up to 320 for the fourth. It continues to fluctuate in that manner until the nineteenth item is produced.

The runs with the long transient state depend on periodic maintenance without the presence of regular repair teams. The runs with the large transient time have combinations of short MTTF's and long repair times, or periodic maintenance without regular repair.

Those runs with a long transient state have low production levels over the time interval (0,10000) minutes, while those runs with a short transient state have high production levels, as shown in Figure 22. In the latter example, the transient state terminates with the production of item 4.

Steady-state has a smaller variation in production rate. The smaller variation leads to a smoother running system. In Figures 21 and 22 the variation is 50 and 30, respectively. The regularity of production rate allows system planning to be easier as a result of consistency. This consistency can assist in reducing cost because of the predictability of the system.

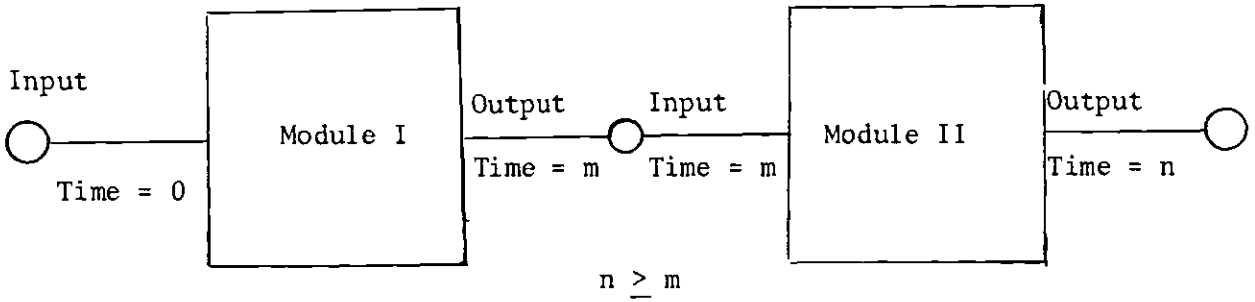
The longer the steady-state during the computer simulation runs, the higher the level of productivity. As shown in Figures 20, 21, and 22, the length of steady-state increases from Figure 20 to Figure 21 to Figure 22, and production level increases in the same manner.

4.7 Module Output to Module Input

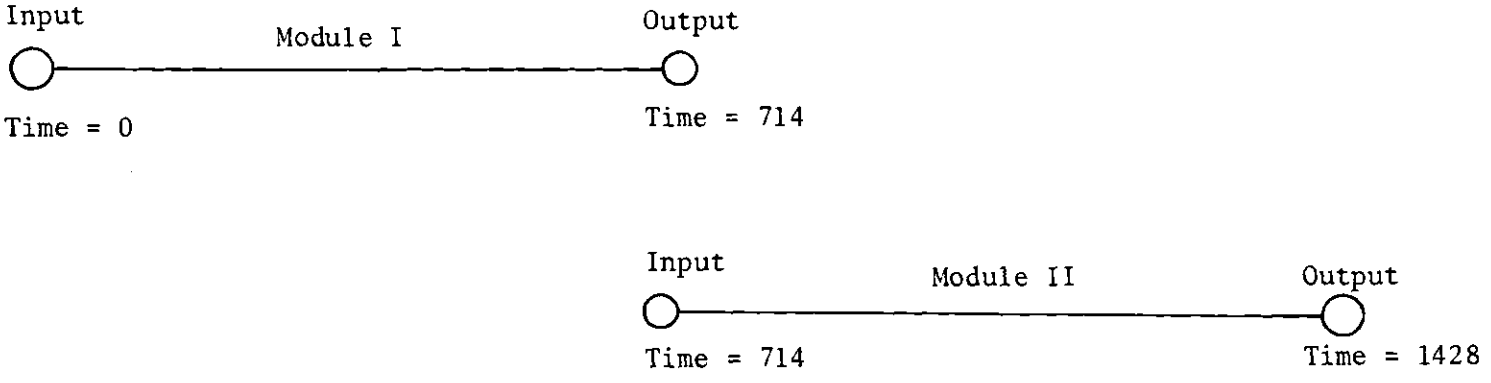
The modular concept, as pointed out in Section 3.3, is one approach to solving the problem of analyzing a complex system. As shown by Swanson (29), this approach works for operating a complex system.

Given that the system modeled in Chapter III is the first module for a complex system, the output from the first module becomes input for the second module, top drawing in Figure 24. The data from the first module was collected over the time interval (0,10000) in minutes. Since this time interval represents the system's run time, the second and all succeeding modules are observed over the same time interval.

The departure times, m, for items from the first module are used as input times, m, for the second. For the computer simulation to input the items at the proper times, time designated generators are used. The sequence which has the same output intervals is placed in the same generator.



A. Direct Output-Input Flow



B. Example of Output-Input

Figure 24. Module Interfacing

For example, the output from one module has several groups where items exit at approximately the same interval, then each group is separated into a generator. Given the first group starts at time 800, there are five items in that group, and the exit interval is 110 minutes. The generator for that group is the GPSS statement:

GENERATE 800,5 TIME (110).

This statement places the first item into the second module at time 800 and five of them will be generated, each at a 110 minute time increment.

The same applies to each group. The number of items in the group is dependent upon the uniformity of the output from the first module.

Using the output from the fourth and sixteenth runs of the first experiment, and having the second module with the same operating characteristics as the first module, the outputs from the first modules are 89 and 44 for runs four and sixteen, respectively. When interfaced with the second module, the outputs are 89 and 23 for runs four and sixteen, respectively.

The output from the second module is the same as the first module on run four because of two variables--preloading and infinite buffer capacity. The second example does not have either of these conditions.

In the latter example, bottom drawing in Figure 24, the first item enters the second module at a clocktime of 714 minutes, whereas in the first module the first item enters at time 0. Therefore, the

earliest time an item can be expected to depart the second module is at time 914. But, the second module has the same operating characteristics as the first. It will encounter the same transient effect as the first. The transient effect starts at 714 instead of zero. The first module never suffers from a lack of input for the first units in a flowline. The second and any subsequent modules do encounter the situation of the first units being idle because there are no inputs available.

Figure 25 has the temporal output for the module configuration as it is in run sixteen of the first experiment. The first sequence of outputs is the production intervals for the first module. Using the same configuration as in the first module for the second module, the outputs from the first module are input into the second module. The temporal output of the second module is shown with the second sequence of outputs.

From this figure it becomes evident that the second module is not a repeat of the first. The interval between the first and second, and the second and third is 100 minutes, each in the first module. However, the interval between the second and third increases to 215 minutes in the second module. The system output with two modules is 23.

Thus the subsequent modules are a direct translation of the first only when the system is preloaded and the buffer capacity is infinite. In order to observe the activities within the second module, it is simulated. The second module's activities are not obtained by shifting the first module over and claiming the sequence of activities

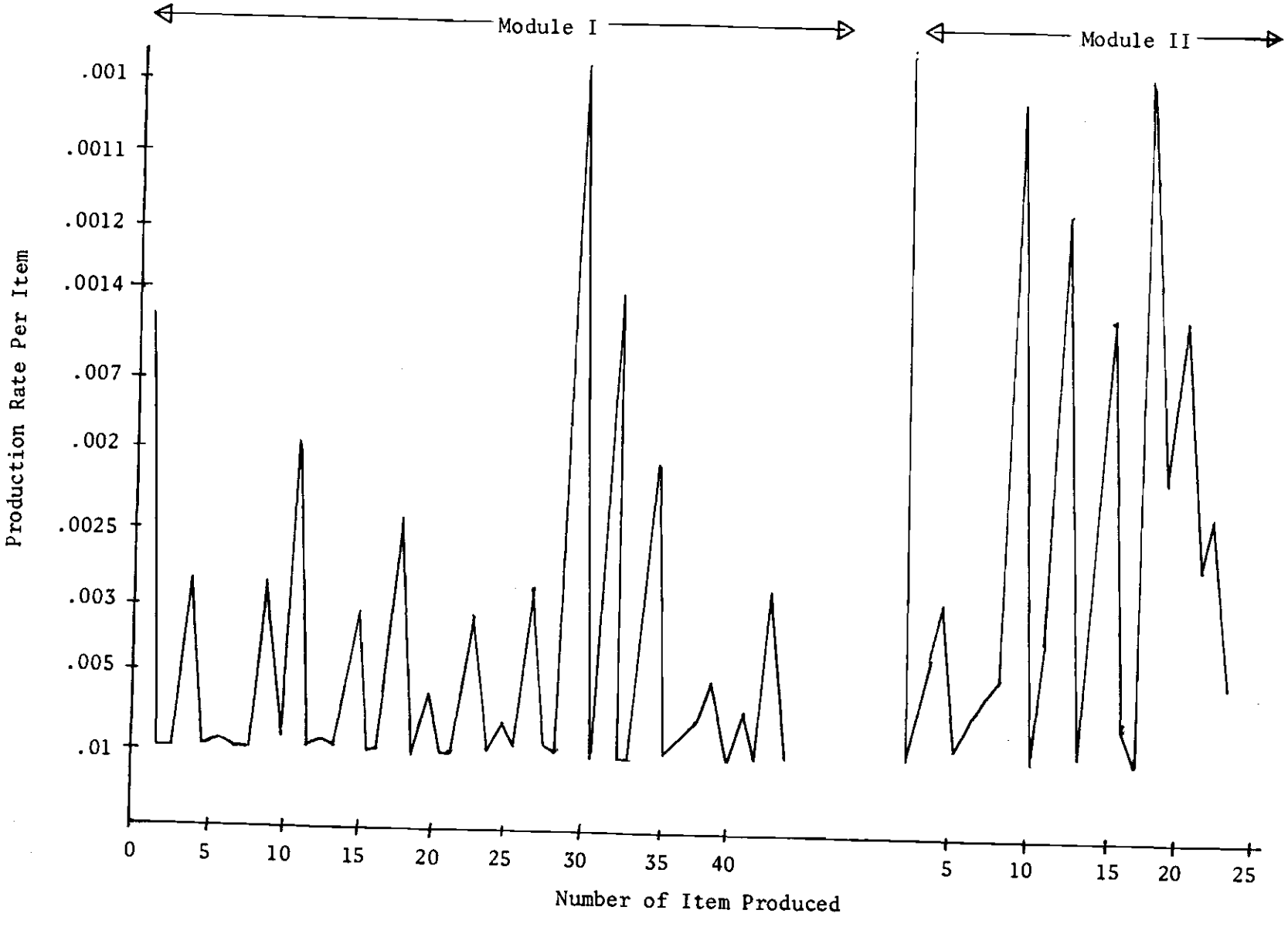


Figure 25. Production Interval--First and Second Modules

is the same.

