To the Graduate Council:

I am submitting herewith a thesis written by Nayef K. Alteneh entitled "Contribution of Cellular Manufacturing Design To The Reduction Of Production Lead Time Utilizing Simulation Modeling." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Dr. Rupy Sawhney, Major Professor

We have read this thesis and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Nayef K. Altemeh

Date 12/1/94

CONTRIBUTION OF CELLULAR MANUFACTURING DESIGN TO THE REDUCTION OF PRODUCTION LEAD TIME UTILIZING SIMULATION MODELING

A Thesis

presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Nayef K. Alteneh

December, 1994

ACKNOWLEDGMENTS

My sincere thanks goes to Dr. Rupy Sawhney for his guidance and support throughout the preparation of this thesis. His invaluable contributions to my studies are unsurpassed and of a great value to me. I wish also to extend my sincere thanks to Dr. Denise Jackson and Dr. Wayne Claycombe for their superb commitment to teaching and assisting students. They have inspired me throughout my Industrial Engineering studies. I wish also to extend my thanks to my wife Tina for her everlasting support, devotion and understanding throughout my school work.

ABSTRACT

The ever-increasing competition in the market place is the driving force behind the rising emphasis on embracing Cellular Manufacturing Systems and their methods in effort to shorten the product lead time. One such method is Lean Production Systems (LPS) which are JIT based production systems that consist of manufacturing cells as their building blocks. The research in this thesis addresses and assesses the impact of some of the critical parameters involved in the design of these cells on cycle times.

Simulation models are developed for various situations to evaluate several issues that are believed to influence the performance of manufacturing cells. These include process/product variation, the product mix, demand increases, setups under random arrivals, SOH, and defect rates. Cell cycle times and line efficiency are the response variables used to measure the cell's performance. The results of the simulation models have shown that product cycle time increases significantly with increased process variation, increased demand, defect rates, and stochastic arrivals which increases setups. While cycle times have increased, the cell's line efficiency has statistically decreased. Other factors such as the product mix, SOH, and increased product variation are shown to have no significant impact on cycle times and the cell's line efficiency was also shown to improve under these conditions. An overall improvement in the cell's performance is also shown to occur with minimized variation in the cell's process.

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION 1

Problem Background 1
Problem Statement 2
Objective 4
Methodology 7
Summary of Chapters 8

2. BACKGROUND 9

Manufacturing Cell Types 9
Cell Design Criteria 10
Cellular Manufacturing Systems 13
Decouplers 14
Types of Models Available for The Analysis of CMS 16

3. THE PHYSICAL SYSTEM AND ITS PARAMETERS 19

System Description 19
Changeovers 25
Setup Improvements and Activities 28
Quality Inspection 33
Operational and Synchronization Activities 35

4. SIMULATION MODELS AND ASSUMPTIONS 40

Objective 40
Initial System Characteristics 40
Assumptions and Parameters 41
The Base Simulation Model Description 43

5. CELL DESIGN SIMULATION EXPERIMENT PERFORMANCE ANALYSIS 48

Objective 48
Base Model 48
Effects of Reworks On Cycle Times 53
Changeover Impact On Cycle Times 56
SOH Impact On Cycle Times 59
Product/Process Variation Impact On Cycle Times 64
Demand Increase Impact On Cycle Times 73
Increased Product Mix Impact on Cycle Times 78

6. RESULTS 85

LIST OF REFERENCES 93

APPENDICES 98

APPENDIX A. Base Simulation Model 99

APPENDIX B. Simulation Model With Changeovers 118

APPENDIX C. Simulation Model With Increased Product Mix 139

VITA 165

LIST OF FIGURES

FIGURE PAGE

1. A Production System With Kanban Links..... 3
2. Bird-Cage Layout..... 11
3. Linear Layout..... 11
4. Isolated Island Layout..... 11
5. Cell Layout With Five Machines and Inspection Stations..... 20
6. Inspection vs. Prevention..... 34
7. A Cell Designed To Machine A Family of Parts Showing Operator Moving Opposite To Part Flow..... 36
8. Parameters of Random Variables Used In The Cell..... 44
9. Base Model Output..... 51
10. Base Model With Reworks..... 55
11. Base Model With Changeovers..... 57
12. SOH Impact On Cycle Times..... 61
13. SOH Impact On A Cycle Time..... 63
14. Base Model With 5% Product Variation..... 65
15. Product Variation Impact On A Cycle Time..... 68
16. Process Variation Impact On Cycle Times..... 69

17. Reduced Process Variation Impact On Cycle Times 72

18. Demand Increase Output 74

19. Demand Fluctuation Impact On Cycle Times 77

20. Product Mix Output 79

21. Improved Process/Product Mix 82

22. Product Mix Impact On Cycle Times 83

23. Product Mix Impact On Changeovers 84

24. Pareto Chart Shows Percent Impact On Cycle Times 90

CHAPTER 1

INTRODUCTION

1.1 PROBLEM BACKGROUND

There is a current emphasis on reducing the product lead time in order to respond to future opportunities as identified by the dynamic nature of customer's demand. A trend in industry to shorten the product lead time is by implementing lean production systems (LPS) which are basically one method of cellular manufacturing systems that utilize pull production- known as JIT production. Lean production systems which desire synchronous production seek to shorten the product lead time by continuously improving and minimizing the cell cycle times to be able to meet the product lead time. The cell cycle time can be defined as the sum of all times of work elements (activities) which are performed at a work station in order to produce a single unit. Efforts to improve and reduce these activities not only reduce the cell cycle time, but will also contribute significantly to the reduction of the product lead time.

Figure 1 represents a "Pull Production System" with manufacturing cells. In this system, cell 3 dictates the required production activities that need to be performed at cell 2, and subsequently, cell 2 dictates the requirements for cell 1. Each subsequent cell in this pull production system withdraws material from the preceding cell utilizing "kanban links" to allow for material and information control between the cells while allowing for production smoothing which is maintained by closely equating the cell cycle times. Pull production systems are in essence linked production cells that require each of its component cells to be in an ideal state of synchronization; that is each cell cycle time must not exceed the product cycle time.

1.2 PROBLEM STATEMENT

The underlying objective in LPS is to equate the cell cycle times such that each cell cycle time does not exceed the demand requirement of the product cycle time (the maximum cell cycle time). Theoretically stated, the objective is to maintain cell cycle times in balance such that: $CT_1 = CT_2 = ...CT_i = CTmax$, where i is the number of cells in the system, and CTmax is the maximum cell cycle time in the production system. This is the product cycle time that corresponds to customer's demand. Product cycle time can be defined then as the maximum time a cell takes to make a product available as imposed by the customer demand, this

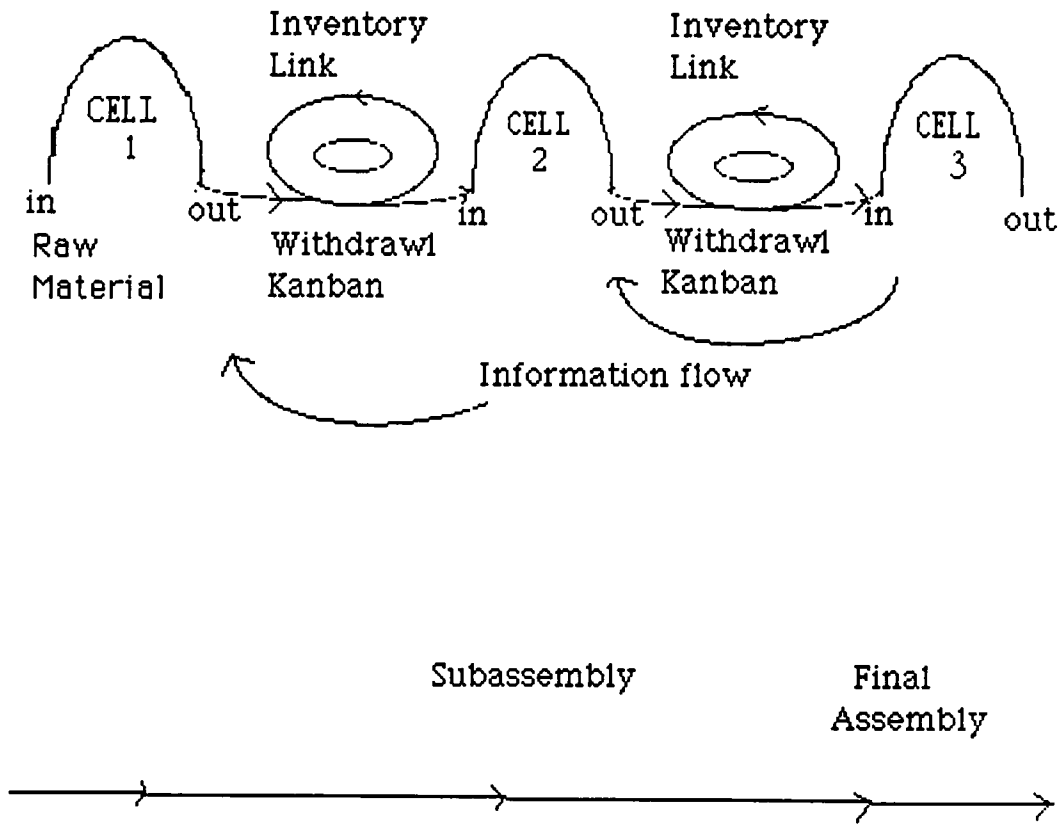


Figure 1. A production system with Kanban Links [20].

includes all the activities necessary to move, process, and inspect the part.

— The underlying critical factor in achieving the system objectives is to be able to adjust cycle times of the different cells in such a way that each cell cycle time does not exceed the production system required demand cycle time. However, the inevitable dynamic changes imposed on cell activities often modify these cycle times thus causing unnecessary delays, blocked order requests and ultimately increasing the product lead time.

1.3 OBJECTIVE

The objective of this research is to identify and test perceived critical parameters involved in the design of cells to assess their impact on pull production systems and to develop a cause and effect knowledge. More specifically, the research objective is to assess the impact of each parameter on cell cycle time, the product cycle time, and line efficiency. Furthermore, the following hypotheses will be tested:

Hypothesis 1: Cell design allows quality to be integrated throughout the cell process. Quality is maintained so only defect-free parts can be moved on to the next process. Integrating quality throughout the

process makes the final quality check somewhat unnecessary. However, the lack of quality checks during the process increases the chances of reworks at the final process stage. In pull production cells, cycle times are mainly dependent on the last activity center in the cell, presence of reworks at the final process stage significantly increase cycle times.

Hypothesis 2: The design flexibility in pull production cells allows for minimal setup activities from one process to the next. Under stochastic part arrivals, however, setups activities increase significantly thus adversely impacting cycle times.

Hypothesis 3: Maintaining an uniform SOH of one unit size between machines is a necessary component in pull production cells and it often acts as a cushion to dampen process variation, machine breakdown, and, to some extent, demand fluctuation. Increasing SOH to more than one unit, however, may not always improve cycle times and it will merely increase WIP inventories.

Hypothesis 4: Increased variation in processing times adversely impact cycle times. In pull production systems, balance is maintained through controlling and minimizing variation between operations. Variation could be either product or process related. Process variation results from variation between machine processing times on the different parts within a family.

Hypothesis 5: Pull production cells design has a limited flexibility for product variation. Increasing product variation beyond the cell limit increases cycle times. Product variation results from the different sizes and shapes of parts within a part family.

Hypothesis 6: Increasing the cell demand rate could increase cycle times. In pull production cells, the demand rate is increased by increasing the order arrival rate.

Hypothesis 7: Increasing the product mix within a part family could adversely effect cycle times. This would occur since increased part family translates in

increased time loss in changeover activities from one part type to the next.

1.4 METHODOLOGY

To assess how critical perceived parameters could impact cell cycle times, and to understand the relationship between cell design criteria and the principles of production smoothing, a manufacturing cell and its activities with the initial conditions will be defined. The cell will form the base model for our experimentation.

Simulation models will be developed based on three-activity categories that include operation and synchronization activities, quality inspection activities, and changeover activities. Operational and synchronization activities include those activities involved in processing parts, loading/unloading and preparing machines between operations. Quality inspection activities include those activities involved in inspecting parts' critical dimensions, and activities associated with reworks and scraps. Changeover activities are those activities that prepare machines for the next part type after the previous part has been processed. An initial simulation model will be developed to simulate all those activities under the first category; "Operation and Synchronization Activities." This model will evaluate cell performance in terms of cycle time without the impact of changeover activities or reworks.

1.5 SUMMARY OF CHAPTERS

This research paper has been segmented in the following manner:

Chapter 1: Provides the introduction, background, objectives, and methodology intended to evaluate a set of hypotheses.

Chapter 2: Provides a background review about cellular manufacturing and simulation modeling.

Chapter 3: Defines the pull production cell to be evaluated, its parameters and activities.

Chapter 4: Describes the initial simulation model.

Chapter 5: Provides the experiment and revisions of the initial simulation model to test above hypotheses and other cell parameters.

Chapter 6: Results and conclusion.

CHAPTER 2

BACKGROUND

2.1 MANUFACTURING CELL TYPES

Manufacturing cell types can be classified in the following manner:

1. Single machine cell: This type of cell contains only a single processing machine, along with its tools, jigs, and fixtures.
2. Group machine cell with manual handling: These cells contain more than one machine tool. The system is designed with similar machinery, tooling, and operations. Material handling is done manually.
3. Group machine cell with semi-integrated handling: Here, several machine tools are contained in the cell, with parts transferred from one machine to another by means of some mechanical system (such as, a conveyor belt).
4. Flexible manufacturing cells: A typical flexible manufacturing system is usually consist of computer-controlled machines, a loading/unloading station and associated storage buffer, connected by an automated

material handling system. Material handling devices serve to transport parts within and in and out of the cells. This system is used to manufacture raw material to finished products in one series of integrated processes. A minimum of human intervention is required in the manufacturing process. Human effort is mainly needed for maintenance and supervision of the general operation.

And since our production system in this research operates with manual handling utilizing pull production, a group machine cell with manual handling is chosen to represent our base cell model.

2.2 CELL DESIGN CRITERIA

One key criteria in cell design is the physical layout of the cell. In the past, manufacturing cells have been designed in the following manner:

1. Bird-cage layouts, Figure 2.
2. Linear layouts, Figure 3.
3. Isolated Island layout, Figure 4.

Although there are some advantages associated with these layouts, there are many disadvantages that include: excess WIP inventories, longer lead times, unbalanced work loads, inability to smooth production, and inability to adapt to increased demand or

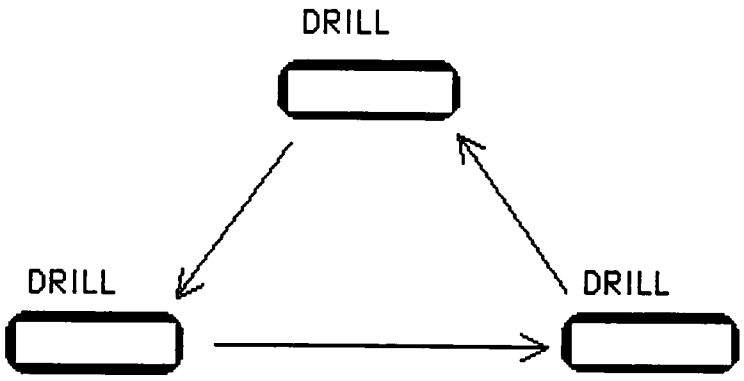


Figure 2. Bird-Cage layout.

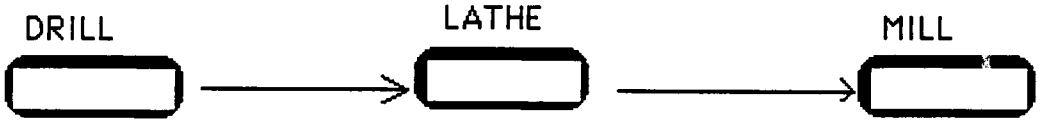


Figure 3. Linear layout.

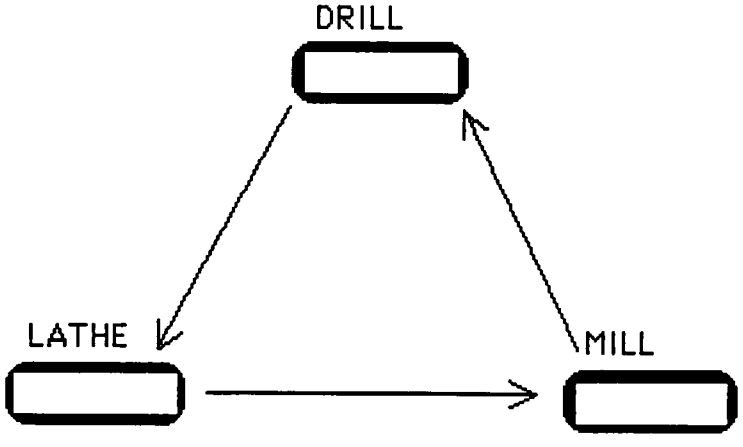


Figure 4. Isolated Island layout.

physical changes. In addition, these type of layouts are inefficient because of the many non-value added activities performed with carrying inventories.

— One alternative approach used to eliminate many of the disadvantages associated with the above layouts is U-shaped cells. One characteristic of these cells is that they are designed where raw material comes in at one end and flows in a clockwise fashion, with no backtracking, to the other end. The necessary elements in forming these cells include material flow and labor control in order to maintain flexibility for demand changes or future expansion. Through the use of inventory links and information for material control, many non-value added activities such those associated with carrying inventories are eliminated. A u-shaped cells must also consider the following criteria:

— ***Product Selection*** - This includes the selection of the product and the number of part families to be manufactured in the cell. A design criteria here is to determine the number of the product mix to be manufactured. An increase in the variety of parts could adversely affect flow times through the cell. This would occur since more part variety leads to increased time loss as a result of frequent changeovers from one part type to the next.

— ***Tooling and Setup*** - To be able to reduce cycle time, the adoption of setup reduction/elimination becomes an integral part in the cell design. Utilization of SMED techniques and

standardization of tools and fixtures could improve setup times and minimize setup activities.

Key Machines - The formation and selection of the part families to be manufactured in a cell is usually dependent on the identification of one or more key machines. The idea is to choose one or more key machines which form the heart of the process. The volume of parts necessary to keep machines loaded and operating close to 100% of the time is then determined. Secondary machines representing lower investment can then be allowed to operate at a smaller utilization rate.

2.3 CELLULAR MANUFACTURING SYSTEMS

A cellular manufacturing system consists of manufacturing cells where each cell is part-oriented and can be designed by implementing group technology and JIT pull systems. A typical cell is U- shaped designed to produce families of parts and usually contains programmable machine tools and trained production workers, each of whom operates more than one piece of equipment within the cell. The cell grouping differs from traditional equipment arrangements in job-shop environments where, historically, equipment has been grouped together according to similar function in what is called a functional layout. In a cellular layout, groups of functionally dissimilar machines are grouped together in such a manner that each cell can complete the

processing of a part family. Unlike flow shops, cells are designed for flexibility and moderate production rates.

The justification for adopting CMS is usually based on two assumptions. One assumption is that setup time is significantly reduced or eliminated between successive jobs within a part family. And secondly, part scheduling is assumed stable and predictable, so part arrival rate can be controlled. To realize these benefits and others, it is critical to understand the factors that influence the design of the system in order to evaluate the success of the system in terms of the cell performance measures such as cycle time.

2.4 DECOUPLERS

One characteristic of pull production cells is that stations have decouplers, the term decoupler was coined by Black & Schroer [19]. These decouplers act as part facilitators with specific input and output points and they can hold more than one part, depending on the variety of parts within a family.

Inventory in these buffer stations is small and it is necessary to reduce process dependency. This allows the operators to delay the process to provide machine adjustments, machine cleaning, and other various off-line activities needed to maintain cell quality and machine function.

This material held within the cell, called the stock-on-hand (SOH), is essential to starting the work within the cell, it includes all material processed in the machines as well as those units in any decoupler between two machines or operations. The necessary number of SOH is the minimum number of pieces necessary to allow work to progress in the cell, without the cost of excessive waiting and high WIP inventories.

Another function of decouplers is to perform part inspections. Each part processed in the previous machine has to go through inspection before it moves to the next process. "Decouplers inspect critical dimensions of parts so that only good parts are pulled to the next station. Such inspections are done automatically. This kind of 100% inspection is referred to as 'poka-yoke' or 'autonomation,' (Black & Schroer, 1988)."

In flexible manufacturing systems, decouplers are used to reduce robot handling and improve robot process capability. For example, part manipulation (reorientation) may require a turned part to be flipped end-over-end. Decouplers can locate a part for the robot griper, manipulate or reorient parts for the robot to insert in the next machine. Gravity chutes attached to the decouplers move the parts from and toward machines.

A pull cellular system is designed to produce parts exactly at the rate the subassembly area needs the parts to avoid increased WIP inventories. Overproduction will result in the need to store parts, transport parts to storage and keep track of the

stored parts (paperwork) and so on. This takes time, requires people, and adds cost. Historically, WIP inventories were justified since they provide buffers between operations to maintain maximum utilization of machines. On the contrary, pull systems adopt low WIP and machine under-utilization. The rationale is that decreased inventories will decrease lead times, expose problems, and reduce labor and costs.

2.5 TYPES OF MODELS AVAILABLE FOR THE ANALYSIS OF CMS

There are three basic types of models available for the analysis of cellular manufacturing systems. These are:

1. Static or Deterministic models
2. Queuing models
3. Computer simulation models

Static models use a set of deterministic equations to calculate gross estimates of such system parameters as throughput time, machine utilization, capacity, etc. These models are generally incapable of providing good estimates of more probabilistic operating characteristics such as queue buildup and simultaneous failures [28]. These occurrences tend to impair system performance, and these type of models often ignore such occurrences.

Queuing models are based on the mathematical theory of queues. These models permit the inclusion of some of the dynamic characteristics of the system (notably, the fluctuations in queue levels). These characteristics are complex and probabilistic in nature, however, queuing models typically predict these aspects of system performance only in a general way, and for simple situations.

Simulation is the operation of computer models of systems for the purpose of studying phenomena expected to occur during their operation. Computer simulation models estimate the behavior of a manufacturing system by modeling the interactions of the system components and their activities. Simulation may be incorporated in the design process for specifying system characteristics or it may be used to verify the performance of a proposed system after the specification process is complete. These characteristics can be modeled as random distributions in the simulation model, and they may be incorporated in the model relatively easily (as compared to the other types of models) so that a high degree of realism can be achieved. The difficulty lies in the collection of a sufficient amount of data (which can be quite expensive) and in the selection of proper distributions to model the activities in the system.

Because of the complexity of the activities occurring during the operation of a typical manufacturing cell, and because of the randomness inherent in many of these activities, simulation

is the most suitable tool for modeling these systems. SIMAN (**SIM**ULATION **AN**ALYSIS) is a general purpose language is chosen to model the cell under study.

The SIMAN framework requires two files as input. The model frame is a coded model of the physical attributes of the system and their logical interrelationships. The experimental frame describes the experimental conditions under which the simulation will be executed. The experimental model allows the manipulation of the experimental conditions of the system to allow for different runs and desired outputs. The model and the experimental frames are linked together to generate the simulated response of the system under the specified conditions [13].

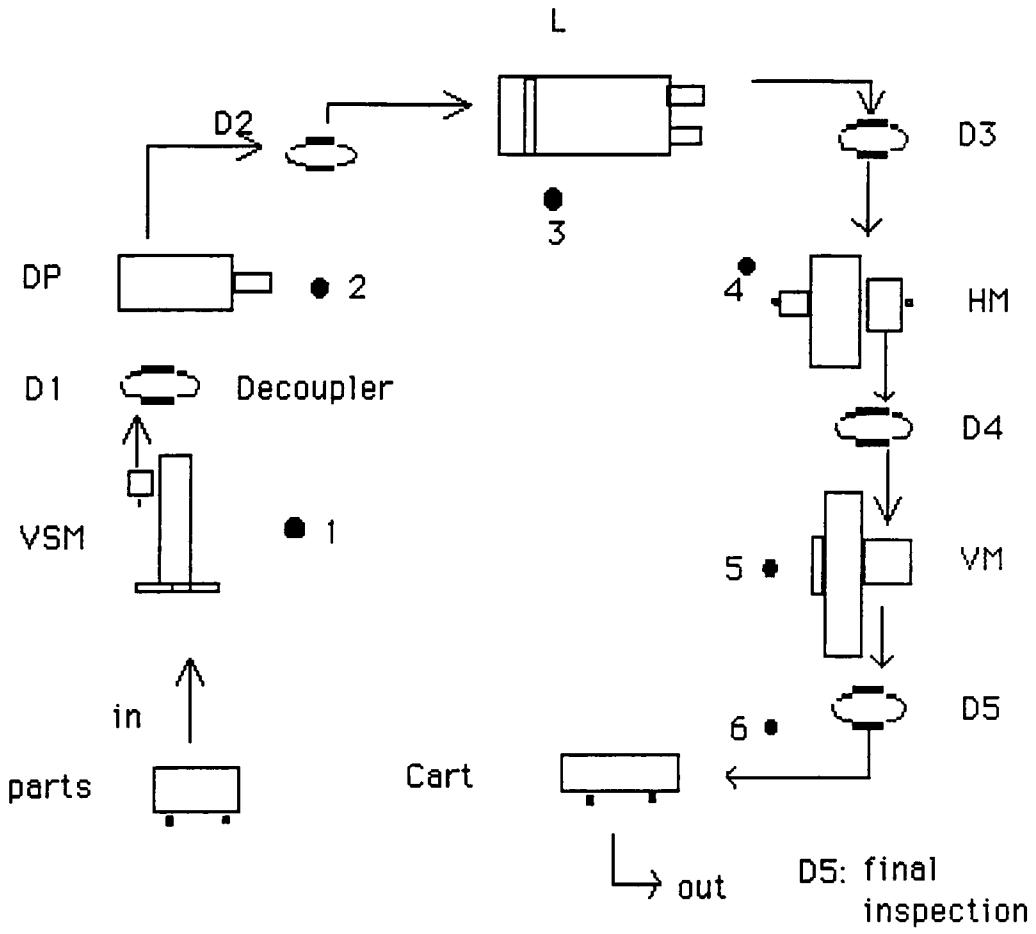
CHAPTER 3

THE PHYSICAL SYSTEM AND ITS PARAMETERS

3.1 SYSTEM DESCRIPTION

Figure 5 shows a typical cell with 5 machines and inspection stations (or decouplers) in between. The system includes, 1 vertical spindle machine, 1 drill press, 1 lathe, 1 horizontal milling machine, and 1 vertical milling machine. Each of these machines normally has a different but close processing times. In this cell raw products come in a cart where they are initially processed at the spindle machine, the part proceeds through several operations and ends up at a final inspection station where parts are collected in an empty cart.

A completion of a loop in the cell signifies the cell cycle time. This time is determined by the rate that the subassembly system needs the parts this system makes. The cycle time spent on a part is the sum of the manual time the operator spends at the machines, processing times, plus the time spent in moving parts from one machine to the next. The manual operation time is the time required for the operator to load/unload the machine,



key

VM = Vertical milling machine

HM = Horizontal milling machine

L = Lathe

DP = drill press

VSM = vertical spindle machine

Material movement path

Worker position

Inspection Station

Figure 5. Cell layout with five machines and inspection stations.

SEE P150 P-25

position the part, perform any assembly or inspection, and start the machine.

In a pull system cell, the cell can operate with a variable number of operators, the number of workers depend on the size of cell and the functions to be performed. For a cell with few machines, one multifunctional worker can walk around the cell, from one machine to the next, where he can do several tasks that include loading and unloading, inspecting, etc. As the operator moves in the cell throughout a cycle, he/she completes a part. The time required to complete one unit is referred to as the cycle time (CT).

The theoretical objective in such a system is to balance tasks so they will not exceed the total cycle time of the cell. One key factor in controlling task time is by establishing standard-operation procedures [21]. This technique seeks to achieve work balancing in the manufacture of a product, establish a work sequence, and control inventory at a minimum quantity that is necessary to perform the work sequence. The traditional method of providing task balancing can be summed up in the following steps:

A. The first step is to find the demand rate time. This is the time found from dividing the total daily available time (less the time when the equipment is not operating) by the daily customer requirements. For example, if the daily operating

time less (breaks, changeovers, and etc.), is 800 minutes 60 seconds, and daily demand is 800 units, then the demand rate time is 60 seconds. This is the cycle time required to process one unit to meet demand.

B. The second step includes determining operator's manual time required to load/unload the machine or fixture, position the part, perform any assembly inspection, and to start the machine. This manual time can be measured with a stop watch and does not include any walking time between machines.

C. The third step involves determining machine processing time. This is the time needed by the machine to complete one unit unattended. This time is measured from the time the operator initiates the cycle until the machine stops. Completion time is the sum of the manual and machine processing time. It is the time required for a single operation to be accomplished. For example, if the manual time for the first operation is 20 seconds and the processing machine time is 30 seconds, then the completion time is 50 seconds. Production capacity for each station can then be determined by dividing the daily operating time by the completion time of a part; this is the maximum daily

capacity for each machine in the cell; (800 * 60)/50 = 960 parts/day.

D. The total machine processing time for all operations divided by the demand rate time will give us an indication of the minimum number of machines required in the cell to meet production capacity. For example, if the total machining processing time is found to be 300 seconds, then the minimum number of machines required: 300/60 = 5.

In the same fashion the minimum number of operators can be determined by dividing the total manual time by the demand rate time. This number is subject to change depending on demand changes, operator flexibility and training, and the degree of automation in the cell. In this situation if more than one operator is required, then balancing between operators could be achieved through assigning operators different (stations) where they can walk from one machine to the next. The sum of the operations times for each operator is approximately equal. For the typical assembly line, there would be an operator at each station, and the operating times at each station are equal (balanced). This is not the case for a pull system. The removal of material from the kanban square at station i by the operator at station i+1 is a signal for the operator at station i to proceed with the next operation.

From the above information a standard work sequence could be established for each operator. The cycle time for each operator/machine in the cell and the actual amount of time required to produce a single unit can also be determined. Ideally, cycle time (including walk time) should equal the demand rate time. When both are equal, production output will correspond exactly with customer or next cell demand, this promotes production smoothing, reduces WIP inventories, and achieves lean production system' objectives of cycle time balancing. One other critical objective in pull production cells is to maintain machine times (MT) to be less than product cycle time (CT). In this case, the machine time is the processing time the machine spends to complete an operation on a part without any operator assistance. If the machining time is found to be greater than the cycle time, there are certain methods that can be used to solve this problem. One such method is to "duplicate the process" within the cell, where the worker alternates between the process and its duplicate during each trip through the cell. Other less costly alternatives could include the following:

(1) Adjusting SOH within the cell (between machines).
(2) Adjusting machine processing times.
(3) Increase/decrease the number of operators in the cell.

The cell, see Figure 5, in this paper is designed to fabricate a family of parts where parts are made one at a time in lot sizes of one. Parts removed from the final inspection decoupler (inspection station) dictate what the cell needs to make and when it needs to make it. For example, if part type A is pulled out of the final decoupler, the worker unloads machine number five, removes part type A from decoupler four and loads it into machine number five. This creates the need to replace the part in decoupler four. The worker walks to machine four, unloads a part, places the part in decoupler four. In return, this creates the need to replace the part in machine four and so on. Thus, the parts are pulled through the system one at a time and as needed.

3.2 CHANGEOVERS

A manufacturing cell that is producing a family of parts must have the machines changed over between parts. Changeover occurs when machine preparation is required to process a particular job after the preceding job is finished. The importance of reduced changeovers stems from the fact that it has a significant impact on flexibility, shorter production cycles, and cost. Quick changeovers also deliver a favorable impact on quality. For example, minimizing trial processing reduces material waste: fewer adjustments mean less chance for a variation; smaller batches provide a quicker feedback loop and deliver

fewer defects in inventory should an undetected problem develop.

In a typical cell, changeover activities are handled sequentially. Setup work begins at the head (or start) of the cell and moves through the cell just ahead of the first unit for the part. Setup becomes an integral part of the operation of the cell. When numerous processes are involved in the manufacturing cell of a family of parts, sequential setup changes are utilized. Table 1 describes a changeover from part A to part B for a cell with four processes.

Table 1. Sequential setup from A to B for four processes [12].

Processes	No.1	No.2	No.3	No.4
	A	A	A	A
setup change #1	setup	A	A	A
setup change #2	B	setup	A	A
setup change #3	B	B	setup	A
setup change #4	B	B	B	setup
	B	B	B	B

The worker is moving in the same direction as the part flow and changes each machine over from A to B sequentially. After the worker has made four trips around the cell, the cell is making B parts. The setup times may be long compared to the machining time (the run time) but eventually the setup times should be

reduced to less than the average time the worker spends at each machine.

A typical manufacturing sequence is to setup the equipment or machine tool, run some parts, inspect the parts, adjust the machine, run another sample, measure, adjust, and so forth until an acceptable part is made. This method consists of non-value added activities and generates reworks/scrap, and creates many opportunities for nonproductive time.

Setup procedures are usually thought of as infinitely varied, depending on the type of operation and type of equipment being used. Yet when these procedures are analyzed from a different viewpoint, it can be seen that all setup operations comprise a sequence of steps as follows:

Preparation, after-process adjustment, checking of materials, tools, etc. This step ensures that all parts and tools are where they should be and that they are functioning properly. Also included in this step is the period after processing when those items are removed and returned to storage, machinery is cleaned, etc.

Mounting and removing blades, tools, parts, etc. This includes the removal of parts and tools after the completion of processing and the attachment of the parts and tools for the next lot.

Measurements, settings, and calibrations. This step refers to all of the measurements and calibration that must be

made in order to perform a production operation, such as centering, dimensioning, measuring temperature or pressure, etc.

Trial runs and adjustments. In these steps, adjustments are made after a test piece is machined. The greater the accuracy of the measurements and calibrations in the preceding step, the easier these adjustments will be [12].

Traditionally, setup activities and their percentages are allocated as follows:

1. Preparation, function checks, and movement: 30%
2. Attachment and removal of parts: 5%
3. Centering, dimensioning, setting conditions: 15%
4. Trial processing and adjustments: 50%

3.3 SETUP IMPROVEMENTS AND ACTIVITIES

One important concept utilized in reducing setup time is to convert internal setup operations to external operations. Internal operations refer to setup actions that require the machine to be stopped. External or off-line elements refer to actions that can be taken while the machine is running. These two kinds of elements must be rigorously separated [18]. Once the machine is stopped, the operator should not leave it to handle any of the external setup. As part of the external setup, the die, tools, and materials should be ready for insertion into the machine. Any modification

or repairs to tooling or dies should have been made in advance. In the internal setup, only removal and insertion operations should take place. Importantly, among the elements that can be readily shifted from internal to external are:

Searching time- Trying to find the correct fixture, looking for the right tools, carts, bolts, and so on.

Waiting time - Waiting for skid, carts, or instructions.

Setting time - Setting dies, tooling, fixtures, and so forth.

The activities involved in the setup are sequential and done by the operator in the cell. Utilization of SMED techniques, simultaneous task execution and continuous preparation, and function checks of raw materials, tools and attachment devices reduces setup time significantly in the cell. To perform the setup activities, a cart is used to hold fixtures and tools needed to perform the setups quickly and efficiently. The cart is easy to roll in and out of the way and it is located in a designated area close by the cell.

In the machining cell for shafts, adjustments, for most part, are virtually eliminated through the use of one touch setup changes. For example, if four different lengths of shaft are made on the same lathe. Each shaft must have a hole drilled in it for a locking screw. A stopper is used to automatically control the depth of the hole being drilled. Typically, the position of the stopper on the machine has to be changed and adjusted for each

different shaft. To eliminate this adjustment, a rotary stopper is used that has four different thicknesses of stoppers. The stopper is properly rotated to correspond to the length of the shaft, thus reducing setup times. On the other hand, the vertical milling machine setup is completely eliminated for four milling parts by permanently locating four fixtures permanently on a holding table, eliminating the need for fixture changes setup. This is justifiable since machines or cells process families of parts. Reducing the variety of parts coming to the machines diminish machine modifications so that setup times can be eliminated altogether.

Setup activities included in this cell are described for each machine below starting with the spindle machine. Each activity takes some time to finish, breaking down these activities as such gives a better understanding of how to reduce setup if it is considered excessive and to eliminate those activities that are non-value added by employing SMED techniques and concepts. A description of main setup activities in the cell follows:

1. **Vertical Spindle Machine Setup**

   This operation involves cutting the shaft ends.

   -Detach the old fixture from the fixture-holder plates of the machine.

   -Push a rolling cart over to machine and secure the table next to the machine.

-Push the old fixture onto the cart table.

-Rotate the table and unload the new fixture onto the bolster.

-Pull the table away from the machine and attach the new fixture to the machine using the same set of bolts.

-Return cart to designated area.

2. **Drill Press**

This operation involves drilling a hole for a stationary screw in the motor core shaft. Since cores are of different lengths, stoppers have to be repositioned each time the operation is changed. Typically this procedure requires repeated test runs and adjustments. This problem is eliminated by mounting the needed number of stopper plugs with different heights on a single plate. Activities that take place are:

-At each operation change, the plate is rotated to set the stopper at the desired height.

-The plate is secured.

3. **Lathe Operation**

This operation involves changing four bits at every setup. A tool box is located near the machine.

-Remove the four bits for the previous operation.

-Mount the four new bits.

-Use gauges to adjust bits.

-After repeated adjustments have yielded the correct measurements, begin next operation.

4. **Horizontal Milling Machine**

The operation involves aligning the center of the milling machine cutter with the center of the workpiece.

The processing on the machine involves a constant back-and-forth movement of the table, with a reference line.

-Workpiece is attached to the center of the table, in alignment with the cutter.

-Workpiece is fastened by using clamps.

-Settings are adjusted according to the reference line.

The vertical milling machine setup activities have been eliminated through the use of SMED techniques. On the other hand, Lathe setup activities could be improved by eliminating adjustments and using a standardized plate with different size-bits mounted on it. Spindle setup activities, meanwhile, could be reduced by using a rotating fixture that does not require to be changed between setups. Listing setup activities for the machines in such detail can provide a good tool to understand where long setups actually exist, furthermore, this method can also reveal where the non-value added activities are occurring in the cell.

3.4 QUALITY INSPECTION

A prime requirement of a "pull" production system is "Total Quality Control." Downtime, scrap, reworks and their associated waiting times are reduced significantly in these systems by error preventing, and these are monitored and controlled before the final product level is achieved. Preventing errors from occurring during a process implies "quality by prevention." This requires studying problems and making permanent changes to prevent errors. The more emphasis that is placed on preventing errors, the greater the chance of doing the right activities the first time. In an ideal cell, quality checks are introduced throughout the process and problems are corrected before they reach the end operation. This ensures better products, less costs, and faster yield [1], see Figure 6.

Each processing operation has a corresponding inspection operation that is performed by the decoupler and monitored by the operator. If more than one operation is performed on a part as it passes from one machine to the next, then a corresponding number of inspections will be performed sequentially on that part by decouplers. A description of quality inspection activities follows:

-After each machine processes a part it is automatically inspected by a decoupler.