In the cases where the system is subdivided into two modules, the output of the second module is the system's output.

4.8 Comparison of Applebaum's and Simulation Results

At the outset, it was stated that a comparison of the results from the simulation would be made with results obtained using Applebaum's methodology on the same system configuration. The comparisons were made on three configurations which collectively contain most of the variables included in this study.

The first system configuration equates to the tenth run in the first set of 32. Subsystem A1 has one unit; A2 has two units; B1 has two units; and B2 has one unit. There is one repair team available to each subsystem. The subsystems with two units are the first divergents from Applebaum's system. For independence, the failure and repair rates of subsystems must not be dependent. A subsystem with two units can be equated to a system with two subsystems having one repair team, thus evoking dependence. Buffer capacity is one, meaning that its only use is as a channeling device for items going from one subsystem to another rather than being used for in-process storage. Flowline "B" is involved in equipment sharing, however, flowline "A" is not. The system is at its low level, except B2. Low level repair rates exist for A1 and B1. This combination of failure rates and repair rates place A2 in a unique position; its repair rate is 1.5 times greater than its failure rate. To further compound this situation, periodic maintenance is scheduled every 240 minutes and it is to last

an average of 180 minutes.

Using the model set forth by Applebaum, there are several features of the above described system which are not accounted for, among which are: (1) performing periodic maintenance, (2) dependence due to having one repair term for subsystems with two units, and (3) equipment sharing on a flowline.

The results of Applebaum and the simulation are 0.921 and 0.838, respectively.

The second example run is 11 from set one. The major changes between this run and the previous example are: (1) susbystem A1 now has two units instead of one, and (-) flowline "A", as well as flowline "B" are equipment sharing, (3) A2 has one unit, (4) B1's failure rate is at its high level, while B2's failure rate is at its low level, (5) the repair rates of A1 and B1 are at high level, and (6) periodic maintenance is performed by B1 and B2.

The results of these comparisons are much closer at 0.926 and 0.900 for Applebaum and computer simulation respectively. Recall from the analysis that next to equipment sharing, the production time in A1 ranks second on the scale for isolated variables. Also recall that preferred level is having two units in A1. Although the repair rate exceeds the failure rate, the complexity of periodic maintenance is not in this subsystem.

Example three is the case of each variable being at its high level. The run which provides these conditions is number 32 of the first set of experiments. This is the first example where buffer capacity is infinite. The system is completely void of periodic

maintenance and the failure rate is 5.33 times greater than the repair rate. Each subsystem has but one unit. Therefore, the independence due to repair and failure much closer approximates the description in Applebaum's model. This is further substantiated by the results from the comparison where Applebaum predictes system availability as 0.9154 and the computer runs availability is 0.8980. That is within the confidence interval of $\pm$.03 established during the validation of the computer program.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

The conclusions, Section 5.1, reached as a result of the research are divided into two categories: (1) those limited to the model used in the study, and (2) those which apply to systems in general. The category of each conclusion is identified as the conclusive statement is presented.

Section 5.2 presents the recommendations for furthering the research of this study and other related areas. The other related areas include topics in system reliability and design of experiments.

5.1 Conclusions

5.1.1 Model to Study Complex Systems

A model was designed and a program written to accommodate complex sophisticated systems. The program represented a modular approach to system analysis. The module was multi-channel with multi-subsystems. The module was written with the necessary flexibility to adjust to larger systems.

The modular representation was checked for logical flow first by parts, followed by checking the flow with all operations working simultaneously. The modular representation was validated by collapsing variables to equate the module to a simpler system which fits the analytical model presented by Applebaum. Comparisons of the validation program and Applebaum's model yielded results which were all

within the variance established for pure or simulation error.

The model was used to study large complex systems via the modular approach. Twenty-seven design variables allowed the module to adjust to various system configurations. The list of variables included: (1) production time, (2) failure rate, (3) repair rate, (4) buffer capacity, (5) maintenance policy, (6) loadsharing, (7) equipment sharing, and (8) preloading. The responses from the model were system production and system availability.

5.1.2 The Experiments

An experiment was designed to test each of the 27 variables at two levels by measuring productivity and availability. The design was a 2^{27-22}_{III} fractional factorial design.

The first fraction provided estimates of 27 contrasts. For productivity, the contrast composed of repair teams and its aliased two factor interactions had the greatest magnitude, 32.75; the contrast containing the failure rate of the second subsystem in the first flowline, A2, and the aliased two factor interactions was second with an estimate of effect of 16.25.

For availability, the contrast containing system down criteria had the highest estimate, 66.18×10^{-2}. The contrast immediately following was preventive maintenance for A2 with a magnitude of 27.83×10^{-2}.

The aliasing structure of an experiment with 27 variables has too many two factor interactions in each contrast to assess the effects of module configuration with the estimates provided by the first fraction. Thus, after looking at the results of the first fraction, the

mirror image of the first was chosen as the second fraction to isolate the main effects. The magnitude of effect corresponding to repair teams for productivity was 33.66, which is three times greater than the absolute value for any other effect on productivity. The repair rate of the second subsystem in the first flowline was second with a value of -10.78.

For availability, system down criteria had the greatest magnitude of the main effects. System down had a value of 60.52 compared with the second highest value of -20.43 corresponding to periodic maintenance for the second subsystem in the first flowline.

The third experiment was conducted to ascertain if any of the two factor interactions were of the same magnitude as the main effects. After scrutinizing the contrasts, the third fraction was to isolate repair teams and its two factor interactions.

The interaction of repair teams and periodic maintenance had the highest valued affect, 9.14, for productivity. The interaction of repair teams and periodic maintenance for B2 ranked highest for availability with a value of 3.36. The magnitude of effect of the combined two factor interactions was the result of summing a large number of interactions, rather than particular interaction.

To test for significance of variables, ANOVA's were constructed for productivity and availability. The ANOVA for productivity showed the following variables were significant at the .05 level: (1) repair teams, (2) repair rate of A2, (3) failure rate of B2, and (4) preloading. Subsystem down criteria for A2, failure rate of A1, and repair rate of B2 were significant at the .1 level. With the extraction of other two

factor interactions whose effects are of relatively large magnitudes, other configuration constraints which show signs of dominance will show up significant. The qualitative variables which are dominate will continue to be and those which do not show signs of dominance are not likely to change or become dominant.

The ANOVA for availability had 12 of the 27 variables significant at the .05 level. Included as significant were: (1) system down criteria, (2) periodic maintenance of A2 and B2, (3) production time of the first subsystem in the first flowline, A1 and A2, (4) repair teams, (5) repair rate of A2, (6) failure rate of B1 and A2, and (7) preloading.

Upon completion of the experimental runs, additional runs were done to illustrate the interfacing of modules.

5.1.3 Maximizing Productivity and Availability

Steepest ascent was used to maximize productivity and availability. The analytical results were verified by computer simulation runs. The maximum possible was obtained both analytically and with the simulation runs.

After ascertaining the levels for the configuration constraints to maximize productivity using steepest ascent, a branch and bound mixed-integer programming algorithm yielded the optimal levels to maximize productivity. The production times, which dictate the number of units in a subsystem, were irrelevant for three of the four subsystems, as long as the production rate maintained line balance. The high mean time to failure required using steepest ascent was not needed for the optimal solution.

Although the production level for each solution was the system's maximum of 99 for the production run used, the levels of the configuration constraints differed. Thus, it can be seen that when the objective is of maximizing productivity, different configurations will provide the maximum. However, other constraints will dictate the optimal solution for maximization.

5.1.4 Transient and Steady-State Effects

The length of time the system was in the transient state was inversely proportional to the number of items produced over a given time interval. If the batch size was small, then the transient phenomenon was in effect longer than if the batch size was large. Therefore, the more items produced in a given time interval the less effect the transient phenomenon has on productivity.

For minimum batch sizes from the modeled system, the operation was constrained by the necessity of the presence or absence of certain system configuration constraints. For intermediate batch sizes greater than 56, the system must be equipped with regular repair teams, infinite buffers and have preloading.

Although some combinations of system configuration constraints bear great resemblance to systems assuming independence by having the same availability, the two could be equal for different reasons. Thus, this study in no way invalidates the assumption of independence by the models of Applebaum, Buzacott, ARINC, et al. However, the user of such models should be fully cognizant that the presence of other feasible system configuration constraints at particular levels can have a significant effect on an estimate of system availability.

5.2 Recommendations

This study does not consider economic constraints which are placed on a system. An investigation of the effects of economic constraints upon the model deserves consideration. The economic constraints might include such things as the cost and trade-offs associated with buffer size, maintenance procedures--including repair team locations, load sharing, and equipment sharing.

An evaluation of system design alternative layouts is a worthy area of study. The objective of such a study would be to focus upon functional system layout versus group layout. An investigation could reveal the optimal feasible alternatives for different type systems.

Time to failure and time to repair were assumed to be exponentially distributed in this study. However, a confirmation or negation of the inferences of this study is needed as applied to systems having other than exponentially distributed time to failure and time to repair.

APPENDICES

APPENDIX A

COMPUTER PROGRAM

```
ZIMMERMAN-LE*IEZKE.SYSTEM
                JOB
        FA11    FUNCTION,EXP RF$1,1,782
        FA12    FUNCTION,EXP RF$2,1,782
        FA21    FUNCTION,EXP RF$3,1,1190
        FA22    FUNCTION,EXP RF$4,1,1190
        FB11    FUNCTION,EXP RF$5,1,1090
        FB12    FUNCTION,EXP RF$6,1,1090
        FB21    FUNCTION,EXP RF$7,1,1340
        FB22    FUNCTION,EXP RF$8,1,1340
        RA11    FUNCTION,EXP RF$9,1,56
        RA12    FUNCTION,EXP RF$10,1,56
        RA21    FUNCTION,EXP RF$7,1,1
        RA22    FUNCTION,EXP RF$6,1,1
        RB11    FUNCTION,EXP RF$5,1,93
        RB12    FUNCTION,EXP RF$4,1,93
        RB21    FUNCTION,EXP RF$3,1,6
        RB22    FUNCTION,EXP RF$2,1,6
                INITIAL LSAB,1
                INITIAL SA,0
                INITIAL SB,0
                INITIAL NA1,2
                INITIAL PESA1,200
                INITIAL NA2,1
                INITIAL PESA2,100
                INITIAL NB1,2
                INITIAL PESB1,200
                INITIAL NB2,2
                INITIAL PESB2,200
                INITIAL BUFA1,100
                INITIAL BUFA2,100
                INITIAL BUFB1,100
                INITIAL BUFB2,100
                INITIAL REFA1,1
                INITIAL REFA2,1
                INITIAL REFB1,1
                INITIAL REFB2,1
                INITIAL PMTA1,240
                INITIAL PMTA2,240
                INITIAL PMTB1,10005
                INITIAL PMTB2,10005
                INITIAL TDT,0
                INITIAL TUT,0
                INITIAL LASUP,0
                INITIAL CTR,0
        VCTR    VARIABLE X$CTR+1
        VBIA    VARIABLE P$SYSU
        FA1D    VARIABLE (X$PESA1*3)/4
        VBA1    VARIABLE X$PL
                INITIAL PL,5
        FA2D    VARIABLE (X$PESA2*3)/4
```

```
FB1S    VARIABLE (X$FESB1*3)/4
VB31    VARIABLE X:PL1
        INITIAL PL1=0
FB2S    VARIABLE (X$FESB2*3)/4
LSFA11  VARIABLE (FN$FA11*3)/4
VARA11  VARIABLE X$MA11+X$P*TA1
LSFA12  VARIABLE (FN$FA12*3)/4
VARA12  VARIABLE X$MA12+X$P*TA1
LSFA21  VARIABLE (FN$FA21*3)/4
VARA21  VARIABLE X$MA21+X$P*TA2
LSFA22  VARIABLE (FN$FA22*3)/4
VARA22  VARIABLE X$MA22+X$P*TA2
LSFB11  VARIABLE (FN$FB11*3)/4
VARB11  VARIABLE X$MB11+X1P*TB1
LSFB12  VARIABLE (FN$FB12*3)/4
VARB12  VARIABLE X$MB12+X$P*TB1
LSFB21  VARIABLE (FN$FB21*3)/4
VARB21  VARIABLE X$MB21+X$P*TB2
LSFB22  VARIABLE (FN$FB22*3)/4
VARB22  VARIABLE X$MB22+X$P*TB2
VRA11   VARIABLE X$ST$11+FN$FA11
VLA11   VARIABLE X$ST$11+(FN$FA11*3)/4
LOA11   BVARIABLE (LS$TA12 EQ 1) AND (X$LSAB EQ 1)
RLPA11  BVARIABLE (C11 GE X$SLA11) OR (LS$TA11 EQ 1)
ROPA11  BVARIABLE (C11 GE X$SRA11) OR (LS$TA11 EQ 1)
VRA12   VARIABLE X$ST$12+FN$FA12
VLA12   VARIABLE X$ST$12+(FN$FA12*3)/4
LOA12   BVARIABLE (LS$TA11 EQ 1) AND (X$LSAB EQ 1)
RLPA12  BVARIABLE (C11 GE X$SLA12) OR (LS$TA12 EQ 1)
ROPA12  BVARIABLE (C11 GE X$SRA12) OR (LS$TA12 EQ 1)
FURA11  BVARIABLE (LS$TA11 EQ 1) AND (FU$RTA11 EQ 1)
FUCA11  BVARIABLE (LS$TA11 EQ 0) AND (FU$CREMA1 EQ 0)
FURA12  BVARIABLE (LS$TA12 EQ 1) AND (FU$RTA12 EQ 1)
FUCA12  BVARIABLE (LS$TA12 EQ 0) AND (FU$CREMA1 EQ 0)
VRA21   VARIABLE X$ST$21+FN$FA21
VLA21   VARIABLE X$ST$21+(FN$FA21*3)/4
LOA21   BVARIABLE (LS$TA22 EQ 1) AND (X$LSAB EQ 1)
RLPA21  BVARIABLE (C11 GE X$SLA21) OR (LS$TA21 EQ 1)
ROPA21  BVARIABLE (C11 GE X$SRA21) OR (LS$TA21 EQ 1)
VRA22   VARIABLE X$ST$22+FN$FA22
VLA22   VARIABLE X$ST$22+(FN$FA22*3)/4
LOA22   BVARIABLE (LS$TA21 EQ 1) AND (X$LSAB EQ 1)
RLPA22  BVARIABLE (C11 GE X$SLA22) OR (LS$TA22 EQ 1)
ROPA22  BVARIABLE (C11 GE X$SRA22) OR (LS$TA22 EQ 1)
FURA21  BVARIABLE (LS$TA21 EQ 1) AND (FU$RTA21 EQ 1)
FUCA21  BVARIABLE (LS$TA21 EQ 0) AND (FU$CREMA2 EQ 0)
FURA22  BVARIABLE (LS$TA22 EQ 1) AND (FU$RTA22 EQ 1)
FUCA22  BVARIABLE (LS$TA22 EQ 0) AND (FU$CREMA2 EQ 0)
VRB21   VARIABLE X$ST$21+FN$FB21
VRB11   VARIABLE X$ST$11+FN$FB11
```

```
VLB11   VARIABLE X$ST$11+(F[illegible]$FB11*3)/4
LOB11   BVARIABLE (LS$TB11 EQ 1) AND (X$LSAB EQ 1)
RLPB11  BVARIABLE (C$1 GE X$SL$11) OR (LS$TB11 EQ 1)
ROPB11  BVARIABLE (C$1 GE X$SR$11) OR (LS$TB11 EQ 1)
VRB12   VARIABLE X$ST$12+FU$FB12
VLB12   VARIABLE X$ST$12+(FU$FB12*3)/4
LOB12   BVARIABLE (LS$TB11 EQ 1) AND (X$LSAB EQ 1)
FUCB12  BVARIABLE (LS$TB12 EQ 0) AND (FU$CRE$B1 EQ 0)
FURB12  BVARIABLE (LS$TB12 EQ 1) AND (FU$RTB12 EQ 1)
FUCB11  BVARIABLE (LS$TB11 EQ 0) AND (FU$CRE$B1 EQ 0)
FURB11  BVARIABLE (LS$TB11 EQ 1) AND (FU$RTB11 EQ 1)
VLB21   VARIABLE X$ST$21+(FU$FB21*3)/4
LOB21   BVARIABLE (LS$TB22 EQ 1) AND (X$LSAB EQ 1)
ROPB12  BVARIABLE (C$1 GE X$SR$12) OR (LS$TB12 EQ 1)
RLPB12  BVARIABLE (C$1 GE X$SL$12) OR (LS$TB12 EQ 1)
RLPB21  BVARIABLE (C$1 GE X$SL$21) OR (LS$TB21 EQ 1)
ROPB21  BVARIABLE (C$1 GE X$SR$21) OR (LS$TB21 EQ 1)
VRB22   VARIABLE X$ST$22+FU$FB22
VLB22   VARIABLE X$ST$22+(FU$FB22*3)/4
FURB22  BVARIABLE (LS$TB22 EQ 1) AND (FU$RTB22 EQ 1)
LOB22   BVARIABLE (LS$TB21 EQ 1) AND (X$LSAB EQ 1)
ROPB22  BVARIABLE (C$1 GE X$SR$22) OR (LS$TB22 EQ 1)
RLPB22  BVARIABLE (C$1 GE X$SL$22) OR (LS$TB22 EQ 1)
FURB21  BVARIABLE (LS$TB21 EQ 1) AND (FU$RTB21 EQ 1)
FUCB21  BVARIABLE (LS$TB21 EQ 0) AND (FU$CRE$B2 EQ 0)
FUCB22  BVARIABLE (LS$TB22 EQ 0) AND (FU$CRE$B2 EQ 0)
B21F    BVARIABLE (LS$TB21 EQ 1) OR (LS$TB22 EQ 1)
B11F    BVARIABLE (LS$TB11 EQ 1) OR (LS$TB12 EQ 1)
A21F    BVARIABLE (LS$TA21 EQ 1) OR (LS$TA22 EQ 1)
A12F    BVARIABLE (LS$TA11 EQ 1) AND (LS$TA12 EQ 1)
A11F    BVARIABLE (LS$TA11 EQ 1) OR (LS$TA12 EQ 1)
A22F    BVARIABLE (LS$TA21 EQ 1) AND (LS$TA22 EQ 1)
B12F    BVARIABLE (LS$TB11 EQ 1) AND (LS$TB12 EQ 1)
B22F    BVARIABLE (LS$TB21 EQ 1) AND (LS$TB22 EQ 1)
SYS1    BVARIABLE (BV$A11F EQ 1) OR (BV$A21F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B21F EQ 1)
SYS2    BVARIABLE (BV$A11F EQ 1) AND (BV$A22F EQ 1) AND (BV$B12F EQ 1) AN
+D (BV$B22F EQ 1)
SYS3    BVARIABLE (BV$A12F EQ 1) OR (BV$A21F EQ 1) OR (BV$B12F EQ 1) OR (
+BV$B22F EQ 1)
SYS4    BVARIABLE (BV$A11F EQ 1) AND (BV$A21F EQ 1) AND (BV$B11F EQ 1) AN
+D (BV$B21F EQ 1)
SYS5    BVARIABLE (BV$A12F EQ 1) AND (BV$A22F EQ 1) AND (BV$B11F EQ 1) AN
+D (BV$B21F EQ 1)
SYS6    BVARIABLE (BV$A11F EQ 1) OR (BV$A21F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B22F EQ 1)
SYS7    BVARIABLE (BV$A12F EQ 1) AND (BV$A21F EQ 1) AND (BV$B11F EQ 1) AN
+D (BV$B22F EQ 1)
SYS8    BVARIABLE (BV$A11F EQ 1) OR (BV$A21F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B21F EQ 1)
```

```
SYS9   BVARIABLE (BV$A11F EQ 1) OR (BV$A21F EQ 1) OR (BV$B12F EQ 1) OR (
+BV$B21F EQ 1)
SYS10  BVARIABLE (BV$A11F EQ 1) AND (BV$A22F EQ 1) AND (BV$B11F EQ 1) AN
+D (BV$B22F EQ 1)
SYS12  BVARIABLE (BV$A11F EQ 1) AND (BV$A21F EQ 1) AND (BV$B12F EQ 1) AN
+D (BV$B22F EQ 1)
SYS17  BVARIABLE (BV$A11F EQ 1) AND (BV$A21F EQ 1) AND (BV$B11F EQ 1) AN
+D (BV$B22F EQ 1)
SYS18  BVARIABLE (BV$A11F EQ 1) OR (BV$A22F EQ 1) OR (BV$B12F EQ 1) OR (
+BV$B21F EQ 1)
SYS21  BVARIABLE (BV$A11F EQ 1) OR (BV$A22F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B22F EQ 1)
SYS25  BVARIABLE (BV$A12F EQ 1) AND (BV$A21F EQ 1) AND (BV$B12F EQ 1) AN
+D (BV$B22F EQ 1)
SYS26  BVARIABLE (BV$A11F EQ 1) OR (BV$A22F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B21F EQ 1)