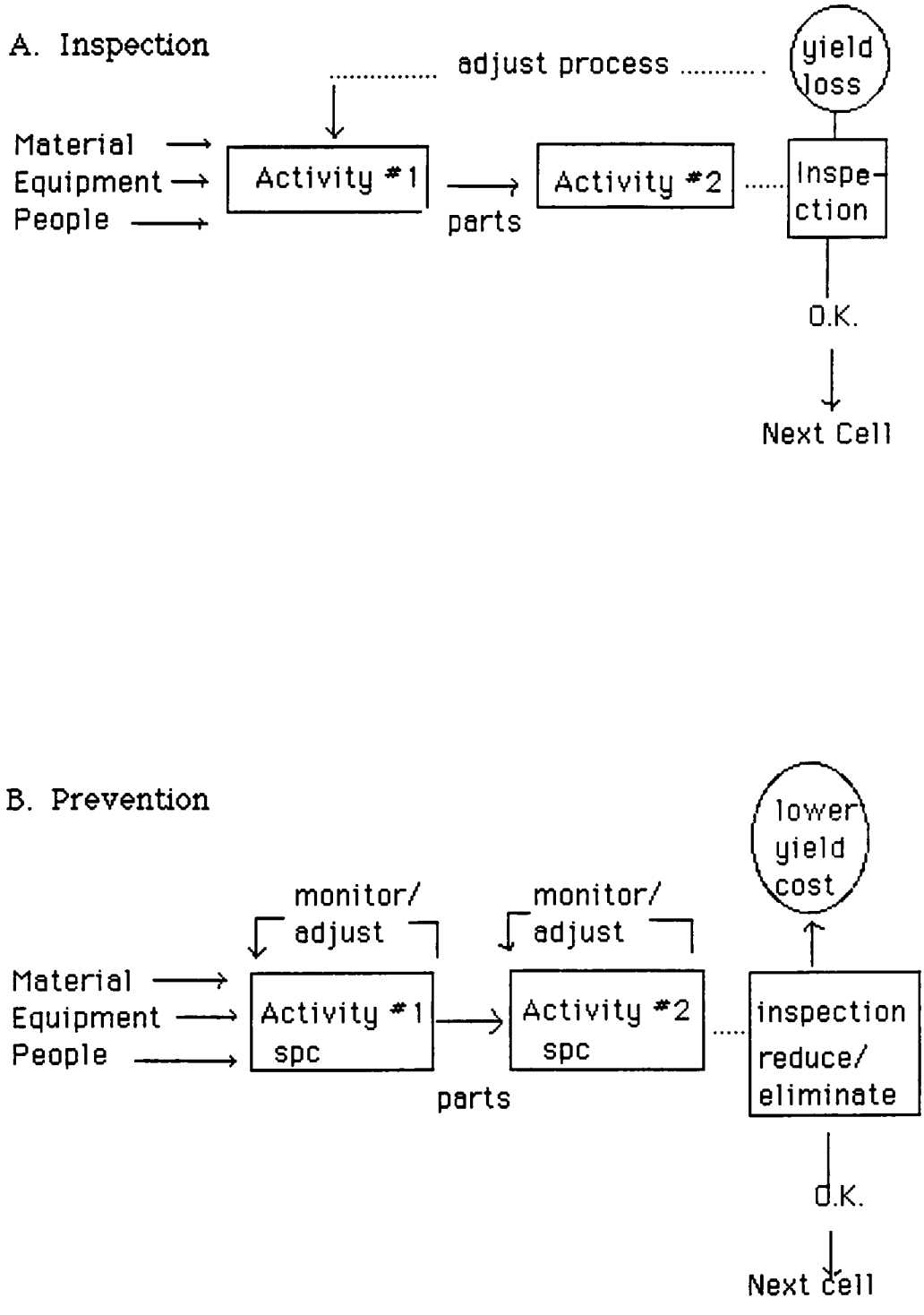


Figure 6. Inspection Vs. prevention [1].

-Defect-free parts are loaded into next machine to be processed.

-Parts failing inspection are carried off by the operator and returned to the proper (previous) processing machine for re-work operation. Parts may be re-worked once before being scrapped.

-If an operator should scrap a part, a scrap ticket is completed and given to a supervisor. The supervisor then replaces the scrapped piece from a controlled area to maintain the lot size.

-Re-work operations are assumed to take the same time as the original processing operation.

-Only defect-free parts are moved from one cell to the next.

3.5 OPERATIONAL AND SYNCHRONIZATION ACTIVITIES

The cell in this thesis is designed for the manufacture of a family of shafts. The raw material for the shafts arrive in a cart. Five machining processes, with inspections, are needed to produce each shaft. Figure 7 shows the cell with the shaft parts in front of machine 1. The cell shows the worker moving in the opposite direction of the parts, and the parts are pulled through the cell by the decouplers. Decouplers can hold all different part types. Parts removed from the final inspection decoupler triggers the following

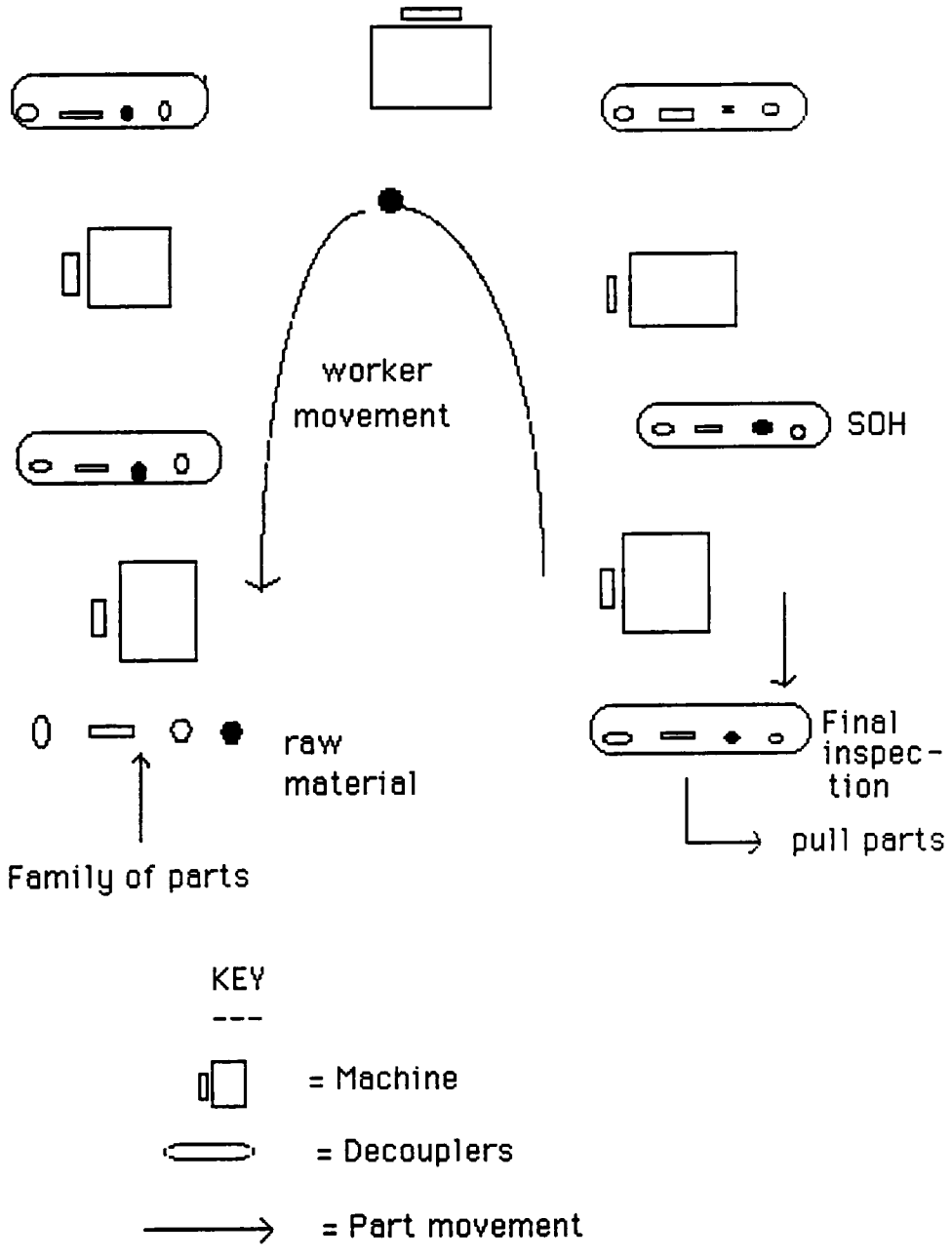


Figure 7. A cell designed to machine a family of parts showing operator moving opposite to part flow.

activities, assuming that parts are held in both machines and decouplers:

Operator:

-checks part type pulled out.

-unloads processed part from last machine (machine 5).

-deposits processed part into final decoupler (decoupler 5) for final inspection.

-unloads same part type from decoupler 4.

-loads part into machine 5 for final process, operator continues to perform other activities while machine completes processing part.

-moves back to machine 4 and unloads part.

-loads part into decoupler 4.

-moves back to decoupler 3, unloads same part type.

-loads part into machine 4, machine processes part.

-moves back to machine 3, unloads part.

-deposits part into decoupler 3.

-moves back to decoupler 2, unloads part.

-deposits part into machine 3.

-moves back to machine 2, unloads part.

-deposits part into decoupler 2.

-moves back to decoupler 1, unloads same part type.

-deposits part into machine 2.

-moves back to machine 1, unloads part.

-deposits part into decoupler 1 for initial inspection.

-moves to part cart.

-unloads same part type from cart.

-deposits part into machine 1.

Synchronization implies: the proper timing of material movement between machines and balancing of operations [18]. This is done by using pull signals for material flow control. Leveling quantities and synchronizing processes can significantly reduce delays and waiting times, thus greatly reducing manufacturing throughput time. For example, using one-piece flow to eliminate lot delays for two serial processes reduces throughput time. Synchronization activities are carried out from one operation to the next, where they become an integral part of the cell operation needed to promote production smoothing and operation balancing. These activities include the following:

-At the completion of each machining cycle, the machine is stopped automatically. All the components and attachments are returned to the start position.

-If there is space to hold a finished workpiece in the output (downstream) decoupler, the workpiece is automatically ejected from the machine to the decoupler. If not, the machine must wait until space is available.

The empty decoupler provides a signal to the upstream machine to make another part.

-When an unprocessed workpiece is available in the upstream decoupler, it is automatically fed to the machine, using guides to position it with human assistance.

-When the unprocessed workpiece is located and clamped into the machine, the next machining cycle is automatically started.

The cycle is repeated every time a part is pulled out of last decoupler. If no part is pulled out, operator performs activities such as machine adjustments, maintenance, and inspection.

CHAPTER 4

SIMULATION MODELS AND ASSUMPTIONS

4.1 OBJECTIVE

The objective of this section is to describe the system under study and the assumptions used. The rationale for the selection of the probability distributions used in the simulation and their parameters are also presented.

4.2 INITIAL SYSTEM CHARACTERISTICS

The system under study was shown in Figure 5. It consists of five machines of which each performs an operation on the part. The operator in the system seizes the machine at station i when it becomes available and unloads finished part i into an automated inspection station (referred to as a decoupler), the final inspection station performs a final quality check before it is pulled out by a subassembly area.

Incoming material is always available at station one where raw material is pulled out as needed. At the end, an empty storage area (empty cart) is available for storing finished parts,

where they can be moved to subassembly systems by using "POK" and "WLK" signals.

Initially, the system is in the idle state. Each machine is empty and idle, part holders between the machines hold one unit of each part type which has already been inspected after being processed on the previous machine. The simulation model starts when a fully processed part is pulled out of the final inspection station.

The operator checks part type pulled out. If part pulled out is the same as previously processed part in the machines, then the previously mentioned activities take place. If part pulled out is of a different part type than the previous processed part, the operator performs sequential setup activities starting at the first machine. Operator seizes machine and performs setup. After completion of first machine setup, operator moves to the next machine performing needed setup activities. After the last machine is setup, operator checks for the parts pulled and initiates the new cycle, otherwise, off-line activities can be performed.

4.3 ASSUMPTIONS AND PARAMETERS

1. The order rate is exponentially distributed. This rate reflects the timing of the release of the order to the production system. A manufacturing cell usually is only a small part of the whole

production system and hence its activity depends on the activity of the subassembly systems. Exponential distribution is often used to model this variable when it is subject to be controlled by external or other component mechanism of the system.

2. Setup is required each time the cell starts to process a different part. Variation in setup times is caused by the different activities performed from one machine to the next. Setups are required due to the differences in part size and shape, and since these parts are from the same part family and the setup activities stay the same from one process to the next, the variation is small. Therefore, the setup times are assumed to follow an uniform distribution for individual machines.

3. The processing times in each machine follow a normal distribution. Since the machines are designed with line-balancing consideration, the processing times in each machine are closely similar, in the simulation model they are initially assumed to be equal, thus processing time at each station follows the normal distribution with a relatively large standard deviation (13%) to account for part type variation.

4. The machines are assumed to have no downtime for the period of time simulated.

5. Transfer batch size between machines is one unit.

6. Since reworking a part takes the same amount of time as the original processing time, then part repair is normally distributed.

7. Load/unload machines for each run follows a normal distribution.

8. Automatic inspection time for each part follows a normal distribution.

9. Operator's movement time between machines follows a normal distribution.

10. Unloading/loading decoupler time requires minimal operator assistance and is considered negligible.

The parameters of the random variables and where they are used in the simulation are depicted in Figure 8.

4.4 THE BASE SIMULATION MODEL DESCRIPTION

The simulation begins with pull signals arriving exponentially every eight minutes. There are four part types that constitute a family of parts. Parts initially pulled follow a discrete probability distribution in a uniform batch size of one each. Arrival (pull) of parts initiates the cycle.

The base model assumes that setup activities are performed whenever a different part is pulled out. If machines were originally setup to process Part A and the next part pulled is of part type B, changeover activities are performed accordingly. Once the cycle is initiated, the simulation model starts by initializing the current simulation time. A signal is created to

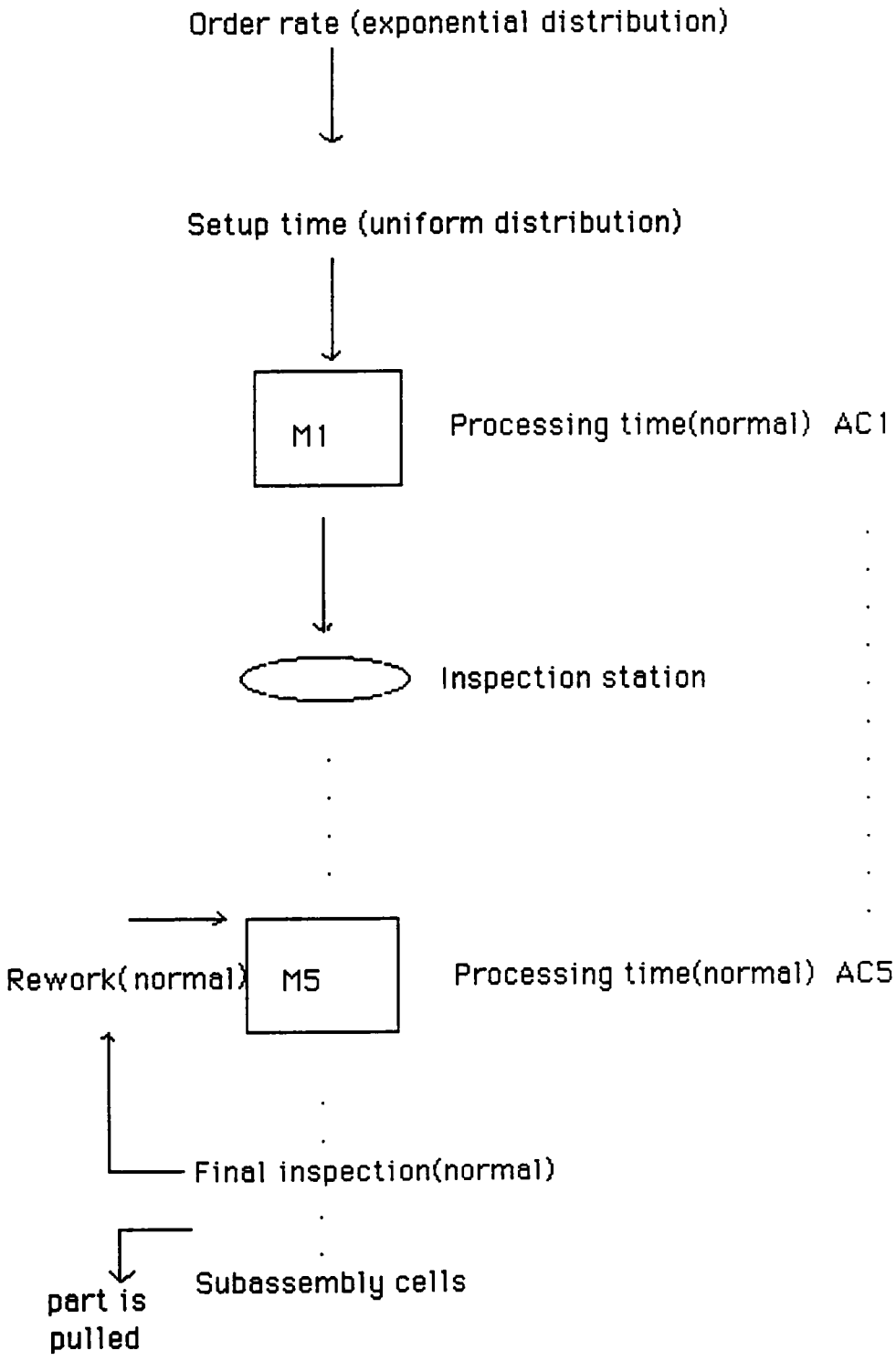


Figure 8. Parameters of random variables used in the cell.

announce the arrival of a part request. The request waits in a queue until it is sensed. Once entity is sensed, it is directed toward the vertical milling machine where it attempts to seize it. If the machine is busy, the entity waits for a signal until the machine is available. If the machine is available, part type is determined at this point and then it is directed to the appropriate block. According to part type, a decoupler resource is searched for part availability in stock, if part is not available, entity has to wait in a queue until the part becomes available in stock. If part is available, the operator is seized at this point, the operator walks to decoupler 4, unloads same part type and then directs the part into machine five (vertical milling machine) to be processed. At this point, decoupler 4 is reduced by one part type, this is shown using the "alter" command in the simulation model. In response to the removal of the part, a signal is sent out to indicate that the part needs a replacement. Using the branch command, one duplicate of the entity is sent out to activity center 4 (AC4), while the original duplicate proceeds to seize the vertical milling machine. Once machine is seized part is processed unattended while the operator moves on to do other activities simultaneously. Once part is finished being processed, it is manually unloaded by the operator and then directed into an inspection station. Operator monitors the inspection procedure and the inspection time is recorded.

Once part is completely inspected it goes through a branch where it is either deposited into decoupler 5 or it is sent to a rework block. If part is good, resource decoupler 5 capacity is then increased by one unit. If part is directed to the rework block, it can only be reworked once before it is scraped, scrap probabilities increase after reworks. At the rework block, time is initialized and the vertical milling machine is set to a busy mode to allow for reprocessing reworked parts, reprocessing time is equal to the original processing time. Once the part is finished, the operator unloads part, time is recorded and the machine mode is set to idle indicating it is available. Once a part is reworked, it is visually inspected, if the problem is corrected, the part is deposited into decoupler 5 where its capacity is increased by one unit, otherwise, the part is scraped. If a part is scraped, a scrap ticket is filled out, time is recorded and the part is replaced by a good one from a controlled area and decoupler 5 is again increased by one unit. In any case, a signal is sent out to indicate the machine's availability and entity is disposed of. Statistics are collected about cycle times, and the process is repeated for each part type.

Once an entity arrives at activity center 4, it attempts to seize the horizontal milling machine, if the machine is busy, the entity waits in the (delaylathe) block until the machine becomes available. If the machine is idle, part type is determined and a check of part availability in decoupler 3 is done. If decoupler 3

does not have the part, then entity has to be held in a waiting block until part becomes available. If part is available, the operator is seized, he unloads part from decoupler 3, removal of part from decoupler 3 decreases its capacity by 1 unit, part is then transported to the lathe machine to be processed. If machine is available, part is processed unattended. Once part is finished being processed, it is unloaded and then directed into decoupler 4, increasing its capacity at this point by one unit, time is observed and statistics are collected about replacement times. The same process is repeated through activity center 3 and activity center 2 for all part types.

Once an entity reaches activity center 1 (spindle machine), it attempts to seize the machine. The entity waits at a delay block if the machine is busy, otherwise, the part type is determined and it is directed to the appropriate block. The attempt to replace the part merely depends on a few activities that include seizing the part cart, retrieving a part, and then loading that part into the spindle machine to be processed. When the part is finished, it is unloaded into decoupler 1, thus increasing its capacity by one unit. Once this is done, a signal is sent out indicating that part has been replaced in decoupler 1, entity is then disposed of. Analysis of the initial simulation run is described in the next chapter.

CHAPTER 5

CELL DESIGN SIMULATION EXPERIMENT PERFORMANCE ANALYSIS

5.1 OBJECTIVE

The objective of this section is to evaluate the performance of the system in question. Simulation models will be run initially under a set of specified conditions that will later be manipulated to assess the impact of varying cell parameters on product cycle times. The base model simulates the cell under a set of ideal conditions. Ideal conditions allow the cell product cycle times to meet the required production system demand cycle time.

5.2 BASE MODEL

To evaluate cell performance in terms of cycle times, a base model is developed that simulates the manufacture of a family of four parts. Part requests arrive randomly. Every time a different part request arrives, a setup has to be performed.

Since this system is a pull production system, satisfying requests mainly depend on the availability of parts in decoupler

5, and on the activities involved in replacing these parts. For example, if part A is pulled out of decoupler 5, operator moves back to decoupler 4 and unloads same part type. Operator proceeds by loading part into the vertical milling machine where it is processed. Once part is processed it goes through inspection and finally, assuming it is a defect-free part, it is deposited into decoupler 5 where it becomes available to be pulled. Thus, the availability of parts mostly depend on how efficient the previous activities are performed. Replacing parts in decoupler 4 depend also on similar activities that are performed from one process to the next. And depending on how each activity adds to the process could positively or adversely influence cycle times. If production smoothing is a problem in the cell, the ultimate result is prolonged waiting times caused by inefficient part flow from one process to the next.

Before each part is pulled from decoupler 5, it has to go through inspection, this is the final inspection that has to be performed to ensure the conformity of parts to quality standards. To assess the effect of reworks and scrap on cycle times, probabilities are manipulated between runs to allow for reworks. Traditionally, the final inspection should be merely a precautionary procedure to ensure the full conformance of parts to quality standards, since parts are continuously inspected and monitored throughout the cell. Implementing a preventive quality

inspection program in the cell will naturally reduce/eliminate reworks at the end operation, thus improving cycle times.

The first case analysis involves running the program under the above conditions, however, assuming there are no reworks present or changeovers required. Figure 9 shows the values obtained from running the simulation. Results show that the cycle time for part A is (4.87) minutes including inspection time, this time meets the required cycle time making a part available in decoupler 5 every 4.87 minutes. This part stays there until it is pulled by the next subassembly cell. If part is pulled out, only then can replacement activities take place. The cycle time for part B is shown to equal (8.00) minutes including inspection time. Cycle times are also shown for part C and D to equal (5.73) and (6.87) minutes, respectively, these times also meet the required 8-minute cycle time. Replacement times shown is the time it takes to replace parts pulled from decoupler 4, although these times meet the production system required time, they still exceed the theoretical (station) times. For example, theoretically, replacing part A should have taken about (4.5) minutes, this includes walking, unloading, and processing times. The difference can be allocated to shortages of SOH, waiting on the operator/inspection station, or as a result of the variation inherent in the process.

Other statistics are also shown such as utilization; operator utilization is shown to equal 98%, spindle utilization is highest among the machines at 78%, and an overall cell utilization of about

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/21/1994
Model revision date: 10/21/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	3.3025	1.2719	.06108	15.468	23
A INSPECTION TIME	1.5698	.14478	1.2086	2.1000	23
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	6.2992	.75288	.03477	11.417	8
B INSPECTION TIME	1.7112	.20284	1.1010	2.1245	8
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	4.1346	.71012	.35094	8.7173	9
C INSPECTION TIME	1.5956	.28191	.96795	2.4906	9
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	5.3181	.94271	.52336	14.099	7
D INSPECTION TIME	1.4119	.23789	.87216	1.8211	7
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	5.2048	1.0673	.49533	19.768	23
DECOUPLER4B REPL.TIME	3.7311	.94978	.45478	7.9332	7
DECOUPLER4C REPL.TIME	7.2683	.36949	4.0000	11.253	7
DECOUPLER4D REPL.TIME	5.8165	1.1091	.00000	17.160	6

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	78	Infinite
A PARTS REWORKED	0	Infinite
A PARTS SCRAPED	0	Infinite
B PARTS REWORKED	0	Infinite
B PARTS SCRAPED	0	Infinite
C PARTS REWORKED	0	Infinite
C PARTS SCRAPED	0	Infinite
D PARTS REWORKED	0	Infinite
D PARTS SCRAPED	0	Infinite

Figure 9. Base model output.

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR UTIL	98.245	.13365	.00000	100.00	100.00
SPINDLE UTIL	78.481	.52364	.00000	100.00	100.00
DRILL UTIL	55.144	.90191	.00000	100.00	100.00
LATHE UTIL	36.517	1.3185	.00000	100.00	100.00
HMILL UTIL	43.855	1.1315	.00000	100.00	.00000
VMILL UTIL	45.846	1.0868	.00000	100.00	.00000
CELL UTIL	51.969	.45625	.00000	100.00	60.000
NO.DELAYED AT M5	2.6534	1.1353	.00000	9.0000	6.0000
NO.DELAYED AT M4	.22184	2.0440	.00000	2.0000	.00000
NO.DELAYED AT M3	.02905	6.3916	.00000	2.0000	.00000
NO.DELAYED AT M2	.29734	2.2207	.00000	3.0000	.00000
NO.DELAYED AT M1	.24201	2.0009	.00000	3.0000	.00000
NO.IN D4A	1.4203	.55377	.00000	3.0000	3.0000
NO.IN D4B	.20828	1.9497	.00000	1.0000	.00000
NO.IN D4C	.25553	1.7069	.00000	1.0000	.00000
NO.IN D4D	.36740	1.3122	.00000	1.0000	.00000
NO.IN D3A	1.1040	.74154	.00000	3.0000	2.0000
NO.IN D3B	.26023	1.6860	.00000	1.0000	.00000
NO.IN D3C	.25276	1.7194	.00000	1.0000	.00000
NO.IN D3D	.29478	1.5467	.00000	1.0000	.00000
NO.IN D2A	.61394	1.3319	.00000	3.0000	.00000
NO.IN D2B	.15624	2.3239	.00000	1.0000	.00000
NO.IN D2C	.26731	1.6556	.00000	1.0000	.00000
NO.IN D2D	.24095	1.7749	.00000	1.0000	.00000
NO.IN D1A	.37668	1.4477	.00000	2.0000	.00000
NO.IN D1B	.11495	2.7747	.00000	1.0000	.00000
NO.IN D1C	.21009	1.9390	.00000	1.0000	.00000
NO.IN D1D	.13840	2.4950	.00000	1.0000	.00000

Run Time: 0 min(s) 2 sec(s)
Simulation run complete.

Figure 9. (continued)

52%. Cell utilization can be further increased by minimizing process variation and by increasing the number of operators in the cell. On the average, there were 2.6 requests blocked in front of machine 5, and the production rate is shown to have stopped at 78 units. Statistics in the figure show that there were no queues formed within the cell in front of the machines, this is attributable to the fact that the one piece part movement eliminates queues between machines or processes.

Overall results show a favorable cell performance in terms of cycle times which meet required demand time of 8 minutes, and since these product cycle times do not exceed the production system demand cycle time, these conditions under which the cell operates will form the basis for our base model. The cell performance, however, could be enhanced by further improvements in the cell process. Changes in cell parameters will be modeled in later simulations to assess their impact on cycle times. The next case will involve analyzing the impact of quality problems in the cell on product cycle times.

5.3 EFFECTS OF REWORKS ON CYCLE TIMES

To assess the impact of reworks on cycle times, the base simulation model was altered to allow for reworks. Increased machine use and increased demand, are some of the factors that could lead to quality deterioration in the cell, increased reworks

and scraps bring cycle times of a particular cell out of balance with the other cells in the production system. Figure 10 shows the results obtained with the presence of reworks and scraps. Results show an overall significant increase in product cycle times. This increase in cycle time is mainly attributed to the time spent waiting on reprocessing reworked parts. Due to the presence of reworks and scraps, cycle and replacement times have exceeded the 8-minute requirement time. For example, cycle time for part A is shown to be about (17.32) minutes, an increase from the base model of about 56%. This increase also accounts for (9.32) minutes increase over the production system required demand time. This increase in time is accounted for by the presence of reworks and scraps. For part B, cycle time is shown to be about (12.31) minutes, an approximate increase of 21% from the base model. This increase also exceeds required cycle time by (4.31) minutes. Similar observations can be made for parts C and D, as for part C there was a 37% increase in cycle time from the base model, again exceeding required cycle time by (4.33) minutes. As for part D, there was a 26% increase in cycle time with (3.66) minutes increase over required cycle time of 8 minutes. Presence of reworks has also influenced the production rate; number of requests processed has decreased by 11 units from the base model.

The above data has shown that increased quality problems in the cell, not only leads to a significant increase in product cycle

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/23/1994
Model revision date: 10/23/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	6.6296	1.0281	.37189	25.980	24
A INSPECTION TIME	2.0665	1.0910	1.2140	12.598	24
A REWORK TIME	5.6535	.57826	3.4829	14.582	10
A SCRAP TIME	2.9683	.30534	2.0309	3.8401	3
B CYCLE TIME	6.6709	.70097	1.1848	15.275	8
B INSPECTION TIME	1.6209	.19995	1.1010	2.1206	8
B REWORK TIME	4.0260	.00402	4.0146	4.0375	2
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	4.3750	.56193	1.2738	8.7173	7
C INSPECTION TIME	1.5554	.21977	.96795	2.0768	7
C REWORK TIME	6.3863	.00000	6.3863	6.3863	1
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	4.4917	.61056	.33147	8.3962	8
D INSPECTION TIME	1.4238	.33538	.87216	2.3678	8
D REWORK TIME	5.7518	.00000	5.7518	5.7518	1
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	9.7769	.98898	.44843	30.981	23
DECOUPLER4B REPL.TIME	2.8947	.93969	.45478	7.2528	7
DECOUPLER4C REPL.TIME	9.9827	.58581	4.4723	20.344	6
DECOUPLER4D REPL.TIME	6.5946	1.1838	.00000	20.025	7

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	67	Infinite
A PARTS REWORKED	10	Infinite
A PARTS SCRAPED	3	Infinite
B PARTS REWORKED	2	Infinite
B PARTS SCRAPED	0	Infinite
C PARTS REWORKED	1	Infinite
C PARTS SCRAPED	0	Infinite
D PARTS REWORKED	1	Infinite
D PARTS SCRAPED	0	Infinite

Figure 10. Base model with reworks.

times, but it will also decrease the production rate. From this we conclude that the importance of maintaining quality throughout the cell could significantly improve the cell performance. To achieve the desired goals of JIT and the benefits of lean production systems, cycle times have to meet demand times as imposed by other subassembly cells. By maintaining a quality program throughout the cell, only then, a higher performance standards and prevention/reduction of reworks at the final process can be achieved.

5.4 CHANGEOVER IMPACT ON CYCLE TIMES

The adoption of CMS implies the reduction or complete elimination of setups from one part to the next within a part family. In a pull system with the parts being pulled in a batch size of one, where changing the cell is needed from one part type to the next, it becomes rather imperative to reduce and eliminate all changeover activities where possible. So far, the base model has evaluated the performance of a pull-production cell assuming there was zero setups. The next case will evaluate the cell performance in terms of cycle times providing that changeover activities are required each time a different part is pulled out.

Figure 11 shows the result obtained under the current conditions. The figure shows the individual machine setup times as well as the total setup time which includes walking time

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/27/1994
Model revision date: 10/27/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
SPINDLE SETUP	2.5347	.13890	2.0313	2.9641	16
DRILL SETUP	1.4762	.20953	1.0828	1.9805	15
LATHE SETUP	4.5845	.06576	4.0805	4.9962	15
HMILL SETUP	1.2488	.13343	1.0112	1.4609	15
VMILL SETUP	.27735	.59158	.07841	.49313	15
TOTAL SETUP	10.395	1.0080	.69095	34.095	15
A CYCLE TIME	3.6199	.76727	.32555	7.7799	8
A INSPECTION TIME	1.5062	.08226	1.3104	1.7098	8
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	6.6240	.76621	.58563	16.031	6
B INSPECTION TIME	1.6538	.21445	1.3229	2.1077	6
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	3.9615	.50025	.40982	5.7710	7
C INSPECTION TIME	2.5529	.67787	1.0827	5.0574	7
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	10.243	.49967	4.1019	16.497	6
D INSPECTION TIME	1.5543	.24001	1.0576	2.1042	6
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	5.3257	.85134	.44070	13.756	8
DECOUPLER4B REPL.TIME	7.8170	.70740	1.5169	18.105	6
DECOUPLER4C REPL.TIME	9.7227	.56051	4.0000	20.106	7
DECOUPLER4D REPL.TIME	8.1364	.62486	5.2806	18.105	6

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	53	Infinite
NO.OF CYCLES	13	Infinite
NO.OF CHANGEOVERS	40	Infinite

Figure 11. Base model with changeovers.

between machines, and if any, waiting time. Setup time shown is about (10.4) minutes, this is the additional time it takes before the next part is available. There are two scenarios that could exist at this point depending on the policy and procedures under which the cell operates. The first case assumes that there are fully processed parts already available in the last decoupler, thus if a request for a particular part arrives it is instantly satisfied, this eliminates the need to wait on parts until after the cell is completely changed over. Replacing the part, however, will require that the cell has to be changed over before the next part can be processed, thus, assuming the next part request will arrive in 8 minutes, it will have to wait for (7.53) minutes before it is satisfied. The other case assumes that parts are not available in the last decoupler, in this case the cell has to be changed over before a part can be processed. For example, to satisfy demand it will take (10.4) minutes to change over the cell, as shown in Figure 11, plus (5.13) minutes worth of processing and inspection time. Obviously conditions under the first case are more favorable. However, for whichever case, it is obvious that changeover activities significantly increase cycle times. Changeover activities, however, can be improved depending on the policy and procedures practiced in the production system. For example, in this cell part requests arrive randomly in a uniform batch size of one, this requires that the cell has to be changed over frequently; this can be seen in the figure where it was necessary

to change over the cell 75% of the time. An alternative approach to reduce setups is by arranging consecutive arrivals of same part type, this reduces the frequency of changing the cell from one part type to the next. Under the given conditions of random arrivals, it becomes rather essential to reduce and eliminate changeover activities in this cell wherever possible.

5.5 SOH IMPACT ON CYCLE TIMES

In pull production cells, it is desirable to maintain SOH at a minimum level to minimize WIP inventories. Traditionally, maintaining high SOH was believed to increase utilization. The rationale was that the cost of idle time is higher that the cost of maintaining WIP inventories. SOH in pull production systems are the number of part units necessary to reduce process dependency in the cell, unlike WIP inventories, it can be increased or decreased accordingly to dampen some of the variations that could occur in the cell as a result of breakdowns, process variation, or demand fluctuations. Increasing SOH in pull production systems, however, may not always improve cycle times, thus if increasing SOH does not improve the cell performance, efforts can then be geared toward improving other aspects in the cell such as minimizing variation, or eliminating non-value added activities. To evaluate the impact of increasing SOH on cell performance, SOH was increased from one unit for

each part type to up to 5 units. A sample of the results obtained are shown in Figure 12 for SOH of 2 units. Results showed no improvements in cycle times were achieved. Increasing SOH to more than 2 units has also revealed similar results, these results were plotted in Figure 13. As shown in the figure, there were no improvements in cycle times, actually in some cases cycle times have exceeded the required cycle time.

Increased SOH in pull cells does not generally improve cycle times since regardless of the number of stock available, each unit pulled has to go through a specific set of activities that would not change with the manipulation of SOH, these results go in line with similar results found by Black [19]. For example, before the next part can be processed it is delayed until the previous part completes its inspection. Maintaining one unit of SOH in this cell, under the given conditions, seems to suit the cell's needs best. There seems to be relatively small improvement in cycle times at the 5 unit stock level, however, these improvements are off-set by the high number of stock maintained in the cell. For a part family of 4 parts, this means a total of 20 parts in each decoupler, which is an undesirable high number of WIP to maintain within the cell. Further inspection of the results reveals that the cell cycle time has actually increased from the base model value.

Increasing the number of SOH in the cell, not only increased the cell cycle time, but has also shown no improvement in the cell utilization. These results lead to the conclusion that pull

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/21/1994
Model revision date: 10/21/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	5.0264	.92689	.29102	17.761	18
A INSPECTION TIME	3.5561	1.2240	1.3055	13.120	17
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	7.1547	.42940	2.0554	12.595	12
B INSPECTION TIME	1.5992	.17634	1.1010	2.0587	12
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	7.3596	.92740	.70502	24.501	11
C INSPECTION TIME	2.3525	.72093	.96795	5.9576	11
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	4.1759	.60952	.74231	8.4806	11
D INSPECTION TIME	1.5470	.26604	.87216	2.1737	11
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	4.9747	1.1403	.48563	16.977	18
DECOUPLER4B REPL.TIME	5.9062	.96341	.45478	19.724	10
DECOUPLER4C REPL.TIME	10.163	.49907	4.0000	21.900	9
DECOUPLER4D REPL.TIME	5.2911	.87660	.00000	12.495	9

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	62	Infinite

Figure 12. SOH impact on cycle times.

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR UTIL	98.245	.13365	.00000	100.00	100.00
SPINDLE UTIL	71.323	.63409	.00000	100.00	100.00
DRILL UTIL	45.697	1.0901	.00000	100.00	100.00
LATHE UTIL	49.023	1.0197	.00000	100.00	100.00
HMILL UTIL	44.482	1.1172	.00000	100.00	.00000
VMILL UTIL	29.771	1.5359	.00000	100.00	.00000
CELL UTIL	48.059	.49371	.00000	100.00	60.000
NO.DELAYED AT M5	.10894	3.1292	.00000	2.0000	.00000
NO.DELAYED AT M4	.05030	4.5219	.00000	2.0000	.00000
NO.DELAYED AT M3	.23641	3.6826	.00000	4.0000	1.0000
NO.DELAYED AT M2	.48008	1.6901	.00000	3.0000	.00000
NO.DELAYED AT M1	.16970	2.8329	.00000	2.0000	.00000
NO.IN D4A	1.5424	.44663	.00000	2.0000	2.0000
NO.IN D4B	1.4267	.53991	.00000	2.0000	.00000
NO.IN D4C	1.1561	.72483	.00000	2.0000	.00000
NO.IN D4D	1.3237	.66713	.00000	2.0000	.00000
NO.IN D3A	1.4875	.87562	.00000	4.0000	3.0000
NO.IN D3B	.76855	1.1570	.00000	2.0000	.00000
NO.IN D3C	.82903	1.0070	.00000	2.0000	.00000
NO.IN D3D	1.1063	.80507	.00000	2.0000	.00000
NO.IN D2A	.82791	1.0851	.00000	3.0000	3.0000
NO.IN D2B	.67799	1.3313	.00000	2.0000	.00000
NO.IN D2C	.65114	1.2749	.00000	2.0000	.00000
NO.IN D2D	.88920	.97916	.00000	2.0000	.00000
NO.IN D1A	.56061	1.4498	.00000	3.0000	3.0000
NO.IN D1B	.91900	1.0533	.00000	2.0000	.00000
NO.IN D1C	.52333	1.3920	.00000	2.0000	.00000
NO.IN D1D	.60067	1.1900	.00000	2.0000	.00000

Run Time: 0 min(s) 2 sec(s)
Simulation run complete.