SYS27  BVARIABLE (BV$A12F EQ 1) AND (BV$A21F EQ 1) AND (BV$B11F EQ 1) AN
+D (BV$B21F EQ 1)
SYS29  BVARIABLE (BV$A12F EQ 1) OR (BV$A22F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B21F EQ 1)
SYS30  BVARIABLE (BV$A12F EQ 1) OR (BV$A21F EQ 1) OR (BV$B12F EQ 1) OR (
+BV$B22F EQ 1)
SYS35  BVARIABLE (BV$A11F EQ 1) OR (BV$A21F EQ 1) OR (BV$B12F EQ 1) OR (
+BV$B22F EQ 1)
SYS36  BVARIABLE (BV$A12F EQ 1) AND (BV$A21F EQ 1) AND (BV$B12F EQ 1) AN
+D (BV$B22F EQ 1)
SYS37  BVARIABLE (BV$A12F EQ 1) OR (BV$A21F EQ 1) OR (BV$B11F EQ 1) OR (
+BV$B22F EQ 1)
SYS38  BVARIABLE (BV$A11F EQ 1) AND (BV$A22F EQ 1) AND (BV$B12F EQ 1) AN
+D (BV$B21F EQ 1)
SYS41  BVARIABLE (BV$A11F EQ 1) AND (BV$A21F EQ 1) AND (BV$B12F EQ 1) AN
+D (BV$B22F EQ 1)
A22FK  BVARIABLE (FU$FA22 EQ 1) AND (X$NA2 LT 2)
B22FK  BVARIABLE (FU$FB22 EQ 1) AND (X$NB2 LT 2)
CA1    VARIABLE C$A1A2+X$HA1
CA2    VARIABLE C$A2+X$HA2
CB1    VARIABLE C$B1B2+X$HB1
CB2    VARIABLE C$B2+X$HB2
VHA1   VARIABLE Y$HA1+1
VHA2   VARIABLE Y$HA2+1
VHB1   VARIABLE Y$HB1+1
VHB2   VARIABLE Y$HB2+1
PLA1   VARIABLE Y$PLUSA1+1
PLA2   VARIABLE Y$PLUSA2+1
PLB1   VARIABLE Y$PLUSB1+1
PLB2   VARIABLE Y$PLUSB2+1
SHA1   VARIABLE Y$HA1-1
SHA2   VARIABLE Y$HA2-1
```

```
SHB1    VARIABLE X$SHB1-1
SHB2    VARIABLE X$SHB2-1
MINA1   VARIABLE X$MINUSA1+1
MINA2   VARIABLE X$MINUSA2+1
MINB1   VARIABLE X$MINUSB1+1
MINB2   VARIABLE X$MINUSB2+1
CAPA1   BVARIABLE (G$A1A2 LT 3) AND (X$BUFA1 GE 3)
CAPA2   BVARIABLE (G$A2 LT 3) AND (X$BUFA2 GE 3)
CAPB1   BVARIABLE (G$B1B2 LT 3) AND (X$BUFB1 GE 3)
CAPB2   BVARIABLE (G$B2 LT 3) AND (X$BUFB2 GE 3)
        GENERATE 0,1,1 TIME(1)
        SAVEX REPB1,X$REPB1
        SAVEX PMTB1,X$PMTB1
        SAVEX REPB2,X$REPB2
        SAVEX PMTB2,X$PMTB2
        SAVEX REPA1,X$REPA1
        SAVEX PMTA1,X$PMTA1
        SAVEX REPA2,X$REPA2
        SAVEX PMTA2,X$PMTA2
        SAVEX NA1,X$NA1
        SAVEX PESA1,X$PESA1
        SAVEX LSAB,X$LSAB
        SAVEX SA,X$SA
        SAVEX BUFA1,X$BUFA1
        SAVEX PL,X$PL
        SAVEX PESA2,X$PESA2
        SAVEX BUFA2,X$BUFA2
        SAVEX NA2,X$NA2
        SAVEX NB1,X$NB1
        SAVEX PESB1,X$PESB1
        SAVEX SB,X$SB
        SAVEX BUFB1,X$BUFB1
        SAVEX PLB,X$PLB
        SAVEX PESB2,X$PESB2
        SAVEX BUFB2,X$BUFB2
        SAVEX NB2,X$NB2
        TERMINATE
        GENERATE 1,1,1 TIME(1)
        COMPARE BVA2FK EQ 1 GOTO(WAIT1)
        GENERATE 1,1,1 TIME(1)
        COMPARE BVB2FK EQ 1 GOTO(WAIT1)
        GENERATE 0 TIME(0)
        SEIZE PA11
        ADVANCE GOTO(+1,A12D)
        COMPARE LSAB,12 EQ 0 GOTO(NLSA11)
A12D    ADVANCE GOTO(+1,NLSA11)
        COMPARE X$LSAB EQ 1
        HOLD PTA11 TIME(V$FA1D)  GOTO(ESA11)
NLSA11  HOLD PTA11 TIME(X$PESA1)
ESA11   SEIZE SHA
```

```
        HOLD PSA11 TIME(X$SA)
        RELEASE SHA
        ADVANCE
        COMPARE V$CA1 LE X$SOJFA1
        RELEASE PA11  GOTO(WAT1)
        ADVANCE GOTO(+1,ALTA1)
        COMPARE Q$A1A2 LT 5 GOTO(WAT1)
ALTA1   SAVEX HA1,V$VHA1
        SAVEX PLUSA1,V$PLA1
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FALA1
        COMPARE X$HA1 GT 0
        COMPARE BV$C/PA1 EQ 1
        SAVEX HA1,V$SHA1
        SAVEX MIUSA1,V$MIA1
        RELEASE FALA1 GOTO(WAT1)
        GENERATE 0.100 TIME(0)
        SEIZE DBA1
        COMPARE Q$A1A2 LT 5
        ADVANCE GOTO(+1,+3)
        COMPARE FC$DBA1 LE 2
        HOLD DBBA1 TIME(1) GOTO(PLRA1)
        HOLD DBBA1 TIME(V$VBA1)
PLRA1   RELEASE DBA1
WAT1    QUEUE A1A2  GOTO(A21,A22)
        GENERATE 0 TIME(0)
        SEIZE PA12
        COMPARE X$NA1 GE 3
        ADVANCE GOTO(+1,A11D)
        COMPARE LS$T/11 EQ 0  GOTO(NLSA12)
A11D    ADVANCE GOTO(+1,NLSA12)
        COMPARE X$LSAB E) 1
        HOLD PTA12 TIME(VIEA11D)  GOTO(ESA12)
NLSA12  HOLD PTA12 TIME(X$PESA1)
ESA12   SEIZE SHA
        HOLD PSA12 TIME(X$SH)
        RELEASE SHA
        COMPARE V$CA1 LE X$SOJFA1
        RELEASE PA12
        ADVANCE GOTO(+1,ALTA1)
        COMPARE Q$A1A2 LT 5 GOTO(WAT1)
A21     SEIZE PA21
        ADVANCE GOTO(+1,A22D)
        COMPARE LS$T/22 EQ 0  GOTO(NLSA21)
A22D    ADVANCE GOTO(+1,NLSA21)
        COMPARE X$LSAB EQ 1
        HOLD PTA21 TIME(VIEA22D) GOTO(ESA21)
NLSA21  HOLD PTA21 TIME(X$PESA2)
ESA21   SEIZE SHA
```

```
        HOLD PSA21 TIME(X$SA)
        RELEASE SHA
        COMPARE V$CA2 LE X$B(FA2
        RELEASE PA21
        ADVANCE GOTO(+1,ALTA2)
        COMPARE Q$A2 LT 5 GOTO(WAT2)
ALTA2   SAVEX MA2,V$VMA2
        SAVEX PLUSA2,V$PLA2
        TERMINATE
        GENERATE 5 TIME(0)
        SEIZE FALA2
        COMPARE X$MA2 GT 0
        COMPARE V$CFPA2 EQ 1
        SAVEX MA2,V$SMA2
        SAVEX MIUSA2,V$MINA2
        RELEASE FALA2 GOTO(WAT2)
A22     SEIZE PA22
        COMPARE X$NA2 GE 2
        ADVANCE GOTO(+1,A21D)
        COMPARE LS$T#21 EQ 0 GOTO(NLSA22)
A21D    ADVANCE GOTO(+1,NLSA22)
        COMPARE X$LSA2 EQ 1
        HOLD PTA22 TIME(V$FA2D) GOTO(ESA22)
NLSA22  HOLD PTA22 TIME(X$PESA2)
ESA22   SEIZE SHA
        HOLD PSA22 TIME(X$SA)
        RELEASE SHA
        COMPARE V$CA2 LE X$B(FA2
        RELEASE PA22
        ADVANCE GOTO(+1,ALTA2)
        COMPARE Q$A2 LT 5
WAT2    QUEUE A2
        SEIZE AYA
        COMPARE QUEXT3 EQ 1
        LOGIC S,A,ISH
        COMPARE V$LA2 EQ 1
        LOGIC R,A2ISH
        PRINT C,1
        RELEASE AYA
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE PB11
        ADVANCE GOTO(+1,B11D)
        COMPARE LS$T#12 EQ 0 GOTO(NLSB11)
B11D    ADVANCE GOTO(+1,NLSB11)
        COMPARE X$LSB1 E, 1
        HOLD PTB11 TIME(V$FB1D) GOTO(ESB11)
NLSB11  HOLD PTB11 TIME(X$PESB1)
ESB11   SEIZE SHB
        HOLD PSB11 TIME(X$SB)
```

```
        RELEASE SHB
        ADVANCE
        COMPARE V$CB1 LE X$BUFB1
        RELEASE PB11
        ADVANCE GOTO(+1,ALTB1)
        COMPARE Q$B1B2 LT 5 GOTO(WBT1)
ALTB1   SAVEX HB1,V$VHB1
        SAVEX PLUSB1,V$PLB1
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FALB1
        COMPARE X$HB1 GT 0
        COMPARE BV$CAPB1 EQ 1
        SAVEX HB1,V$SHB1
        SAVEX MIUSB1,V$MIN1
        RELEASE FALB1 GOTO(WBT1)
        GENERATE 0,100 TIME(0)
        SEIZE DB1
        COMPARE Q$B1B2 LT 5
        ADVANCE GOTO(+1,+3)
        COMPARE FL$DBB1 LE 2
        HOLD DUBB1 TIME(1) GOTO(PLRB1)
        HOLD DUBB1 TIME(V$VBB1)
PLRB1   RELEASE DBB1
WBT1    QUEUE B1B2 GOTO(B21,B22)
        GENERATE 0 TIME(0)
        SEIZE PB12
        COMPARE X$NB1 GE 2
        ADVANCE GOTO(+1,B12D)
        COMPARE LS$TB11 EQ 0 GOTO(NLSB12)
B12D    ADVANCE GOTO(+1,NLSB12)
        COMPARE X$LSAB EQ 1
        HOLD PTB12 TIME(V$FB12) GOTO(ESB12)
NLSB12  HOLD PTB12 TIME(X$PESB1)
ESB12   SEIZE SHB
        HOLD PSB12 TIME(X$SB)
        RELEASE SHB
        ADVANCE
        COMPARE V$CB1 LE X$BUFB1
        RELEASE PB12
        ADVANCE GOTO(+1,ALTB1)
        COMPARE Q$B1B2 LT 5 GOTO(WBT1)
B21     SEIZE PB21
        ADVANCE GOTO(+1,B21D)
        COMPARE LS$TB22 EQ 0 GOTO(NLSB21)
B21D    ADVANCE GOTO(+1,NLSB21)
        COMPARE X$LSAB EQ 1
        HOLD PTB21 TIME(V$FB2D) GOTO(ESB21)
NLSB21  HOLD PTB21 TIME(X$PESB2)
ESB21   SEIZE SHB
```

```
        HOLD PSB21 TIME(X$SB)
        RELEASE SHB
        COMPARE V$CB2 LE X$$UFB2
        RELEASE PB21
        ADVANCE GOTO(+1,ALTB2)
        COMPARE Q$B2 LT 5 GOTO(WBT2)
ALTB2   SAVEX HB2,V$VHB2
        SAVEX PLUSB2,V$PLB2
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FALB2
        COMPARE X$HB2 GT 0
        COMPARE BV$CAPB2 EQ 1
        SAVEX HB2,V$SHB2
        SAVEX MIUSB2,V$MINB2
        RELEASE FALB2 GOTO(WBT2)
B22     SEIZE PB22
        COMPARE X$HB2 GE 2
        ADVANCE GOTO(+1, B22D)
        COMPARE LS$TB21 EQ 0 GOTO(NLSB22)
B22D    ADVANCE GOTO(+1,NLSB22)
        COMPARE X$LSB EQ 1
        HOLD PTB22 TIME(V$FB22) GOTO(ESB22)
NLSB22  HOLD PTB22 TIME(X$PESB2)
ESB22   SEIZE SHB
        HOLD PSB22 TIME(X$SB)
        RELEASE SHB
        COMPARE V$CB2 LT X$$UFB2
        RELEASE PB22
        ADVANCE GOTO(+1,ALTB2)
        COMPARE Q$B2 LT 5
WBT2    QUEUE B2
        SEIZE BYB
        QUEUE NXT2
        COMPARE LS$AWISH EQ 1
BLAB    ADVANCE
        COMPARE LS$AWISH EQ 0
        RELEASE BYB
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FRA11
        SAVEX STA11,C$1
        SAVEX SRA11,V$VRA11
        SAVEX SLA11,V$VLA11
        ADVANCE GOTO(+1,RPA11)
        COMPARE BV$LOA11 EQ 1
        ADVANCE
        COMPARE BV$RLPA11 EQ 1
        ADVANCE GOTO(+1,ENDA11)
        COMPARE C$1 GE X$SLA11 GOTO(RPA11)
```

```
RMA11   ADVANCE
        COMPARE BVERCPA11 EQ 1
        ADVANCE GOTO(+1,END11)
        COMPARE C$1 GE XDSRA11
RPA11   PREEMPT PTA11
        PRINT C$1
        LOGIC S,TA11
        ADVANCE GOTO(+1,PERA11)
        COMPARE X$REPA1 EQ 1
        QUEUE PARA1 GOTO(PMA11,RMA11)
PERA11  QUEUE POA1 GOTO(PMA11)
PMA11   COMPARE C$1 GE VSVARA11
        SEIZE CREWA1
        HOLD PETA11 TIME(FN$RA11)
        SAVEX IA11,C$1
        PRINT C$1
        RELEASE CREWA1  GOTO(IA11)
RMA11   SEIZE RCRWA1
        HOLD RTA11 TIME(FN$RA11)
        RELEASE RCRWA1
        PRINT C$1
IA11    LOGIC R,TA11
        RETURN PTA11
END11   COMPARE LS$TA11 EQ 0
        RELEASE FAA11
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE DUMA11
        HOLD DA11 TIME(XPPTA11)
        ADVANCE GOTO(+1, A11)
        COMPARE BV$FURA11 EQ 1
        ADVANCE
        COMPARE LS$TA11 EQ 0 GOTO(SCA11)
GA11    ADVANCE GOTO(+1,+2)
        COMPARE LS$TA11 EQ 1 GOTO(ENA11)
        COMPARE BV$FUCA11 EQ 1
SCA11   SEIZE CREWA1
        PRINT C$1
        PREEMPT PTA11
        LOGIC S,TA11
        HOLD DUA11 TIME(FN$RA11)
        SAVEX MA11,C$1
        PRINT C$1
        LOGIC R,TA11
        RETURN PTA11
        RELEASE CREWA1
ENA11   ADVANCE
        COMPARE LS$TA11 EQ 0
        RELEASE DUMA11
        TERMINATE
```

```
        GENERATE 0 TIME(U)
        SEIZE FAA12
        COMPARE X$NA1 GE 2
        SAVEX STA12,C$1
        SAVEX SRA12,V$VRA12
        SAVEX SLA12,V$VLA12
        ADVANCE GOTO(+1,RPA12)
        COMPARE BVBLOA12 EQ 1
        ADVANCE
        COMPARE BVBRLPA12 EQ 1
        ADVANCE GOTO(+1,ENDA12)
        COMPARE C$1 GE XBSLA12 GOTO(PPA12)
RRA12   ADVANCE
        COMPARE BVSNOPA12 EQ 1
        ADVANCE GOTO(+1,ENDA12)
        COMPARE C$1 GE XBSRA12
RPA12   PREEMPT PTA12
        PRINT C$1
        LOGIC S,TA12
        ADVANCE GOTO(+1,PERA12)
        COMPARE X$RCPA1 EQ 1
        QUEUE PARA1 GOTO(PPA12,RMA12)
PERA12  QUEUE PQA1 GOTO(PMA12)
PMA12   COMPARE C$1 GE V$VARA12
        SEIZE CREWA1
        HOLD PETA12 TIME(FN$RA12)
        SAVEX MA12,C$1
        PRINT C$1
        RELEASE CREWA1 GOTO(IA12)
RMA12   SEIZE RCR,A1
        HOLD RTA12 TIME(FN$RMA12)
        PRINT C$1
        RELEASE RCRMA1
IA12    LOGIC R,IA12
        RETURN PTA12
ENDA12  COMPARE LSTA12 EQ 0
        RELEASE FAA12
        TERMINATE
        GENERATE 0 TIME(U)
        SEIZE DUMA12
        COMPARE X$NA1 GE 2
        HOLD DA12 TIME(X$PMTA1)
        ADVANCE GOTO(+1,DA12)
        COMPARE BVSFURA12 EQ 1
        ADVANCE
        COMPARE LSSTA12 EQ 0 GOTO(SCA12)
DA12    ADVANCE GOTO(+1,+2)
        COMPARE LSSTA12 EQ 1 GOTO(ENA12)
        COMPARE BVSFUCA12 EQ 1
SCA12   SEIZE CREWA1
```

```
        PRINT C$1
        PREEMPT PTA12
        LOGIC S,TA12
        HOLD DUM12 TIME(FNBRA12)
        SAVEX MA12,C$1
        PRINT C$1
        LOGIC R,TA12
        RETURN PTA12
        RELEASE CREWA1
ENA12   ADVANCE
        COMPARE LSST 12 EQ 0
        RELEASE DUMA12
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FAA21
        SAVEX STA21,C$1
        SAVEX SHA21,VBVR$21
        SAVEX SLA21,VBVL$21
        ADVANCE GOTO(+1,RRA21)
        COMPARE BVBLDA21 EQ 1
        ADVANCE
        COMPARE BVBLPA21 EQ 1
        ADVANCE GOTO(+1,ENDA21)
        COMPARE C$1 GE XBSLA21 GOTO(RPA21)
RRA21   ADVANCE
        COMPARE BVARDA21 EQ 1
        ADVANCE GOTO(+1,ENDA21)
        COMPARE C$1 GE XBSRA21
RPA21   PREEMPT PTA21
        PRINT C$1
        LOGIC S,TA21
        ADVANCE GOTO(+1,PERA21)
        COMPARE XTR_PA2 EQ 1
        QUEUE PARA2 GOTO(PERA21,PMA21)
PERA21  QUEUE PDA2 GOTO(PMA21)
PMA21   COMPARE C$1 GE VBVRA21
        SEIZE CREWA2
        HOLD PETA21 TIME(FNBRA21)
        SAVEX MA21,C$1
        PRINT C$1
        RELEASE CREWA2 GOTO(IA21)
RMA21   SEIZE RCRWA2
        HOLD PTA21 TIME(FNBRA21)
        RELEASE RCRWA2
IA21    LOGIC R,TA21
        RETURN PTA21
ENDA21  ADVANCE
        COMPARE LSST21 EQ 0
        RELEASE FAA21
        TERMINATE
```

```
        GENERATE 0 TIME(0)
        SEIZE DUMA21
        HOLD DA21 TIME(X$PMTA2)
        ADVANCE GOTO(+1,GA21)
        COMPARE BV$FUGA21 EQ 1
        ADVANCE
        COMPARE LG$TA21 EQ 0 GOTO(SCA21)
GA21    ADVANCE GOTO(+1,+2)
        COMPARE LG$TA21 EQ 1 GOTO(ENA21)
        COMPARE BV$FUGA21 EQ 1
SCA21   SEIZE CREWA2
        PRINT C*1
        PREEMPT PTA21
        LOGIC S,TA21
        HOLD DUA21 TIME(EN$RA21)
        SAVEX MA21,C$1
        PRINT C*1
        LOGIC R,TA21
        RETURN PTA21
        RELEASE CREWA2
ENA21   ADVANCE
        COMPARE LG$TA21 EQ 0
        RELEASE DUMA21
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE PMA22
        COMPARE X*NA2 GE 2
        SAVEX STA22,C$1
        SAVEX SPA22,V$VRA22
        SAVEX SLA22,V$VLA22
        ADVANCE GOTO(+1,RPA22)
        COMPARE BV$LOA22 EQ 1
        ADVANCE
        COMPARE BV$RLPA22 EQ 1
        ADVANCE GOTO(+1,ENDA22)
        COMPARE C*1 GE X$SLA22 GOTO(RPA22)
RRA22   ADVANCE
        COMPARE BV$ROPA22 EQ 1
        ADVANCE GOTO(+1,ENDA22)
        COMPARE C*1 GE X$SRA22
RPA22   PREEMPT PTA22
        PRINT C*1
        LOGIC S,TA22
        ADVANCE GOTO(+1,PERA22)
        COMPARE X*REPA2 EQ 1
        GUIDE PARA2 GOTO(PMA22,PMA22)
PERA22  GUIDE POA2 GOTO(PMA22)
PMA22   COMPARE C*1 GE V$VARA22
        SEIZE CREWA2
        HOLD PETA22 TIME(EN$RA22)
```

```
        SAVEX MA22,C$1
        PRINT C*1
        RELEASE CREWA2 GOTO(IA22)
RMA22   SEIZE RORWA2
        HOLD RTA22 TIME(FN$RA22)
        PRINT C*1
        RELEASE RORWA2
IA22    LOGIC R,TA22
        RETURN RTA22
ENDA22  ADVANCE
        COMPARE LS$TA22 EQ 0
        RELEASE FAA22
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE DUMA22
        COMPARE X$NA2 GE 2
        HOLD DA22 TIME(X$PMTA2)
        ADVANCE GOTO(+1,GA22)
        COMPARE BV$FURA22 EQ 1
        ADVANCE
        COMPARE LS$TA22 EQ 0 GOTO(SCA22)
GA22    ADVANCE GOTO(+1,+2)
        COMPARE LS$TA22 EQ 1 GOTO(ENA22)
        COMPARE BV$FUCA22 EQ 1
SCA22   SEIZE CREWA2
        PRINT C*1
        PRECEPT PTA22
        LOGIC S,TA22