Figure 12. (continued)

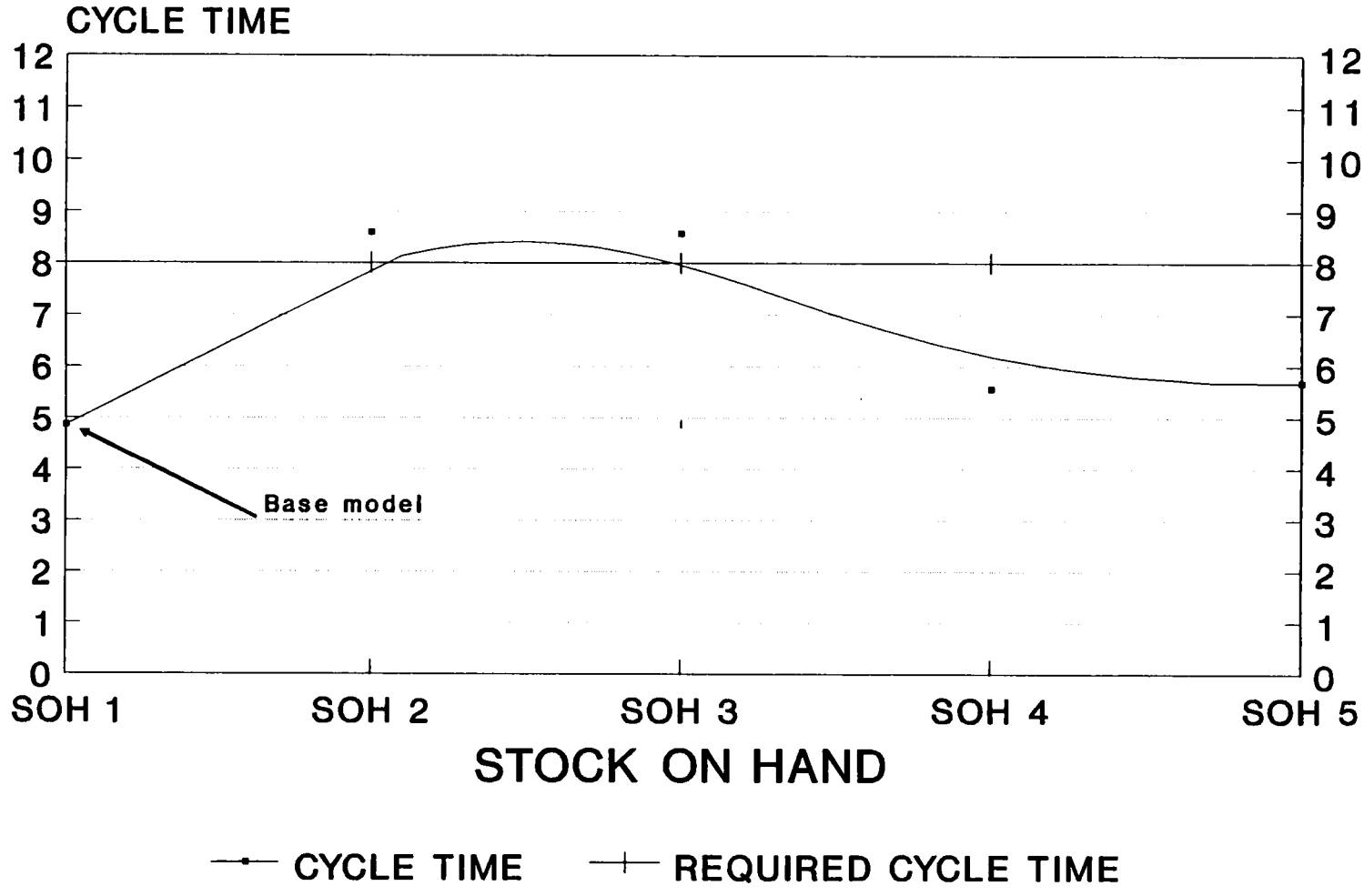


Figure 13. SOH impact on A cycle time.

production systems operate best under the minimum number of SOH.

5.6 PRODUCT/PROCESS VARIATION IMPACT ON CYCLE TIMES

Pull production cells are normally designed to minimize and expose variations as they occur within the cell. Variation within a cell can be either product or process related. Product variation is attributed to the variation in the size and shape of parts within the part family. Process variation results from the variation in machine processing times. In simulation, process variation is often simulated by the different distributions that imply the extent of variation present. For example, using normal distribution suggests that random variation could occur from one process to the next, and it could be small in one case and large in the next. Uniform distribution, meanwhile, implies that the production line is relatively standard and contains no special cases of prolonged processing times, variation is maintained within a range of a minimum and a maximum value. To better understand the effects of increasing/decreasing variation in the cell, two cases of product/process variation are evaluated.

The first case involves assessing the impact of increasing product variation on cycle times. Figure 14 shows a sample of results obtained with a 5% increase in product variation. As shown, cycle times have increased for all part types. With the

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/21/1994
Model revision date: 10/21/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	4.1363	1.0126	.32776	13.851	11
A INSPECTION TIME	1.5962	.14162	1.3350	1.9713	11
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	6.1905	.78295	.27969	15.453	18
B INSPECTION TIME	2.1641	1.1927	1.2882	12.453	18
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	7.0134	.82861	.23593	19.873	15
C INSPECTION TIME	2.0103	.61642	.94897	5.0659	15
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	6.1940	.74007	.17707	15.599	12
D INSPECTION TIME	1.8160	.23969	1.2374	2.6493	12
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	5.2682	.97685	.48441	15.066	11
DECOUPLER4B REPL.TIME	4.8061	1.1148	.47783	21.871	16
DECOUPLER4C REPL.TIME	7.7563	.31125	4.0000	11.068	14
DECOUPLER4D REPL.TIME	8.4055	1.5095	.00000	42.949	11

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	56	Infinite

Figure 14. Base model with 5% product variation.

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR UTIL	94.712	.23629	.00000	100.00	100.00
SPINDLE UTIL	79.119	.51373	.00000	100.00	100.00
DRILL UTIL	55.458	.89620	.00000	100.00	.00000
LATHE UTIL	55.331	.89849	.00000	100.00	.00000
HMILL UTIL	50.608	.98791	.00000	100.00	100.00
VMILL UTIL	42.382	1.1660	.00000	100.00	.00000
CELL UTIL	56.580	.45930	.00000	100.00	40.000
NO.DELAYED AT M5	.44659	1.8039	.00000	4.0000	.00000
NO.DELAYED AT M4	.26582	2.2530	.00000	3.0000	.00000
NO.DELAYED AT M3	.05328	4.9286	.00000	2.0000	.00000
NO.DELAYED AT M2	.15775	2.8089	.00000	3.0000	.00000
NO.DELAYED AT M1	.52475	1.8240	.00000	4.0000	.00000
NO.IN D4A	1.3849	.48234	.00000	2.0000	2.0000
NO.IN D4B	1.4757	.48219	.00000	2.0000	.00000
NO.IN D4C	.96369	.65378	.00000	2.0000	1.0000
NO.IN D4D	.76184	.65635	.00000	2.0000	1.0000
NO.IN D3A	1.3940	.48925	.00000	2.0000	.00000
NO.IN D3B	1.5948	.52342	.00000	3.0000	1.0000
NO.IN D3C	.95780	.72158	.00000	2.0000	.00000
NO.IN D3D	.58058	.84994	.00000	1.0000	1.0000
NO.IN D2A	1.2998	.55332	.00000	2.0000	2.0000
NO.IN D2B	.46402	1.0748	.00000	1.0000	.00000
NO.IN D2C	.81303	.79824	.00000	2.0000	1.0000
NO.IN D2D	.49187	1.0164	.00000	1.0000	.00000
NO.IN D1A	1.0866	.57241	.00000	2.0000	2.0000
NO.IN D1B	.45409	1.0965	.00000	1.0000	1.0000
NO.IN D1C	.44409	1.1188	.00000	1.0000	.00000
NO.IN D1D	.49823	1.0035	.00000	1.0000	.00000

Run Time: 0 min(s) 2 sec(s)
Simulation run complete.

Figure 14. (continued)

exception of part A, data show that part cycle times have slightly exceeded the required cycle time of 8 minutes. To further examine the impact of increasing product variation on cycle times, product variation was increased in intervals of 5% and up to 30%. Plotting results revealed that cycle times have increased when product variation was increased. A sample plot is shown for part A in Figure 15, the plot shows similar results to the other parts. Inspection of the plot reveals that significant changes in cycle times can be observed at the 10% variation and above. The cell seemed to absorb any product variation increase below 5%. Given the fact that product variation has already been accounted for in the base model, further increase in product variation beyond 5% may be a little extreme.

The next case involves evaluating the impact of increasing process variation on cycle times. This variation is attributed to the different processing times each machine spends on different parts. These variations often occur as machines become older, machine replacement, or as a result of introducing a new part family to the cell. The simulation model was revised to include the new machine processing times, values are tightly varied around the original value of 3.5 minutes with a standard deviation of .45 minutes. Results obtained from running the simulation are shown in Figure 16. Results show that cycle times for all parts, with the exception of part B, have increased. For parts C and D, cycle times have exceeded the required cycle time of 8 minutes.

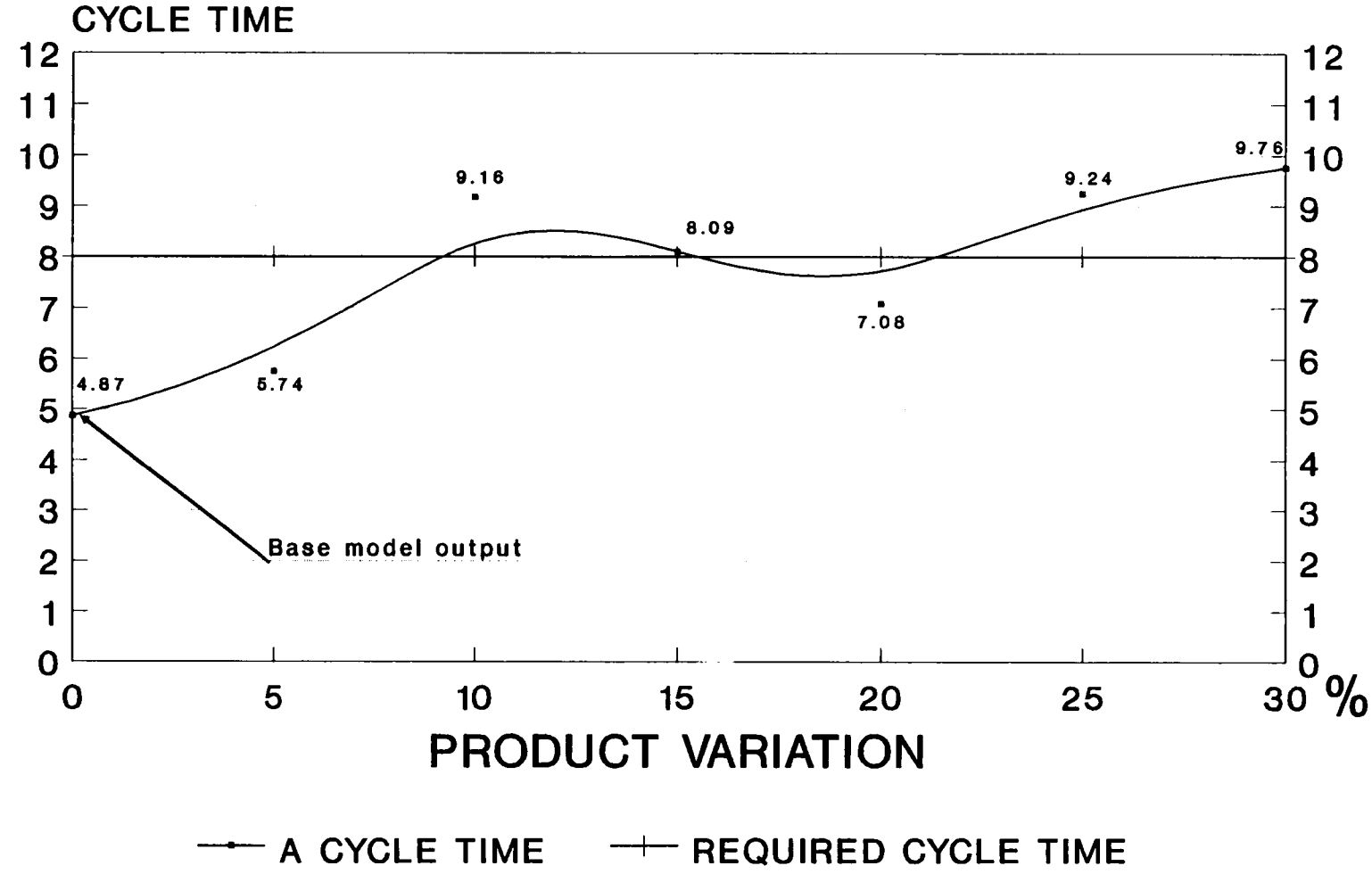


Figure 15. Product variation impact on A cycle time.

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/12/1994
Model revision date: 10/12/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	6.0739	.72325	1.6781	17.104	11
A INSPECTION TIME	1.5319	.18022	1.0708	1.9190	11
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	3.8872	.95642	.15702	10.415	8
B INSPECTION TIME	2.1259	.64653	1.3996	5.4622	8
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	9.2184	.75568	3.3456	24.952	10
C INSPECTION TIME	1.9774	.55856	.96795	4.9322	10
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	11.522	.83821	2.4182	32.230	7
D INSPECTION TIME	1.5340	.13577	1.2374	1.8996	7
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	11.023	1.2145	.45780	30.348	10
DECOUPLER4B REPL.TIME	7.8849	1.2193	.53622	27.341	7
DECOUPLER4C REPL.TIME	8.1509	.59261	4.0000	17.590	9
DECOUPLER4D REPL.TIME	15.529	1.0301	2.6665	35.261	5

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	77	Infinite

Figure 16. Process variation impact on cycle times.

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR UTIL	98.202	.13532	.00000	100.00	100.00
SPINDLE UTIL	85.519	.41150	.00000	100.00	.00000
DRILL UTIL	78.928	.51670	.00000	100.00	.00000
LATHE UTIL	57.434	.86089	.00000	100.00	100.00
HMILL UTIL	65.047	.73305	.00000	100.00	100.00
VMILL UTIL	46.632	1.0698	.00000	100.00	100.00
CELL UTIL	66.712	.34568	.00000	100.00	60.000
NO.DELAYED AT M5	8.0051	.72620	.00000	17.000	8.0000
NO.DELAYED AT M4	.06081	4.5450	.00000	3.0000	1.0000
NO.DELAYED AT M3	.08114	4.7335	.00000	2.0000	.00000
NO.DELAYED AT M2	.20736	2.1646	.00000	3.0000	.00000
NO.DELAYED AT M1	.14952	2.3850	.00000	1.0000	.00000
NO.IN D4A	.11454	2.7790	.00000	1.0000	.00000
NO.IN D4B	.37774	1.2835	.00000	1.0000	.00000
NO.IN D4C	1.6149	.44938	.00000	3.0000	1.0000
NO.IN D4D	.37357	1.2949	.00000	1.0000	.00000
NO.IN D3A	.10859	2.8651	.00000	1.0000	.00000
NO.IN D3B	.24326	1.7638	.00000	1.0000	.00000
NO.IN D3C	1.4491	.54537	.00000	3.0000	.00000
NO.IN D3D	.26132	1.6813	.00000	1.0000	.00000
NO.IN D2A	.16203	2.2742	.00000	1.0000	.00000
NO.IN D2B	.21489	1.9114	.00000	1.0000	.00000
NO.IN D2C	.94831	.67231	.00000	2.0000	.00000
NO.IN D2D	.26218	1.6775	.00000	1.0000	.00000
NO.IN D1A	.16164	2.2774	.00000	1.0000	1.0000
NO.IN D1B	.21747	1.8969	.00000	1.0000	.00000
NO.IN D1C	.36340	1.3236	.00000	1.0000	.00000
NO.IN D1D	.28011	1.6031	.00000	1.0000	.00000

Run Time: 0 min(s) 1 sec(s)
Simulation run complete.

Figure 16. (continued)

Close examination of the results obtained show that on the average there were 8 blocked requests, a significant problem in pull systems. To further assess the impact of process variation on cycle times, variation is minimized by uniformly distributing processing values around the same mean for each part type, results obtained and corresponding means are plotted in Figure 17. As shown in the figure, cycle times for parts C and D have decreased, however, there were no significant improvements in cell performance in terms of meeting required demand. But, blocked orders were reduced significantly from 8 to an average of 1 unit.

In conclusion, the results have proved that although the cell design flexibility allows for product variation, the cell capacity to accommodate increased product variation is, somewhat, limited. In our model cell, variation beyond 8% is not absorbed without exceeding the product cycle time. Increasing the product variation beyond the cell capacity has shown to adversely impact cycle times. Overall, process variation seemed to have the most impact on cycle times and it significantly increased the number of blocked orders in front of the cell. Although some improvements have been shown in cell performance after changing the process to uniform distribution, still there were no overall improvement in cycle times.

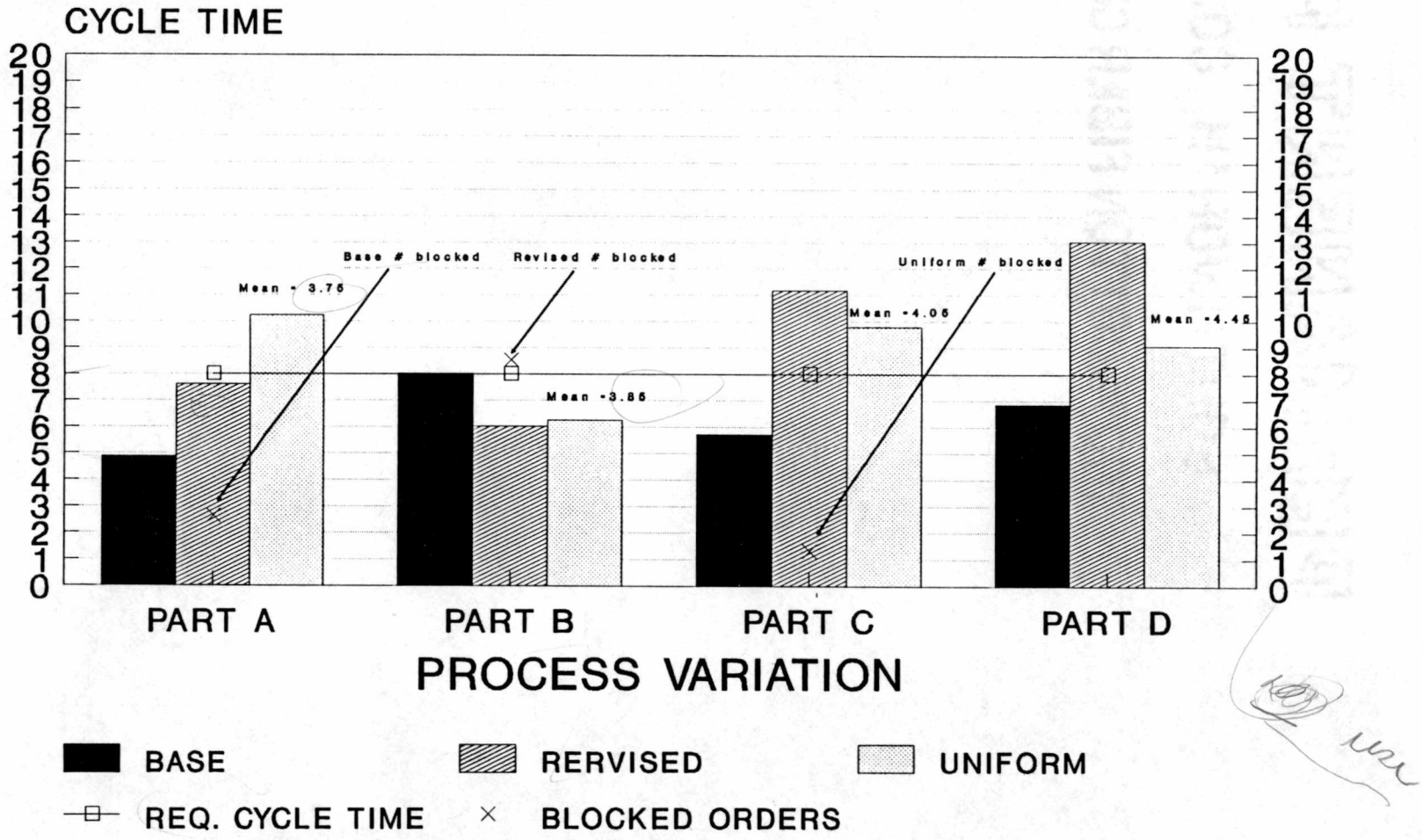


Figure 17. Reduced process variation impact on cycle times.

5.7 DEMAND INCREASE IMPACT ON CYCLE TIMES

Demand increase is a common component of production systems, demand increase is defined here as the increase in the number of request order arrivals per time period. In order to dampen the effect of increased demand increase, pull production cells counter this problem by maintaining SOH of one unit in each decoupler, increased machine reliability, and enhanced quality program are some of the factors that could be used to improve cell performance against demand fluctuation. To assess the impact caused by changed demand on cycle times, three cases were examined.

The first case examined the impact of increasing demand by 25%, since this is a pull system, demand increase is simply simulated by increasing the number of request arrivals to the cell. Increased arrival is represented by decreasing the between-arrival time of incoming order requests, thus increasing the demand by 25% results in decreasing the required cycle times from 8 minutes to 6 minutes. Figure 18 is a sample of the results that shows the change in cycle times. As shown in the figure part cycle times, with the exception of part B, have exceeded the required cycle time of 6 minutes. This increase could be accounted for as a result of operator's delay, process variation, or shortages of stock. It is also shown that the operator's utilization

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/21/1994
Model revision date: 10/21/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	5.4837	.78707	.35882	15.619	21
A INSPECTION TIME	1.6142	.16743	1.0963	2.0901	21
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	4.1974	.78046	.49336	8.9699	10
B INSPECTION TIME	1.6069	.16930	1.3139	2.2216	10
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	5.6956	.82122	.91983	16.325	10
C INSPECTION TIME	2.9190	1.2082	.96795	12.883	10
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	4.6507	1.2182	.06218	13.516	6
D INSPECTION TIME	1.6154	.17005	1.4639	2.1650	6
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	14.945	.66021	.42044	38.452	18
DECOUPLER4B REPL.TIME	6.5580	.95439	.37115	20.041	10
DECOUPLER4C REPL.TIME	16.368	.80215	4.0000	42.452	9
DECOUPLER4D REPL.TIME	5.3430	1.4433	.00000	18.537	5

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	74	Infinite

Figure 18. Demand increase output.

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR UTIL	99.883	.03421	.00000	100.00	100.00
SPINDLE UTIL	76.492	.55437	.00000	100.00	100.00
DRILL UTIL	50.728	.98554	.00000	100.00	.00000
LATHE UTIL	47.101	1.0598	.00000	100.00	100.00
HMILL UTIL	49.913	1.0017	.00000	100.00	100.00
VMILL UTIL	43.475	1.1402	.00000	100.00	100.00
CELL UTIL	53.542	.33141	.00000	100.00	80.000
NO.DELAYED AT M5	.83183	1.1733	.00000	4.0000	.00000
NO.DELAYED AT M4	.02521	6.2179	.00000	1.0000	.00000
NO.DELAYED AT M3	.02352	6.7342	.00000	2.0000	.00000
NO.DELAYED AT M2	.04414	4.6533	.00000	1.0000	.00000
NO.DELAYED AT M1	.30580	2.0861	.00000	3.0000	.00000
NO.IN D4A	.14234	2.4547	.00000	1.0000	.00000
NO.IN D4B	1.1340	.67670	.00000	2.0000	1.0000
NO.IN D4C	.17095	2.2022	.00000	1.0000	.00000
NO.IN D4D	.22417	1.8604	.00000	1.0000	.00000
NO.IN D3A	.14105	2.4677	.00000	1.0000	.00000
NO.IN D3B	.38495	1.2640	.00000	1.0000	.00000
NO.IN D3C	.14195	2.4586	.00000	1.0000	.00000
NO.IN D3D	.20449	1.9724	.00000	1.0000	.00000
NO.IN D2A	.18286	2.1139	.00000	1.0000	.00000
NO.IN D2B	.36751	1.3119	.00000	1.0000	.00000
NO.IN D2C	.12334	2.6661	.00000	1.0000	.00000
NO.IN D2D	.05181	4.2779	.00000	1.0000	.00000
NO.IN D1A	.11769	2.7380	.00000	1.0000	.00000
NO.IN D1B	.21278	1.9235	.00000	1.0000	.00000
NO.IN D1C	.06763	3.7130	.00000	1.0000	.00000
NO.IN D1D	.05400	4.1854	.00000	1.0000	.00000

Run Time: 0 min(s) 2 sec(s)
Simulation run complete.

Figure 18. (continued)

has increased from about 98% to about 100%, cell utilization has also increased slightly from 52% in the base model to 54%.

The next case examined the impact of increasing demand by 37%, in this case request arrival occurs every 5 minutes. Inspection of results obtained revealed similar conclusions, part cycle times have also exceeded required cycle time of 5 minutes. With the exception of increase in production rate, similar results in cycle times were also obtained after increasing demand by 50%. Increasing demand to 50%, however, has increased the production rate by about 15% only. At the same time, operator and cell utilization have slightly changed, these only slight changes are a results of constraints imposed by the cell activities. For example, before the next part can be processed, the previous part has to go through inspection. The number of operators in the cell also play a role in constraining the production rate. For example, before a part can be replaced the operator has to move, load, and unload the part for processing and inspection.

Results obtained from the previous three cases were plotted in Figure 19. In each case, part cycle times have exceeded associated required cycle times. Given the cell is under an improved set of conditions, the cell would have the capacity to cope with increased demand to some extent, factors such as increasing number of operators, improving process/product variation would likely improve the cell performance in terms of cycle times. However,

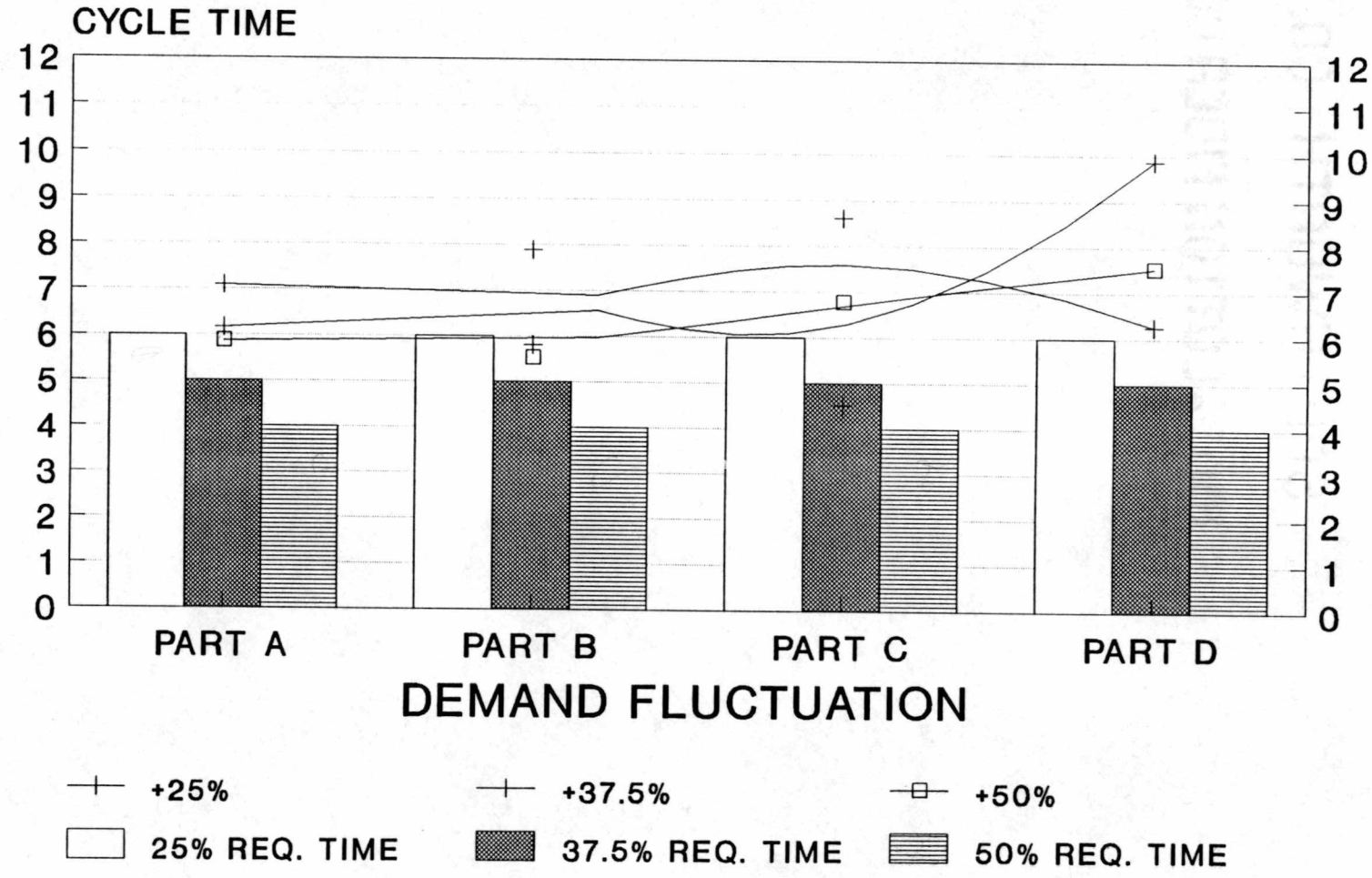


Figure 19. Demand fluctuation impact on cycle times.

under the current condition the cases tested showed that increased demand adversely impacted cycle times.

5.8 INCREASED PRODUCT MIX IMPACT ON CYCLE TIMES

To assess the impact of increasing product mix on cycle times, the product family was increased from 4-part types to 6-part types. A revised simulation model was developed from the initial model to accommodate the increase in product mix. New probabilities were assigned to include parts E and F, but still closely maintaining original product ratio in the new probability distribution.

Figure 20 depicts the results obtained from running the revised simulation model. Cycle times for parts B and D, as shown, have slightly exceeded the required cycle times. As for part B, the value exceeds the demand time by (.59) minutes, and as for part D, there was an increase of (1.64) minutes in cycle time. Although, cycle times for parts A and C have increased slightly from the base model, their cycle times remained below the required demand time of 8 minutes.

Close inspection of the above data leads to the fact that there is a slight variation between cycle times obtained and required demand time, one such possible source of variation could be process or product related. To further prove this point, processing times were reduced by 10%, and standard deviation was reduced

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/21/1994
Model revision date: 10/21/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	5.9487	.54652	.20911	12.652	12
A INSPECTION TIME	1.6598	.19748	1.2892	2.2517	12
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	7.0939	.40856	4.2847	12.076	7
B INSPECTION TIME	1.5022	.18207	1.0230	1.8106	7
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	4.7342	.71409	.96973	12.249	9
C INSPECTION TIME	2.3656	.79080	.90427	5.6372	9
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	7.9979	.83383	4.0735	15.698	3
D INSPECTION TIME	1.6385	.36620	1.1064	2.2888	3
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	7.8776	.58655	.37048	15.208	11
DECOUPLER4B REPL.TIME	7.0400	.62795	.87415	15.404	7
DECOUPLER4C REPL.TIME	10.072	.31660	4.5513	16.831	9
DECOUPLER4D REPL.TIME	6.6645	.06898	6.3198	7.1864	3

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	50	Infinite
A PARTS REWORKED	0	Infinite
A PARTS SCRAPED	0	Infinite
B PARTS REWORKED	0	Infinite
B PARTS SCRAPED	0	Infinite
C PARTS REWORKED	0	Infinite
C PARTS SCRAPED	0	Infinite
D PARTS REWORKED	0	Infinite
D PARTS SCRAPED	0	Infinite

Figure 20. Product mix output.

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
OPERATOR UTIL	83.702	.44127	.00000	100.00	100.00
SPINDLE UTIL	61.060	.79857	.00000	100.00	100.00
DRILL UTIL	53.292	.93619	.00000	100.00	100.00
LATHE UTIL	47.390	1.0536	.00000	100.00	100.00
HMILL UTIL	48.600	1.0284	.00000	100.00	100.00
VMILL UTIL	36.051	1.3319	.00000	100.00	.00000
CELL UTIL	49.279	.64225	.00000	100.00	80.000
NO.DELAYED AT M5	.37224	2.0408	.00000	3.0000	.00000
NO.DELAYED AT M4	.08018	3.7977	.00000	2.0000	.00000
NO.DELAYED AT M3	.01185	9.8695	.00000	2.0000	.00000
NO.DELAYED AT M2	.00859	10.742	.00000	1.0000	.00000
NO.DELAYED AT M1	.21450	2.7582	.00000	3.0000	.00000
NO.IN D4A	.76322	.55698	.00000	1.0000	.00000
NO.IN D4B	.86898	.38829	.00000	1.0000	1.0000
NO.IN D4C	.76688	.55135	.00000	1.0000	1.0000
NO.IN D4D	.95586	.21489	.00000	1.0000	1.0000
NO.IN D4E	.83962	.43706	.00000	1.0000	1.0000
NO.IN D4F	.02362	6.4287	.00000	1.0000	.00000
NO.IN D3A	.70313	.64978	.00000	1.0000	.00000
NO.IN D3B	.81532	.47593	.00000	1.0000	1.0000
NO.IN D3C	.79063	.51459	.00000	1.0000	1.0000
NO.IN D3D	.95853	.20799	.00000	1.0000	1.0000
NO.IN D3E	.83887	.43827	.00000	1.0000	1.0000
NO.IN D3F	1.0000	.00000	1.0000	1.0000	1.0000
NO.IN D2A	.59834	.81933	.00000	1.0000	.00000
NO.IN D2B	.66396	.71142	.00000	1.0000	1.0000
NO.IN D2C	.71110	.63739	.00000	1.0000	1.0000
NO.IN D2D	.94942	.23082	.00000	1.0000	1.0000
NO.IN D2E	.83131	.45047	.00000	1.0000	1.0000
NO.IN D2F	1.0000	.00000	1.0000	1.0000	1.0000
NO.IN D1A	.53127	.93931	.00000	1.0000	.00000
NO.IN D1B	.60601	.80632	.00000	1.0000	1.0000
NO.IN D1C	1.5515	.42683	.00000	2.0000	1.0000
NO.IN D1D	.61542	.79052	.00000	1.0000	1.0000
NO.IN D1E	.74749	.58122	.00000	1.0000	1.0000
NO.IN D1F	1.0000	.00000	1.0000	1.0000	1.0000

Run Time: 0 min(s) 2 sec(s)
Simulation run complete.

Figure 20. (continued)

by 15%, changing the original values to improve from (3.5,.45) to (3.1,.1) minutes. Results obtained from running the revised model are shown in Figure 21. As was expected, part cycle times have actually improved as a result of improving the process/product variation. Improvement was also observed in the production rate which has increased from 50 to 75 units (this is only 3 units less than the base model output). Figure 22 shows a plot of the above cases just examined. The results support the fact that increasing product mix has had an insignificant impact on cycle times, but slight improvements in the process seemed to further dampen this impact.

The final case examined the impact of increased product mix on changeovers. As shown in Figure 23, 93% of the time the cell had to be changed over, this account for an increase of 18% from the base model with a family of four parts.

As seen so far, changeover activities were identified as one means which causes an increase in cycle times. Concepts introduced in Chapter 3 are some of the tools which can be employed to reduce setup times. The ultimate objective is to eliminate setups between the different parts in the family. While this may not be totally feasible, the minimum objective should be to standardize tooling and to use an integrated fixture concept with rapid setting to allow for minimal setup times and activities.

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/21/1994
Model revision date: 10/21/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
A CYCLE TIME	4.8320	1.0289	.04123	14.256	9
A INSPECTION TIME	1.6455	.15161	1.3309	2.0390	9
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	5.2788	.79644	.69495	14.721	10
B INSPECTION TIME	1.5163	.15008	1.2798	2.0310	10
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	3.9294	.81428	.17712	12.193	13
C INSPECTION TIME	2.5211	.57652	1.3045	5.1273	13
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	4.2835	1.0842	.10923	12.901	7
D INSPECTION TIME	1.7826	.33792	.77594	2.6670	7
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	5.4971	.36866	.46931	7.6335	9
DECOUPLER4B REPL.TIME	6.6555	.60562	.44568	13.686	10
DECOUPLER4C REPL.TIME	9.0349	.34535	4.3184	15.028	13
DECOUPLER4D REPL.TIME	4.7361	1.1027	.00000	13.738	7

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	75	Infinite

Figure 21. Improved process/product mix.

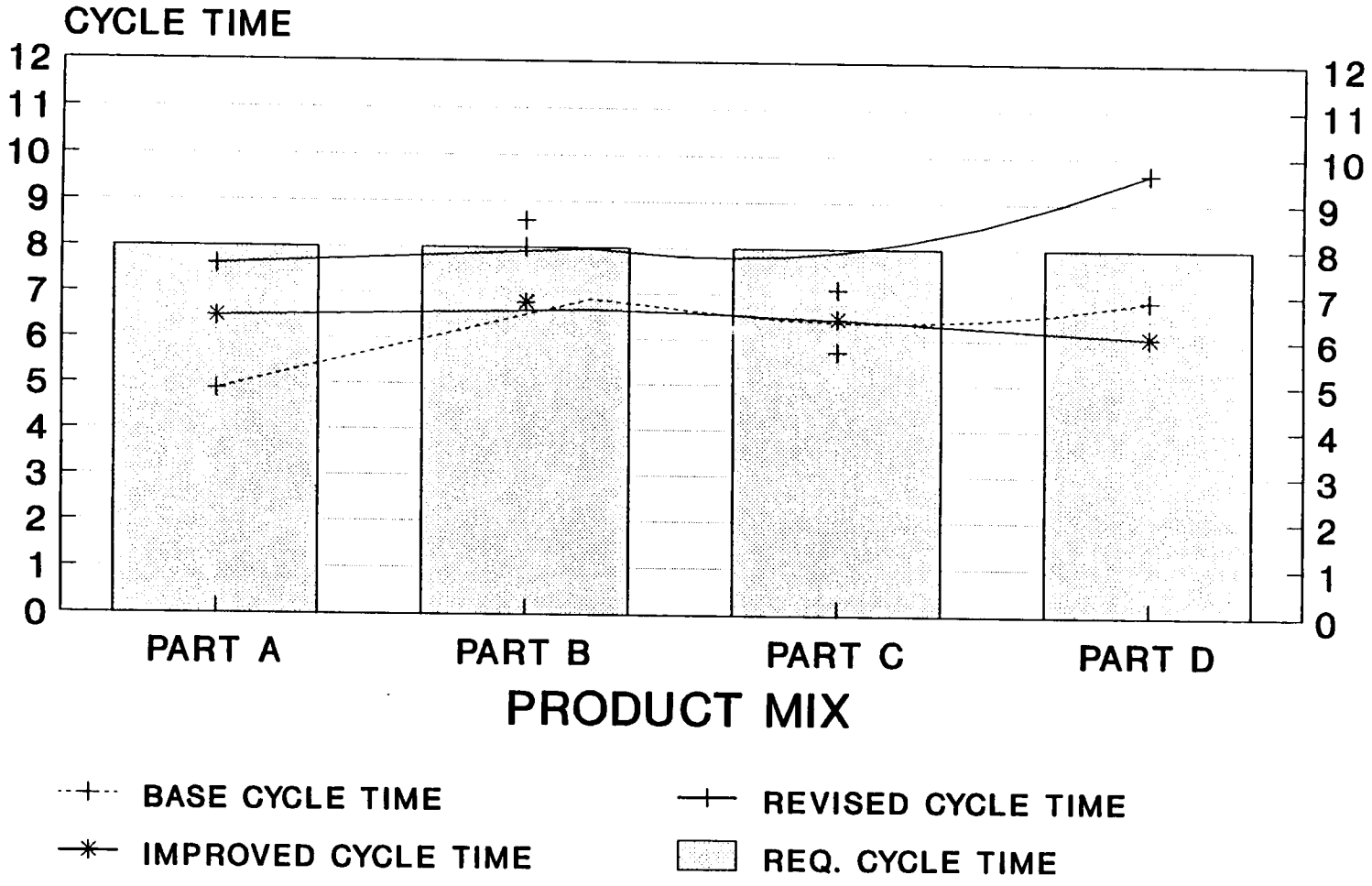


Figure 22. Product mix impact on cycle times.