        HOLD DUA22 TIME(FN$RA22)
        SAVEX MA22,C$1
        PRINT C*1
        LOGIC R,TA22
        RETURN PTA22
        RELEASE CREWA2
ENA22   ADVANCE
        COMPARE LS$TA22 EQ 0
        RELEASE DUMA22
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FAB11
        SAVEX ST$11,C$1
        SAVEX SPB11,V$VPB11
        SAVEX SLB11,V$VLB11
        ADVANCE GOTO(+1,RPB11)
        COMPARE BV$LOB11 EQ 1
        ADVANCE
        COMPARE BV$RLPB11 EQ 1
        ADVANCE GOTO(+1,ENDB11)
        COMPARE C*1 GE X$SLB11 GOTO(RPB11)
RRB11   ADVANCE
```

```
        COMPARE BVTROPB11 EQ 1
        ADVANCE GOTO(+1,ENDB11)
        COMPARE C11 GE X$SR11
RPB11   PREEMPT P1,11
        PRINT C$1
        LOGIC S,T11
        ADVANCE GOTO(+1,PER11)
        COMPARE X$REP31 EQ 1
        QUEUE PARB1 GOTO(PMB11,RMB11)
PERB11  QUEUE PO1 GOTO(PMB11)
PMB11   COMPARE C$1 GE V$VARB11
        SEIZE CREW1
        HOLD PET11 TIME(FN$RP11)
        SAVEX MB11,C$1
        PRINT C$1
        RELEASE CREW1  GOTO(IB11)
RMB11   SEIZE RCPX1
        HOLD RT11 TIME(FN$RB11)
        PRINT C$1
        RELEASE RCPX1
IB11    LOGIC R,T11
        RETURN PT11
ENDB11  ADVANCE
        COMPARE LS$T11 EQ 0
        RELEASE FA11
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE DUM11
        HOLD DB11 TIME(X$PM11)
        ADVANCE GOTO(+1,+11)
        COMPARE BVTFURB11 EQ 1
        ADVANCE
        COMPARE LS$TB11 EQ 0 GOTO(SCB11)
GB11    ADVANCE GOTO(+1,+2)
        COMPARE LS$TB11 EQ 1 GOTO(ENB11)
        COMPARE BVTFUCB11 EQ 1
SCB11   SEIZE CREW1
        PRINT C$1
        PREEMPT P1,11
        LOGIC S,T11
        HOLD DU11 TIME(FN$RB11)
        SAVEX MB11,C$1
        PRINT C$1
        LOGIC R,T11
        RETURN PT11
        RELEASE CREW1
ENB11   ADVANCE
        COMPARE LS$TB11 EQ 0
        RELEASE DUMB11
        TERMINATE
```

```
        GENERATE 0 TIME(0)
        SEIZE FAB12
        COMPARE X$NB1 GE 2
        SAVEX STB12,C$1
        SAVEX SRB12,V$VRB12
        SAVEX SLB12,V$VLB12
        ADVANCE GOTO(+1,RRB12)
        COMPARE BV$LOB12 EQ 1
        ADVANCE
        COMPARE BV$RLPB12 EQ 1
        ADVANCE GOTO(+1,ENDB12)
        COMPARE C$1 GE X$SLB12 GOTO(RPB12)
RRB12   ADVANCE
        COMPARE BV$ROPB12 EQ 1
        ADVANCE GOTO(+1,ENDB12)
        COMPARE C$1 GE X$SRB12
RPB12   PREEMPT PTB12
        PRINT C$1
        LOGIC S,TB12
        ADVANCE GOTO(+1,PERB12)
        COMPARE X$REPB1 EQ 1
        QUEUE PARB1 GOTO(PMB12,RMB12)
PERB12  QUEUE PD 1 GOTO(PMB12)
PMB12   COMPARE C$1 GE V$VARB12
        SEIZE CREWB1
        HOLD RETB12 TIME(FN$RB12)
        SAVEX MB12,C$1
        PRINT C$1
        RELEASE CREWB1 GOTO(IB12)
RMB12   SEIZE RCRWB1
        HOLD RTB12 TIME(FN$RB12)
        PRINT C$1
        RELEASE RCRWB1
IB12    LOGIC R,TB12
        RETURN PTB12
ENDB12  ADVANCE
        COMPARE LS$TB12 EQ 0
        RELEASE FAB12
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE DUMB12
        COMPARE X$NB1 GE 2
        HOLD DB12 TIME(X$PMTB1)
        ADVANCE GOTO(+1,GB12)
        COMPARE BV$FURB12 EQ 1
        ADVANCE
        COMPARE LS$TB12 EQ 0 GOTO(SCB12)
GB12    ADVANCE GOTO(+1,+2)
        COMPARE LS$TB12 EQ 1 GOTO(ENB12)
        COMPARE BV$FUCB12 EQ 1
```

```
SCR12   SEIZE CREW81
        PRINT C*1
        PREEMPT PTB12
        LOGIC S,TB12
        HOLD DUB12 TIME(FN$RB12)
        SAVEX MB12,C$1
        PRINT C*1
        LOGIC R,TB12
        RETURN PTB12
        RELEASE CREWB1
ENB12   ADVANCE
        COMPARE LS$TB12 EQ 0
        RELEASE DUMB12
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FAB21
        SAVEX STB21,C$1
        SAVEX SPB21,V$VRB21
        SAVEX SLB21,V$VLB21
        ADVANCE GOTO(+1,RPB21)
        COMPARE AV$LOB21 EQ 1
        ADVANCE
        COMPARE AV$REPB21 EQ 1
        ADVANCE GOTO(+1,ENDB21)
        COMPARE C*1 GE X$SLB21 GOTO(RPB21)
RRB21   ADVANCE
        COMPARE AV$ROPB21 EQ 1
        ADVANCE GOTO(+1,ENDB21)
        COMPARE C*1 GE X$SRB21
RPB21   PREEMPT PTB21
        PRINT C*1
        LOGIC S,TB21
        ADVANCE GOTO(+1,PERB21)
        COMPARE X$REPB2 EQ 1
        CHANGE PARB2 GOTO(PMB21,RMB21)
PERB21  CHANGE PARB2 GOTO(PMB21)
PMB21   COMPARE C*1 GE V$VARB21
        SEIZE CREWB2
        HOLD PETB21 TIME(FN$RB21)
        SAVEX MB21,C$1
        PRINT C*1
        RELEASE CREWB2  GOTO(IB21)
RMB21   SEIZE RCRWB2
        HOLD RTB21 TIME(FN$RB21)
        PRINT C*1
        RELEASE RCRWB2
IB21    LOGIC R,TB21
        RETURN PTB21
ENDB21  ADVANCE
        COMPARE LS$TB21 EQ 0
```

```
        RELEASE FAB21
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE DUMB21
        HOLD DB21 TIME(X$PMT22)
        ADVANCE GOTO(+1,GB21)
        COMPARE BV$FUR22 EQ 1
        ADVANCE
        COMPARE LS$TB21 EQ 0 GOTO(SCB21)
GB21    ADVANCE GOTO(+1,+2)
        COMPARE LS$TB21 EQ 1 GOTO(ENB21)
        COMPARE BV$FUC22 EQ 1
SCB21   SEIZE CREWB2
        PRINT C*1
        PREEMPT PTB21
        LOGIC S,TB21
        HOLD DU21 TIME(ENB$R22)
        SAVEX MB21,C$1
        PRINT C*1
        LOGIC R,TB21
        RETURN PTB21
        RELEASE CREWB2
ENB21   ADVANCE
        COMPARE LS$TB21 EQ 0
        RELEASE DUMB21
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE FAB22
        COMPARE X*NB2 GE 2
        SAVEX STB22,C$1
        SAVEX SPB22,V$VRB22
        SAVEX SEB22,V$VLB22
        ADVANCE GOTO(+1,RPB22)
        COMPARE BV$LOB22 EQ 1
        ADVANCE
        COMPARE BV$RLP22 EQ 1
        ADVANCE GOTO(+1,ENDB22)
        COMPARE C*1 GE X$SLB22 GOTO(RPB22)
RRB22   ADVANCE
        COMPARE BV$RDP22 EQ 1
        ADVANCE GOTO(+1,ENDB22)
        COMPARE C*1 GE X$SRB22
RPB22   PREEMPT PTB22
        PRINT C*1
        LOGIC S,TB22
        ADVANCE GOTO(+1,PERB22)
        COMPARE X*REP22 GE 1
        QUEUE PARB2 GOTO(PERB22,PMB22)
PEP22   QUEUE PQB2 GOTO(PMB22)
PMB22   COMPARE C*1 GE V$VAR22
```

```
        SEIZE CPEW32
        HOLD PETR22 TIME(FN$RB22)
        SAVEX MP22,C$1
        PRINT C$1
        RELEASE CPEWP2 GOTO(+B22)
RMB22   SEIZE RCRW32
        HOLD RTP22 TIME(FN$RB22)
        PRINT C$1
        RELEASE RCRWP2
IB22    LOGIC R,TB22
        RETURN RTB22
ENDB22  ADVANCE
        COMPARE LS$TP22 EQ 0
        RELEASE FAB22
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE DUMP22
        COMPARE X$NB32 GE 2
        HOLD DB22 TIME(Y$DMTB2)
        ADVANCE GOTO(+1,GB22)
        COMPARE BVSERRB22 EQ 1
        ADVANCE
        COMPARE LS$TP22 EQ 0 GOTO(SCB22)
GB22    ADVANCE GOTO(+1,+2)
        COMPARE LS$TP22 EQ 1 GOTO(ENB22)
        COMPARE BVSERRB22 EQ 1
SCB22   SEIZE CPEW32
        PRINT C$1
        PREEMPT RTB22
        LOGIC S,TB22
        HOLD DUMP22 TIME(FN$RB22)
        SAVEX MP22,C$1
        PRINT C$1
        LOGIC R,TB22
        RETURN RTB22
        RELEASE CPEWP2
ENB22   ADVANCE
        COMPARE LS$TP22 EQ 0
        RELEASE DUMP22
        TERMINATE
        GENERATE 0 TIME(0)
        SEIZE COMP
CSYS    COMPARE BV$SYSV EQ 1
        ASSIGN SYSD,C$1
        PRINT C$1
CYSU    COMPARE BV$SYSV EQ 0
        ASSIGN SYSU,C$1
        PRINT C$1
        SAVEX DT,P$SYSU-P$SYSD
        PRINT X$DT
```

```
SAVEX TOT,X$TOT+X$OT
SAVEX UT,P$SYSM-X$LASUP
SAVEX TUT,X$TUT+X$UT
SAVEX LASUP,V$VBIA
SAVEX CTR,V$VCTR
RELEASE COMP
TERMINATE
START 0,10000
END
```