Summary for Replication 1 of 1

Project: CELLULAR MANUFACTURING
Analyst: N ALTENEH

Run execution date : 10/27/1994
Model revision date: 10/27/1994

Replication ended at time : 480.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
SPINDLE SETUP	2.4211	.11782	2.0121	2.9146	21
DRILL SETUP	1.5508	.19207	1.0646	1.9947	21
LATHE SETUP	4.5490	.07440	4.0467	4.9913	21
HMILL SETUP	1.2778	.11585	1.0039	1.4977	21
VMILL SETUP	.21696	.66585	.02654	.49353	21
TOTAL SETUP	7.2475	.75927	.65921	20.925	21
A CYCLE TIME	5.7726	1.0954	.44748	15.519	6
A INSPECTION TIME	1.5433	.08758	1.3800	1.7766	6
A REWORK TIME	--	--	--	--	0
A SCRAP TIME	--	--	--	--	0
B CYCLE TIME	7.4353	.60264	4.2669	10.604	2
B INSPECTION TIME	1.4381	.02988	1.4077	1.4685	2
B REWORK TIME	--	--	--	--	0
B SCRAP TIME	--	--	--	--	0
C CYCLE TIME	3.7318	.89612	.49551	9.5712	6
C INSPECTION TIME	2.0124	.74686	1.3281	5.0783	6
C REWORK TIME	--	--	--	--	0
C SCRAP TIME	--	--	--	--	0
D CYCLE TIME	5.2885	.18097	4.6118	5.9653	2
D INSPECTION TIME	2.3270	.02611	2.2841	2.3700	2
D REWORK TIME	--	--	--	--	0
D SCRAP TIME	--	--	--	--	0
DECOUPLER4A REPL.TIME	8.5378	.91682	1.5092	20.929	6
DECOUPLER4B REPL.TIME	9.6006	.56618	5.7570	13.444	2
DECOUPLER4C REPL.TIME	9.3007	.53276	4.0000	18.188	6
DECOUPLER4D REPL.TIME	6.7288	.16801	5.9294	7.5282	2

COUNTERS

Identifier	Count	Limit
NO.OF REQUESTS	61	Infinite
NO.OF CYCLES	4	Infinite
NO.OF CHANGEOVERS	57	Infinite

Run Time: 0 min(s) 1 sec(s)
Simulation run complete.

Figure 23. Product mix impact on changeovers.

CHAPTER 6

RESULTS

The objective of this research was to identify and test some of the critical parameters involved in the design of a pull production cell in effort to understand the response of the cell's performance in terms of cycle times as they are influenced by the changes inherent in production systems. A cell was chosen as the base model for analysis utilizing simulation modeling. Several hypothesis were chosen to evaluate the impact of changing parameters on cell cycle times, and consequently on the product cycle time (the required demand cycle time).

- Quality problems in the cell, as was shown in Figure 10, have increased cycle times and decreased the production rate. The increase in the defect rate increases time spent waiting on reprocessing reworks and replacing scraps.

- Most of the impact occurred by changeover activities resulting from the actual time spent in changing over the cell. Changing over the cell, as was shown in Figure 11, has added a significant amount of time to product cycle times, it also caused a reduction in the production rate as a result of the excessive time

spent in changing over the cell from one part type to the next. Depending on the policy and procedures, however, the impact of changeover activities can be dampened significantly as was shown in section 5.4. One other aspect that can be altered to dampen the effects of changeover activities is to change the order rate arrival such that the parts' pull rate follow a predictable schedule "non-random arrival." In this way, changeover-activity cycles will be reduced significantly.

- Increasing the inventory in decouplers beyond an uniform size of one unit, as was shown in Figure 12, showed no improvement in cell cycle times or production output due to the constraint of delaying the processing of the next part until the previous part is fully processed and inspected.

- It was shown that the impact of product variation is fairly controllable below the 10% increase, see Figure 15. Increases in product variation often results from introducing a new product or product family. Although it is inevitable to avoid product variation, the cell seemed to be able to accommodate minimal changes if they should exist. Process variation on the other hand showed a larger effect on cycle times and especially increasing the number of blocked orders. Reduction of process variation showed some improvement, but not to the extent of the desired level.

Process variation, however, can be further minimized by continuous improvement and preventive maintenance.

- Increasing the demand rate on the cell by increasing the number of orders arriving per time period (this requires reducing the between arrival times) has shown to increase cycle times. One alternative way to increase the production rate in pull production cells is by increasing the number of operators. Increasing demand, however, does not often translate in increasing production rate unless the cell is actually equipped to accommodate this increase, otherwise, the result is increased cycle times. Increasing the number of operators in the cell, or increasing stock within the cell, are some of the temporary procedures that can cushion the impact of sudden demand increases. However, the success of dampening the effect of demand increase largely depends on the cell's line efficiency. For example, in the base model simulation output (Figure 9), it was shown that cycle times generated have exceeded the theoretical times referred to as station times (ST), this is the actual time, excluding any waiting times, of all work elements needed to produce a single unit. To be able to meet any changes in the production system demand times, the cell station time (ST) has to be below or equal to the production system product cycle time (CT). Line efficiency is one performance measure of the cell's

ability to meet the production system cycle time, mathematically, it can be stated as follows:

$$LE = \frac{\sum_{i=1}^{k} ST}{k(CT)}, \qquad k = \#\text{ of stations.} \qquad (6.1)$$

The objective is to maintain line efficiency equal to or below 100%. This implies that station times should not exceed the production system required cycle time. Utilizing equation 6.1 and using the 8-minute product cycle time, the following results were obtained:

$$LE = 23.5 / 5\ (8) = .588 = 59\%$$

By increasing demand by 25%, 37%, and 50%, the product cycle times were reduced from 8 minutes to 6, 5, and 4 minutes respectively. Using equations 6.1, again, the following results were obtained:

$$LE = 23.5 / 5(6) = .78 = 78\% \qquad (6.3)$$

$$LE = 23.5/ 5(5) = .94 = 94\% \qquad (6.4)$$

$$LE = 23.5/ 5(4) = 1.175 = 118\% \qquad (6.5)$$

This shows that the cell can actually accommodate a demand increase up to 37%, however, cell cycle times, as was shown in Figure 19, have exceeded the required product cycle times. This increase in cycle times can be attributed to waiting times resulting from shortages in stock, process variation, or slow operator. To be able to meet the new required times, waiting times have to be eliminated. Equation 6.5 shows that under the current conditions, the cell fails to accommodate the 50% increase in demand, the reason for this is that the cell station times exceeds the new required product cycle time. To be able to meet the new product cycle time of 4 minutes, some of the work elements (activities) have either to be reduced or eliminated.

- Increasing the product mix, as was shown in Figure 21, has mostly increased changeover cycles, but there was a minimal impact on product cycle times, slight reduction in the product and process variation have shown positive improvements. Overall, the cell seemed to be able to accommodate the increase in the product mix. Utilizing Pareto analysis, the average percent variation impact was charted in Figure 24. As shown in the figure, setups and quality problems had the most impact on cycle times, followed by process variation and demand fluctuation. Product mix, SOH, and product variation had the least impact on product cycle times.

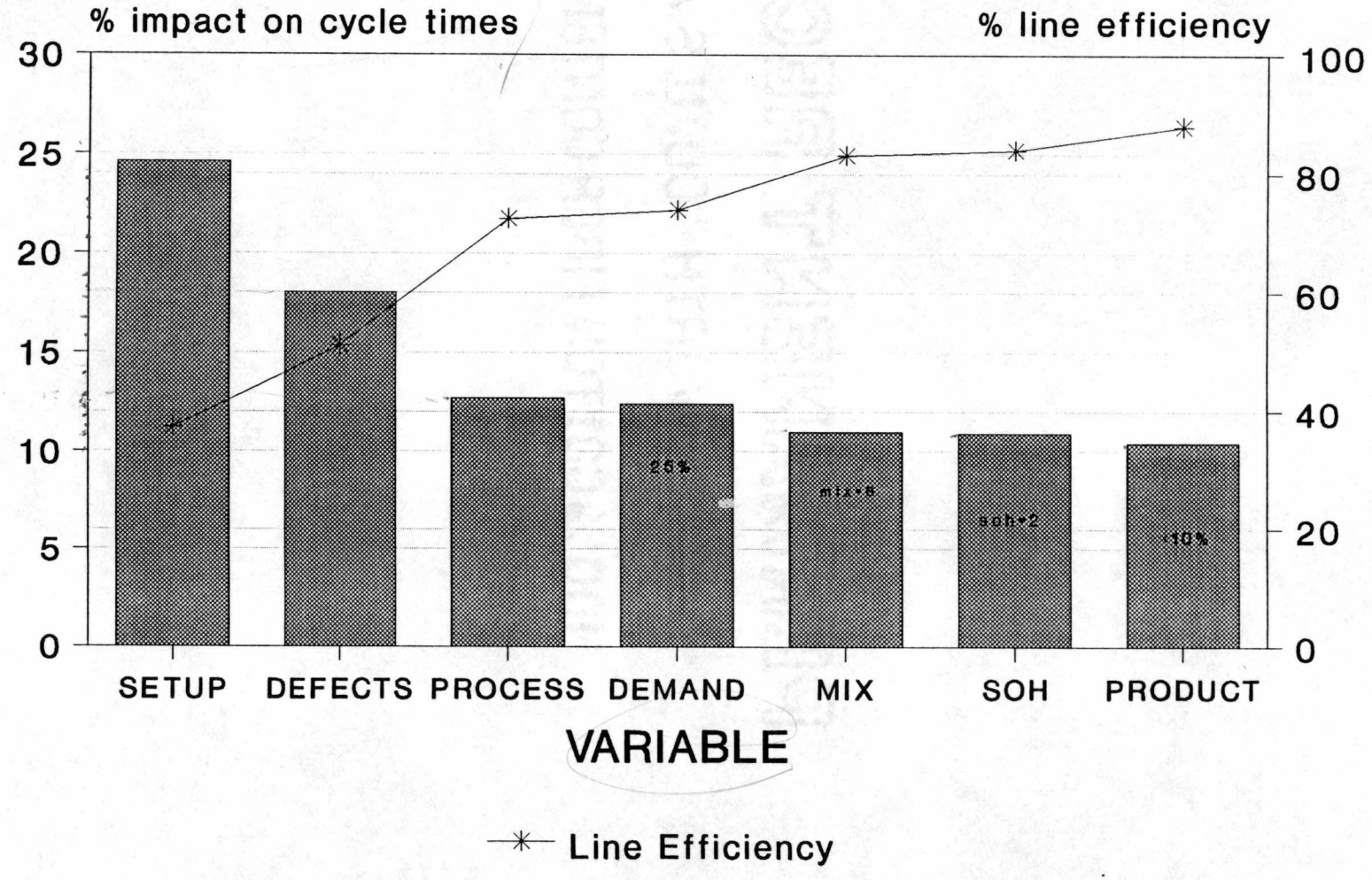


Figure 24. Pareto chart shows percent impact on cycle times.

The impact of the above variables were further analyzed to assess their effect on line efficiency. Analysis done on line efficiency, so far, was done to evaluate the cell performance as a whole against meeting changes in required cycle times as a result of demand fluctuation. In pull production cells, meeting the production system cycle time is mainly dependent on the last activity center (station 5) product cycle times. Changes in these product cycle times, as proven so far, largely influence the over all required production system cycle time, Figure 24 provided us with a visual representation of how much each variable has influenced product cycle times. It can also be shown that these variables have various levels of impact on the cell efficiency, the assumption is that cell efficiency decreases as variation increases and vise versa. Using equation 6.1, product cycle times generated in the simulation models, and station times, efficiency was calculated for each case above, results are shown below:

LE SETUP = 37.5%

LE DEFECTS = 51.1%

LE PROCESS = 72.4%

LE DEMAND = 73.9%

LE PRODUCT MIX = 83.2%

LE SOH = 84.0%

LE PRODUCT = 88.0%

The above values were also plotted in Figure 24, and it became evident that the cell efficiency has actually decreased as variation impact on cycle time increased and vise versa. This shows that the line efficiency is a function of variability, and it is drastically decreases as variability increases.

In today's competitive environment, pull production cells seem to offer the solutions to many of the problems that are facing the manufacturing industry. However, the benefits associated with these cells are rarely achieved to their fullest extent. Often the production system is subject to changes in policy, procedures, products and processes of which the extent of their impact is not fully understood. The objective of this research was to identify some of these issues and to assess their impact on one critical output and an associated benefit with these systems "product lead time."

LIST OF REFERENCES

[1] Bockerstette, Joseph A. and Shell, Richard L., Time-based Manufacturing, McGraw-Hill, Inc., NY, 1993.

[2] Snead, Charles S., Group Technology: Foundation For Competitive Manufacturing, Van Nostrand Reinhold, NY, 1989.

[3] Hernandez, Arnaldo, Just-In-Time Manufacturing: A Practical Approach, Prentice Hall, New Jersey, 1989.

[4] Brimson, James A., An Activity-Based Costing Approach, John Wiley and Sons, Inc., NY, 1991.

[5] Dennis, Pedgen C., Introduction To SIMAN With Version 3.0 Enhancements, Systems Modeling Corporation, PA, 1986.

[6] Shingo, Shiseo, Zero Quality Control: Source Inspection and The Poka-Yoke System, Productivity Press, Cambridge, 1986.

[7] Schroer, Bernard J., Black J. T. , and Zhang, Shou Xiang, "Just-In-Time (JIT), With Kanban Manufacturing System Simulation On A Microcomputer," Simulation, pp. 62-70, 1985.

[8] Mejabi, Olugbenga, and Wasserman, Gary S., "Simulation Constructs for JIT Modeling," International Journal Of Production Research, Vol. 30, No. 5, 1991, pp. 1119-1135.

[9] Sarker, Bhaba R., and Fitzsimmons, James A., "The Performance Of Push and Pull Systems: A Simulation And Comparative Study" International Journal Of Production Research, Vol. 27, No. 10, 1989, pp. 1715-1731.

[10] Leung, Ying T., and Sheen, Gwo-ji, "Resolving Deadlocks In Flexible Manufacturing Cells," Journal Of Manufacturing Systems, Vol. 12, No. 4, pp. 291-304.

[11] Vakharia, Asoo J., "Designing a Cellular Manufacturing System: A Material Flow Approach Based On Operation Sequence," IIE Transactions, Vol. 22, No. 1, 1990, pp. 84-95.

[12] Shingo, Shiegeo, A Revolution In Manufacturing: The SMED System, Productivity Press, Cambridge, 1985.

[13] Dennis Pedgen C., Shannon, Robert, and Sadowski, Randall, Introduction To Simulation Using SIMAN, McGraw-Hill, Inc., 1990.

[14] Crandall, Richard E., and Burwell, Timothy H., "The Effect Of Work-In-Process Inventory Levels On Throughput And Lead Times," Production And Inventory Management Journal, First Quarter, 1993, pp. 6-10.

[15] Haddock, Jorge, "A Simulation Generator For Flexible Manufacturing Systems Design And Control," IIE Transactions, Vol. 20, No. 1, 1988, pp. 22-29.

[16] Deane, Richard H., and Yang Jiagin, "Product Mix Selection And Closed Manufacturing Cell Flow Time Performance," International Journal Of Production Economics, Vol. 28, 1992, pp. 157-169.

[17] Sawhney, Rapinder S., "An Activity-Based Approach For Evaluating Strategic Investments In Manufacturing Companies," Journal Of Manufacturing Systems, Vol. 10, No. 5, pp. 353-367.

[18] Black, J. T., The Design Of The Factory With A Future, McGraw-Hill, Inc., N.Y., 1991.

[19] Black, J.T., and Schroer, Bernard, "Decouplers in Integrated Cellular Manufacturing Systems," Journal of Engineering for Industry, Vol. 110, 1988, pp. 77-85.

[20] Black, J. T., and Schroer, Bernard, "Simulation Of An Apparel Assembly Cell With Walking Workers And Decouplers," Journal Of Manufacturing Systems, Vol. 12, No. 2, 1993, pp. 170-179.

[21] Edwards, David A., Edsell, Robert, and Richa, Carlos, "Standard Operations- The Key To Continuous Improvement In A Just-In-Time Manufacturing System," Production And Inventory Management Journal, Third Quarter, 1993, pp. 7-13.

[22] Cheng, Chun Hung, "A Tree Search Algorithm For Designing A Cellular Manufacturing System," OMEGA Int. Journal of Management Science, Vol. 21, No. 4, 1993, pp. 489-496.

[23] Kerke, Sunder, "Performance Of A Manufacturing Cell With Increased Product Mix," IIE Transactions, Vol. 19, No. 3, 1987, pp. 329-339.

[24] Shafer, Scott M., and Charnes, John, "Cellular Versus Functional Layout Under A Variety Of Shop Operating Conditions," Decision Sciences, Vol. 24, No. 3, pp. 665-680.

[25] Berkley, Blair J., and Kiran Ali, "A Simulation Study of Scheduling Rules In A Kanban-Constrained Flow Shop," Decision Sciences, Vol. 22, 1991, pp. 559-582.

[26] Vakharia, Asoo J., Wemmerlov, Urban, "Designing a Cellular Manufacturing System: A Material Flow Approach Based On Operation Sequences," IIE Transactions, Vol. 22, No. 1, 1990, pp. 84-94.

[27] Tang, Ling-Lang, and Liu, Chang-Yung, "A Study On Decision Rules Of A Scheduling Model In An FMS," Computers In Industry, Vol. 22, 1993, pp. 1-13.

[28] Nevis, James L., Whitney, Daniel E., and DeFazio, Thomas L., Concurrent Design of Products and Processes: A Strategy For The Next Generation In Manufacturing, McGraw-Hill, Inc., N.Y., 1989.

[29] Bose, Gerald F., "Fast Cycle Manufacturing," Small Business Reports Magazine, 1990, pp. 14-20.

[30] Bowman, R. A., "Inventory: The Opportunity Cost Of Quality," IEE Transactions, Vol. 26, No. 3, 1994, pp. 40-47.

[31] Writh, Glen T., Mahmoodi, Farzad, and Mosier, Charles, "An Investigation Of Scheduling Policies In A Dual-Constrained Manufacturing Cell," Decision Sciences, Vol. 24, No. 4, pp. 761-786.

[32] Ketcham, Michael G., Shannon, Robert E., and Hogg, Gary L., "Information Structures For Simulation Modeling Of Manufacturing Systems," Simulation, February 1989, pp. 59-67.

[33] Jacobson, I. D., Townsend M. A., and Allatta, J. T., "Evaluation Of New Technologies In A Shoe Manufacturing Plant Using Simulation," Simulation, November 1993, pp. 303-312.

[34] Minnagunti, Kantha Rao, "Integrated Pull Manufacturing Cell," Thesis, Auburn University, 1992.

[35] Choi, Moo-Jin, "Manufacturing Cell Design," Production and Inventory Management Journal," Second Quarter, 1992, pp. 66-69.

[36] Seifoddini, Hamid, and Hsu, Chen-Pang, "Comparative Study of Similarity Coefficients And Clustering Algorithms In Cellular Manufacturing," Journal Of Manufacturing Systems, Vol. 13, No. 2, pp. 119-127;

[37] Dessouky, Yasser M., and Gerald T., "Scheduling To Minimize Makespan When A Changeover Exists, Journal of Manufacturing Systems, Vol. 9, No. 2, pp. 139-150.