APPENDIX B

PROGRAM SYMBOLOGY

F---* Indicates the function for rate of failure of the given unit.

R--- Indicates the function for repair rate.

TDT Total downtime for the system during the production run.

TUT Total time the system was operational.

LASUP An instrument used to save the time the system last became operational.

CTR A counter to record the number of times the system was down.

VCTR A variable used to update the counter.

VBIA A variable used to place the time in instrument "LASUP".

N-- The number of units in subsystem r.

PES-- The production time for an item in subsystem r.

F--D A variable to compute production time if load sharing.

LSAB 1 if load sharing; 0 if not load sharing.

S- The time an item must spend in the shared facility.

BUF-- The capacity of a given buffer (the capacity is the same for all).

VB-- Safety value variable; the length of time to get item into buffer when preloading.

LSF--- Failure rate when load sharing and production rate is increased.

REP-- The number of regular repair teams available to a given subsystem.

VAR--- Record of time the next periodic maintenance is due.

---

*Note: ---represents the following: the first - is the flow-line; the second - is the ith subsystem, and the third - is the jth unit.

PMT-- Periodic maintenance interval.

ST--- Indicates the current clocktime.

SR--- Indicates the current clocktime plus the time to failure.

VR--- Variable to place the computed time of next failure in SR---.

SL--- Indicates the current clock time plus (the time to failure times a load sharing factor).

VL--- Indicates time the next failure will occur when load sharing.

LO--- Loadsharing and the other unit is inoperative.

RLP--- Regular failure or a periodic preempt under load sharing.

RDP--- Regular failure or periodic preempt not under loadsharing.

FUR--- Unit has failed and is in regular repair.

FUC--- Unit is operational and periodic crew is available.

--iF Subsystem has failed and i is the number of units the criteria requires to be in a failed state.

SYS- System failure criteria for the ith run.

PL and PLB Time to hold transactions when preloading such that the system is not overloaded.

---FK Boolean variable to check when production is not in progress because the unit is non-existent.

C-- Sums the number of items completed from a subsystem and waiting to go to the next subsystem.

VH-- A variable to place items in a ficticious storage area.

PL-- A counter for the total number of items entering the ficticious storage.

SH-- A variable to take items out of the ficticious storage area.

MIN-- Tallies the total number of items extracted from the dummy storage.

CAP-- A Boolean variable to determine if an item should be emitted from the dummy storage into the queue.

P--- The production area for unit i in the subsystem.

T--- A switch to indicate if the unit is in a fail or operate state.

PT--- A workstation in the ith unit of the subsystem.

SH- The workstation shared by subsystems of a flowline.

H-- The dummy storage area to prevent a system overload by having too many parts in the queue.

FAL-- The facility to divert items from the queue to the dummy storage area.

DB-- Facility to preload the system and buffers.

DUB-- Holding area for items to fill buffers and preload the system.

A1A2 Queue representing the buffer for items leaving subsystem A1 and awaiting entry into subsystem A2.

A2 Queue representing the buffer for items leaving subsystem A2 and awaiting entry to the next subsystem if one exists or awaiting link-up with an item or items from other flowlines.

-Y- The link-up facility for items coming from the rth flowline.

NXTB A dummy queue for items from flowline B to be utilized during link-up.

AWISH A switch to signal other flowlines that an item from flowline A is available for link-up.

BLAB A dummy holding area for items from flowline B used in link-up.

PS--- A workstation in the facility shared by subsystems of a given flowline.

B1B2 A queue representing the buffer for items leaving subsystem B1 and awaiting entry into B2.

FA--- Facility to generate failures for the given unit.

PAR-- Queue used for units which fail and will be repaired by either regular maintenance crews or the periodic maintenance team.

PO-- Queue used for units which fail and must remain down until periodic maintenance is performed.

CREW-- The crew which performs periodic maintenance.

PET--- Workstation for repair by periodic crew when equipment failed prior to time for periodic maintenance.

M--- Time last periodic maintenance was completed.

RCRW-- The crew which performs regular maintenance on a fail and fix basis.

RT--- Workstation for repair by the regular repair crew.

DUM--- Facility to generate the succeeding periodic maintenance (PM).

D--- Holding area for transaction carrying time for next PM; the time transaction is held equates to the interval for PM.

DU--- Work station where periodic maintenance is performed.

COMP Facility to compute statistics for system availability.

DT The length of time the system was down based on predetermined criteria.

SYSD Time the system went down.

SYSU Time the system came back into action.

APPENDIX C

NOTES TO THE USER

C.1 Expanding and Collapsing the Module's Size

The current configuration of the program, Appendix A, has two flowlines, two subsystems in each flowline, and a maximum of two units for each subsystem. The program can be expanded or collapsed to conform to other module configurations. The unit is the small entity of the subsystem; the subsystems in the same series form a flowline; and, all parallel flowlines make up the module.

C.1.1 Expanding the Size of a Subsystem

The number of units in a given subsystem is set by placing it in the INITIAL statements identified as N--, where N is followed by the identifier for the subsystem. For example, if subsystem A2 has two units, the INITIAL statement is

```
INITIAL  NA2, 2 .
```

After indicating the number of units, for any number greater than two, the production and failure and repair routines are placed in the program.

C.1.1.1 Production. All units added to the first subsystem in any flowline have the same statements as A12, Lines 277 through 292, for production. All statements identifying the subsystem A12 are replaced by the proper subsystem. The addition of units to subsequent subsystems require the same statements A22, Lines 317 through 331. The

additional units are inserted following the last unit's routine for production.

To collapse the subsystem, all units not desired are removed, provided the user is certain those units are never included for testing. If they are to be tested, but it is not desirous that a given number be active in the current run, the number in the INITIAL statement is used to deactivate a unit. To allow the buffer to be replenished, each deactivated unit, greater than the second, requires the same type statements as Lines 238 and 239, where BV$---FK is the Boolean variable.

C.1.1.2 Failure and Repair Routine. The failure and repair routine, for units other than the first unit, are the same as the statements for A12, Lines 501 through 563. Those modules without periodic maintenance can remove statements 539 through 563 and change all pertinent statements to reflect the same repair crew for a given subsystem. In addition to deleting statements, the block of statements 37 through 40 are set for any time greater than the length of the run.

The time to failure and time to repair densities are added for each additional unit. They are inserted at any point between statements 2 and 210.

C.1.2 Increasing and Decreasing Subsystems

Additional subsystems are established by inserting an INITIAL statement for the given subsystem with the number of units in the subsystem. The units are linked up with a buffer before going to the next subsystem or out of the module. The subsystem configuration follows the sequence of statements for A2 in production, and failure and repair. The buffer associated with that subsystem requires the

insertion of a capacity as in Line 29.

Subsystems may be collapsed using the following procedures:

(1) Set the number of units equal to one

(2) Set production time at zero

(3) Fix the time to failure greater than the length of the production run.

A second alternative is to remove all statements pertaining to that subsystem.

C.1.3 Changing the Number of Flowlines

The current program, Appendix A, has two flowlines, A and B. Additional flowlines are created in the same manner as new subsystems, with one exception. The first subsystem of the flowline has the same statements as A1 for both production, and failure and repair.

The link-up routine is altered by having the COMPARE statements, as in Lines 334 and 336, changed to reflect the comparison for all other flowlines. Thus each flowline contains statements as those in Lines 333 through 340, and the routine in flowline B will compare all other flowlines in Lines 435 and 437.

Flowlines are removed from the module by deleting all statements pertaining to that flowline, and by removing all inquiries in the link-up routine.

C.2 Changing Other Configuration Constraints

The production times may be identified as distributions rather than constants, as they are in Appendix C. The time to failure and time to repair can be distributions other than exponential. The repair policy can also be other than as it appears in the program.

The time required for production by each subsystem is presently set as a constant. The INITIAL statement can be changed to a function format as in Lines 2 through 17. The function can be anyone covered by the GPSS language. The function is called upon in the production routine via a HOLD block. The identification of the function is placed in parenthesis following TIME, as in Line 464.

The time to failure and time to repair distributions are changed by replacing the present density, Lines 2 through 17, with the desired density. Repair crews currently servicing one subsystem can be assigned to service additional subsystems by designating a given repair crew to be seized by the desired units.

BIBLIOGRAPHY

1. Applebaum, Sidney P. "Steady-State Reliability of Systems of Mutually Independent Subsystems." I.E.E.E. on Transaction of Reliability, XIV (1965), 23-9.

2. ARINC. Reliability Engineering. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1964.

3. Atlanta Constitution. "Tech Anti-Inflation Plan Pushed." September 6, 1974, sec. A, 2.

4. Barten, K. A. "A Queueing Simulator for Determining Optimum Inventory Levels in a Sequential Process." Journal of Industrial Engineering, XIII (July-August 1962), 245-52.

5. Box, G. E. P. and Hunter, J. S. "The 2^{K-P} Fractional Factorial Designs." Technometrics, III (August 1961), 311-51.

6. Box, George E. P. and Jenkins, G. M. Time Series Analysis Forecasting and Control. San Francisco: Holden-Day, Inc., 1970.

7. Buzacott, J. A. "Finding the MTBF of Repairable Systems by Reduction of the Reliability Block Diagram." Microelectronics and Reliability, VI (1967), 105-12.

8. Buzacott, J. A. "Methods of Reliability Analysis of Production Systems Subject to Breakdowns." Operations Research and Reliability. New York: Grouchko, Gordon & Breach Science Publishers, 1971.

9. Caplen, Rowland. A Practical Approach to Reliability. London: Business Books Limited, 1972.

10. Drenick, R. F. "Mathematical Aspects of the Reliability Problem." Society for Industrial and Applied Mathematics Journal, VIII (1960), 125-49.

11. Epstein, B. and Hosford, J. "Reliability of Some Two-Unit Redundant Systems." Proceedings of the Sixth National Symposium on Reliability and Quality Control (January, 1960), 469-88.

12. Grauer, Robert T. "Characterization of Educational Dynamics." Unpublished Ph.D. dissertation, Polytechnic Institute of Brooklyn, Brooklyn, New York, 1972.

13. Green, A. E. and Mourne, A. J. Reliability Technology. Bristol, England: John Wiley & Sons LTD., 1972.

14. Gnugnoli, Giuliano and Maisel, Herbert. Simulation of Discrete Stochastic Systems. Chicago: Science Research Associates, Inc., 1972.

15. Heginbotham, W. B. and Rippon, G. E. "An Analytical Model Predicting Fixed Index Assembly Machine Performance." Journal of Engineering for Industry, XCIV (May 1972), 419-24.

16. Hicks, Charles R. Fundamental Concepts in the Design of Experiments. New York: Holt, Rinehart and Winston, Inc., 1973.

17. Hines, William W. "Notes for Systems Engineering 411, Systems Analysis II." Unpublished notes, School of Industrial and Systems Engineering, Georgia Institute of Technology, November 1968.

18. Hines, William W. and Montgomery, Douglas C. Probability and Statistics in Engineering and Management Science. New York: The Ronald Press Company, 1972.

19. Knight, C. R., Jervis, E. R. and Herd, G. R. "The Definition of Terms of Interest in the Study of Reliability." I.R.E. Transactions on Reliability and Quality Control (April 1955), 34-56.

20. Law, S. S., Baxter, R. J. and Massara, G. M. "Analysis of In-Process Buffers for Multi-Input Manufacturing Systems." Unpublished report, Metal Stamping Division, Ford Motor Company, Dearborn, Michigan, May 1973.

21. Lloyd, David K. and Lipow, Myron. Reliability: Management, Methods and Mathematics. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1962.

22. Mihram, G. Arthur. "Some Practical Aspects of Verification and Validation of Simulation Models." Operational Research Quarterly, XXIII (January 1972), 17-29.

23. Mutsenek, K. Ya. and Lobzov, B. A. "The Reliability and Productivity of Automatic Assembly Machines." Russian Engineering Journal, XLVI (1966), 50-53.

24. Naylor, T. H. and Finger, J. M. "Verification of Computer Simulation Models." Management Science, XIV (October 1966), B-92-B-101.

25. Naylor, Thomas; Balintfy, Joseph; Burdick, Donald; and Chu, Long. Computer Simulation Techniques. New York: John Wiley & Sons, Inc., 1966.

26. Nelson, Charles R. Applied Time Series Analysis for Managerial Forecasting. San Francisco: Holden-Day, Inc., 1973.

27. Pritsker, Alan B. and Kiviat, Phillip J. Simulation with GASP II. Englewood Cliffs: Prentice-Hall, Inc., 1969.

28. Smith, David J. Reliability Engineering. New York: Pitman Publishing Corporation, 1972.

29. Swanson, Douglas. "A Newly Developed Assembly System Permits Maximum Equipment Utilization, Uptime, and Productivity." Microtecnic, XXVI (1972), 457.

30. Thomason, R. An Introduction to Reliability and Quality. Brighton, Sussex: Machinery Publishing Company, Ltd., 1969.

31. Van Horn, Richard L. "Validation of Simulation Results." Management Science, XVII (January 1971), 247-58.

32. Weber, P. A., Jr. "A Model for Determining the Effect of In-Process Storage on the Output of a Series of Machines." Unpublished Master's thesis, Georgie Institute of Technology, 1962.

33. Young, H. H. "Optimization Models for Production Lines." Journal of Industrial Engineering, XVIII (January 1967), 70-78.