APPENDICES

APPENDIX A

Base Simulation Model

```
BEGIN;

;                          SIMULATION BASE MODEL

;        CONTRIBUTION OF CELLULAR MANUFACTURING DESIGN TO

;              THE REDUCTION OF PRODUCTION LEAD TIME

;                    UTILIZING SIMULATION MODELING

;                              **********

                   CREATE: EX(ORDERIN);
                   COUNT:NO. OF REQUESTS;
ICYCLE             ASSIGN:PARTTYPE=
                   DISCRETE(.3,1,.55,2,.8,3,1.,4);
                   ASSIGN:X(6)=TNOW;
                   SIGNAL:1;              REQUEST FOR PARTS
                   QUEUE,ARRIVEBLOCK:DETACH;
                   CREATE;
BR1                BRANCH,1:
                            IF,NQ(ARRIVEBLOCK).GT.0,RMV1:
                            ELSE,WAIT1;
WAIT1              QUEUE,HOLDAT5;
                   WAIT:1,1:NEXT(BR1);
RMV1               REMOVE:1,ARRIVEBLOCK,BR2:NEXT(BR1);

;                  AC5 (VERTICAL MACHINE & DECOUPLER 4)

BR2                BRANCH,1:
                            IF,X(5).GT.0,BUSYVMILL:
                            ELSE,IDLEVMILL;
BUSYVMILL          QUEUE,DELAYATV;
                   WAIT:2,1;
IDLEVMILL          BRANCH,1:
                            IF,PARTTYPE==1,TYPE1:
                            IF,PARTTYPE==2,TYPE2:
                            IF,PARTTYPE==3,TYPE3:
                            ELSE,TYPE4;

TYPE1              BRANCH,1:
                            IF,MR(DECOUPLER4A).EQ.0,WAIT4A:
                            ELSE,GET4A;
WAIT4A             QUEUE,HOLD4A;
                   WAIT:3,1;
GET4A              QUEUE, UNLOAD4A;
                   SEIZE: OPERATOR;
                   DELAY: RN(WALKT);
                   RELEASE:  OPERATOR;
                   ALTER: DECOUPLER4A,-1;
```

```
                ALTER: DECOUPLER4A,-1;
                SIGNAL:  4; D4A SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5A;
PROCESS5A       ASSIGN:  X(5)=1; BUSY VMILL
                DELAY: RN(PT5);
                QUEUE, UNL1;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);UNLOAD PART FOR INSP.
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(7)=TNOW-X(6);
                TALLY:  A CYCLE TIME,X(7);
                BRANCH,1:
                        ALWAYS, INSPECT5A;
INSPECT5A       QUEUE, I5A:  MARK(INSP5TA);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:A INSPECTION TIME,INT(INSP5TA);
                BRANCH,1:
                        WITH, 1., GOOD5A:
                        ELSE,RW5A;

RW5A            COUNT:  A PARTS REWORKED:MARK(REWORKTA);


                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5A;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);
                RELEASE:  OPERATOR;
                TALLY:A REWORK TIME,INT(REWORKTA);
                ASSIGN:  X(5)=0;
                BRANCH,1:
                        WITH,1.,GOOD5A:
                        ELSE,SCRAP5A;
SCRAP5A         COUNT:  A PARTS SCRAPED;
                ASSIGN:X(8)=TNOW;
                DELAY:  UN(SCRAPT);
                ASSIGN:X(9)=TNOW-X(8);
                TALLY:A SCRAP TIME,X(9);
GOOD5A          ALTER: DECOUPLER5A, +1;
                SIGNAL: 2:  DISPOSE; VMILL AVAILABLE

;               PART TYPE 2

TYPE2           BRANCH,1:
                        IF, MR(DECOUPLER4B)   .EQ.0,
WAIT4B:
```

```
                        ELSE, GET4B;
WAIT4B          QUEUE, HOLD4B;
                WAIT: 5,1;
GET4B           QUEUE, UNLOAD4B;
                SEIZE:  OPERATOR;
                DELAY: RN(WALKT);
                RELEASE:  OPERATOR;

                ALTER: DECOUPLER4B, -1;
                SIGNAL: 4; D4B SIGNAL FOR REPLACEMENT


                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5B;
PROCESS5B       ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, UNL2;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(10)=TNOW-X(6);
                TALLY:  B CYCLE TIME,X(10);
                BRANCH,1:
                        ALWAYS, INSPECT5B;
INSPECT5B       QUEUE, I5B:  MARK(INSP5TB);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:B INSPECTION TIME,INT(INSP5TB);
                BRANCH,1:
                        WITH, 1., GOOD5B:
                        ELSE, RW5B;
RW5B            COUNT:  B PARTS REWORKED:MARK(REWORKTB);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5B;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(5)=0;
                TALLY:B REWORK TIME,INT(REWORKTB);
                BRANCH,1:
                        WITH,1.,GOOD5B:
                        ELSE,SCRAP5B;
SCRAP5B         COUNT:  B PARTS SCRAPED;
                ASSIGN:X(11)=TNOW;
                DELAY: UN(SCRAPT);
                ASSIGN:X(12)=TNOW-X(11);
                TALLY:B SCRAP TIME,X(12);
GOOD5B          ALTER:  DECOUPLER5B,+1;
```

```
                SIGNAL:  2:  DISPOSE; VMILL AVAILABLE

;               PART TYPE 3

TYPE3           BRANCH,1:
                        IF, MR(DECOUPLER4C) .EQ.0,
WAIT4C:
                        ELSE, GET4C;
WAIT4C          QUEUE, HOLD4C;
                WAIT: 6,1;
GET4C           QUEUE, UNLOAD4C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER4C, -1;
                SIGNAL:  4; D4C SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5C;
PROCESS5C       ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, UNL3;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(13)=TNOW-X(6);
                TALLY:  C CYCLE TIME,X(13);
                BRANCH,1:
                        ALWAYS, INSPECT5C;
INSPECT5C       QUEUE, I5C:  MARK(INSP5TC);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:C INSPECTION TIME,INT(INSP5TC);
                BRANCH,1:
                        WITH, 1., GOOD5C:
                        ELSE,RW5C;
RW5C            COUNT:  C PARTS REWORKED:MARK(REWORKTC);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5C;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(5)=0;
                TALLY:C REWORK TIME,INT(REWORKTC);
                BRANCH,1:
                        WITH,1.,GOOD5C:
                        ELSE,SCRAP5C;
SCRAP5C         COUNT:  C PARTS SCRAPED;
                ASSIGN:X(14)=TNOW;
```

```
                    DELAY:UN(SCRAPT);
                    ASSIGN:X(15)=TNOW-X(14);
                    TALLY:C SCRAP TIME,X(15);
GOOD5C              ALTER:  DECOUPLER5C, +1;
                    SIGNAL:  2:  DISPOSE; VMILL AVAILABLE

;                   PART TYPE 4

TYPE4               BRANCH,1:
                            IF, MR(DECOUPLER4D) .EQ.0,
WAIT4D:
                            ELSE, GET4D;
WAIT4D              QUEUE, HOLD4D;
                    WAIT:  7,1;
GET4D               QUEUE, UNLOAD4D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER4D, -1;
                    SIGNAL:  4; D4D SIGNAL FOR REPLACEMENT
                    BRANCH:
                            ALWAYS, HMILL:
                            ALWAYS, PROCESS5D;
PROCESS5D           ASSIGN:  X(5)=1;
                    DELAY:  RN(PT5);
                    QUEUE, UNL4;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(5)=0;
                    ASSIGN:X(16)=TNOW-X(6);
                    TALLY:  D CYCLE TIME,X(16);
                    BRANCH,1:
                            ALWAYS, INSPECT5D;
INSPECT5D           QUEUE, I5D:  MARK(INSP5TD);
                    SEIZE:  INSP5:  OPERATOR;
                    DELAY:  RN(INSPT);
                    RELEASE:  INSP5:  OPERATOR;
                    TALLY:D INSPECTION TIME,INT(INSP5TD);
                    BRANCH,1:
                            WITH, 1., GOOD5D:
                            ELSE, RW5D;

RW5D                COUNT:  D PARTS REWORKED:MARK(REWORKTD);
                    ASSIGN:  X(5)=1;
                    DELAY:  RN(PT5);
                    QUEUE, REWORK5D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(5)=0;
                    TALLY:D REWORK TIME,INT(REWORKTD);
```

```
                BRANCH,1:
                        WITH,1.,GOOD5D:
                        ELSE,SCRAP5D;
SCRAP5D         COUNT:  D PARTS SCRAPED;
                ASSIGN:X(17)=TNOW;
                DELAY:UN(SCRAPT);
                ASSIGN:X(18)=TNOW-X(17);
                TALLY:D SCRAP TIME,X(18);
GOOD5D          ALTER:  DECOUPLER5D, +1;
                SIGNAL:  2:  DISPOSE; VMILL AVAILABLE

;               AC4 (HORIZONTAL MILL & DECOUPLER 3)

HMILL           ASSIGN:X(19)=TNOW;
                QUEUE, AC4:  DETACH;
                CREATE;
BR3             BRANCH,1:
                       IF,NQ(AC4).GT.0,GETORDER:
                       ELSE,WAIT2;
WAIT2           QUEUE,HOLDBLOCKH;
                WAIT:4,1:NEXT(BR3);
GETORDER        REMOVE:1,AC4,BR4:NEXT(BR3);
BR4             BRANCH,1:
                        IF, X(4) .GT.0, BUSYHMILL:
                        ELSE, IDLEHMILL;
BUSYHMILL       QUEUE, DELAYATH;
                WAIT:  8,1;
IDLEHMILL       BRANCH,1:
                        IF, PARTTYPE==1, TYP1:
                        IF, PARTTYPE==2, TYP2:
                        IF, PARTTYPE==3, TYP3:
                        ELSE, TYP4;

TYP1            BRANCH,1:
                        IF, MR(DECOUPLER3A) .EQ.0,
WAITD3A:
                        ELSE, GETAFD3;
WAITD3A         QUEUE, HOLD3A;
                WAIT:  9,1;
GETAFD3         QUEUE, UNLOAD3A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER: DECOUPLER3A, -1;
                SIGNAL:  10; D3A SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4A;
PROCESS4A       ASSIGN:  X(4)=1; BUSY HMILL
                DELAY:  RN(PT4);
                QUEUE, UNL5;
                SEIZE:  OPERATOR;
```

```
                    DELAY:   RN(UNLOADT);
                    RELEASE:   OPERATOR;
                    ASSIGN:   X(4)=0;
                    SIGNAL:   8; HMILL AVAILABLE
                    ALTER:   DECOUPLER4A, +1;
                    ASSIGN:X(20)=TNOW-X(19);
                    TALLY:DECOUPLER4A REPL. TIME,X(20);
                    SIGNAL:   3:   DISPOSE; D4A REPL. SIGNAL

;                   PART TYPE 2

TYP2                BRANCH,1:
                            IF, MR(DECOUPLER3B) .EQ.0,
WAITD3B:
                            ELSE, GETBFD3;
WAITD3B             QUEUE, HOLD3B;
                    WAIT:   11,1;
GETBFD3             QUEUE, UNLOAD3B;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(WALKT);
                    RELEASE:   OPERATOR;
                    ALTER:   DECOUPLER3B, -1;
                    SIGNAL:   10; D3B SIGNAL FOR REPLACEMENT

                    BRANCH:
                            ALWAYS, LATHE:
                            ALWAYS, PROCESS4B;
PROCESS4B           ASSIGN:   X(4)=1;
                    DELAY:   RN(PT4);
                    QUEUE, UNL6;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:   X(4)=0;
                    SIGNAL:   8; HMILL AVAILABLE
                    ALTER:   DECOUPLER4B, +1;
                    ASSIGN:X(21)=TNOW-X(19);
                    TALLY:DECOUPLER4B REPL. TIME,X(21);
                    SIGNAL:   5:   DISPOSE; D4B REPL. SIGNAL

;                   PART TYPE 3

TYP3                BRANCH,1:
                            IF, MR(DECOUPLER3C) .EQ.0,
WAITD3C:
                            ELSE, GETCFD3;
WAITD3C             QUEUE, HOLD3C;
                    WAIT:   12,1;
GETCFD3             QUEUE, UNLOAD3C;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(WALKT);
                    RELEASE:   OPERATOR;
```

```
                ALTER:  DECOUPLER3C, -1;
                SIGNAL:  10; D3C SIGNAL FOR REPLACEMENT

                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4C;
PROCESS4C       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL7;
                SEIZE:  OPERATOR;
                DELAY:  UNLOADT;
                RELEASE: OPERATOR;
                ASSIGN:  X(4)=0;
                SIGNAL:  8; HMILL AVAILABLE
                ALTER:  DECOUPLER4C, +1;
                ASSIGN:X(22)=TNOW-X(19);
                TALLY:DECOUPLER4C REPL. TIME,X(22);
                SIGNAL:  6:  DISPOSE; D4C REPL. SIGNAL

;               PART TYPE 4

TYP4            BRANCH,1:
                        IF, MR(DECOUPLER3D) .EQ.0,
WAITD3D:
                        ELSE, GETDFD3;
WAITD3D         QUEUE, HOLD3D;
                WAIT:  13,1;
GETDFD3         QUEUE, UNLOAD3D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER3D, -1;
                SIGNAL:  10; D3D SIGNAL FOR REPLACEMENT

                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4D;
PROCESS4D       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL8;
                SEIZE:  OPERATOR;
                RELEASE:   OPERATOR;
                ASSIGN: X(4)=0;
                SIGNAL:8; HMILL AVAILABLE
                ALTER:  DECOUPLER4D, +1;
                ASSIGN:X(23)=TNOW-X(19);
                TALLY:DECOUPLER4D REPL. TIME,X(23);
                SIGNAL:  7:  DISPOSE; D4D REPL. SIGNAL

;               AC3 (LATHE & DECOUPLER 2)

LATHE           QUEUE, AC3:  DETACH;
```

```
                CREATE;
BR5             BRANCH,1:
                        IF, NQ(AC3) .GT.0, NEXTORDER:
                        ELSE, WAIT3;
WAIT3           QUEUE, HOLDBLOCKL;
                WAIT:  10:  NEXT(BR5);
NEXTORDER       REMOVE:  1, AC3, BR6:  NEXT (BR5);
BR6             BRANCH,1:
                        IF, X(3) .GT.0, BUSYLATHE:
                        ELSE, IDLELATHE;
BUSYLATHE       QUEUE, DELAYATL;
                WAIT:  14;
IDLELATHE       BRANCH,1:
                        IF, PARTTYPE==1, TY1:
                        IF, PARTTYPE==2, TY2:
                        IF, PARTTYPE==3, TY3:
                        ELSE, TY4;

TY1             BRANCH,1:
                        IF, MR(DECOUPLER2A) .EQ.0,
WAITD2A:
                        ELSE, GETAFD2;
WAITD2A         QUEUE, HOLD2A;
                WAIT:  15,1;
GETAFD2         QUEUE, UNLOAD2A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2A, -1;
                SIGNAL:  19; D2A SIGNAL FOR REPLACEMENT

                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3A;
PROCESS3A       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL9;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14; LATHE AVAILABLE
                ALTER:  DECOUPLER3A, +1;
                SIGNAL:  9:  DISPOSE; D3A REPL. SIGNAL

;               PART TYPE 2

TY2             BRANCH,1:
                        IF, MR(DECOUPLER2B) .EQ.0,
WAITD2B:
                        ELSE, GETBFD2;
WAITD2B         QUEUE, HOLD2B;
```

```
                WAIT:  16,1;
GETBFD2         QUEUE, UNLOAD2B;
                SEIZE:  OPERATOR;
                DELAY: RN( WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2B, -1;
                SIGNAL:  19; D2B SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3B;
PROCESS3B       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL10;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14; LATHE AVAILABLE
                ALTER:  DECOUPLER3B, +1;
                SIGNAL:  11:  DISPOSE; D3B REPL. SIGNAL

;               PART TYPE 3

TY3             BRANCH,1:
                        IF, MR(DECOUPLER2C) .EQ.0,
WAITD2C:
                        ELSE, GETCFD2;
WAITD2C         QUEUE, HOLD2C;
                WAIT:  17,1;
GETCFD2         QUEUE, UNLOAD2C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2C, -1;
                SIGNAL:  19; D2C SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3C;
PROCESS3C       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL11;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14; LATHE AVAILABLE
                ALTER:  DECOUPLER3C, +1;
                SIGNAL:  12:  DISPOSE; D3C REPL. SIGNAL

;               PART TYPE 4

TY4             BRANCH,1:
```

```
                        IF, MR(DECOUPLER2D) .EQ.0,
WAITD2D:
                        ELSE, GETDFD2;
WAITD2D         QUEUE, HOLD2D;
                WAIT:  18,1;
GETDFD2         QUEUE, UNLOAD2D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2D, -1;
                SIGNAL:  19; D2D SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3D;
PROCESS3D       ASSIGN:  X(3)=1;
                DELAY:RN(PT3);
                QUEUE, UNL12;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14; LATHE AVAILABLE
                ALTER:  DECOUPLER3D, +1;
                SIGNAL:  13:  DISPOSE; D3D REPL. SIGNAL

;               AC2 (DRILL & DECOUPLER 1)

DRILL           QUEUE, AC2:  DETACH;
                CREATE;
BR7             BRANCH,1:
                        IF, NQ(AC2) .GT.0, NEXTENTITY:
                        ELSE, WAIT4;
WAIT4           QUEUE, HOLDBLOCKD;
                WAIT:  19,1:  NEXT(BR7);
NEXTENTITY      REMOVE:  1,AC2, BR8:  NEXT(BR7);
BR8             BRANCH,1:
                        IF, X(2) .GT.0, BUSYDRILL:
                        ELSE, IDLEDRILL;
BUSYDRILL       QUEUE, DELAYATD;
                WAIT:  20,1;


IDLEDRILL       BRANCH,1:
                        IF, PARTTYPE==1, PART1:
                        IF, PARTTYPE==2, PART2:
                        IF, PARTTYPE==3, PART3:
                        ELSE, PART4;

PART1           BRANCH,1:
                        IF, MR(DECOUPLER1A) .EQ.0,
WAITD1A:
                        ELSE, GETAFD1;
```

```
WAITD1A         QUEUE, HOLD1A;
                WAIT:  21,1;
GETAFD1         QUEUE, UNLOAD1A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1A, -1;
                SIGNAL:  25; D1A SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2A;
PROCESS2A       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL13;
                SEIZE:  OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
                ALTER:  DECOUPLER2A, +1;
                SIGNAL:  15:  DISPOSE; D2A REPL. SIGNAL

;               PART TYPE 2

PART2           BRANCH,1:
                        IF, MR(DECOUPLER1B) .EQ.0,
WAITD1B:
                        ELSE, GETBFD1;
WAITD1B         QUEUE, HOLD1B;
                WAIT:  22,1;
GETBFD1         QUEUE, UNLOAD1B;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1B, -1;
                SIGNAL:  25; D1B SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2B;
PROCESS2B       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL14;
                SEIZE:  OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
                ALTER:  DECOUPLER2B, +1;
                SIGNAL:  16:  DISPOSE; D2A REPL. SIGNAL


;               PART TYPE 3
```

```
PART3           BRANCH,1:
                        IF, MR(DECOUPLER1C) .EQ.0,
WAITD1C:
                        ELSE, GETCFD1;
WAITD1C         QUEUE, HOLD1C;
                WAIT:  23,1;
GETCFD1         QUEUE, UNLOAD1C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1C, -1;
                SIGNAL:  25; D1C SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2C;
PROCESS2C       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL15;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
                ALTER:  DECOUPLER2C, +1;
                SIGNAL:  17:  DISPOSE; D2A REPL. SIGNAL


;               PART TYPE 4

PART4           BRANCH,1:
                        IF, MR(DECOUPLER1D) .EQ.0,
WAITD1D:
                        ELSE, GETDFD1;
WAITD1D         QUEUE, HOLD1D;
                WAIT:  24,1;
GETDFD1         QUEUE, UNLOAD1D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1D, -1;
                SIGNAL:  25; D1D SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2D;
PROCESS2D       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL16;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
```

```
                    ALTER:  DECOUPLER2D, +1;
                    SIGNAL:  18:  DISPOSE; D2A REPL. SIGNAL

;                   AC1 (SPINDLE MACHINE)

SPINDLE             QUEUE, AC1:  DETACH;
                    CREATE;
BR9                 BRANCH,1:
                            IF, NQ(AC1) .GT.0, M1:
                            ELSE, WAIT5;
WAIT5               QUEUE, HOLDBLOCKS;
                    WAIT:  25,1:  NEXT (BR9);
M1                  REMOVE:  1, AC1, BR10:  NEXT (BR9);
BR10                BRANCH,1:
                            IF, X(1) .GT.0, BUSYSPINDLE:
                            ELSE, IDLESPINDLE;
BUSYSPINDLE         QUEUE, DELAYATS;
                    WAIT:  26,1;
IDLESPINDLE         BRANCH,1:
                            IF, PARTTYPE==1, PT1:
                            IF, PARTTYPE==2, PT2:
                            IF, PARTTYPE==3, PT3:
                            ELSE, PT4;

PT1                 QUEUE, CARTA;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
                    DELAY:  LOADT;LOAD/POSITION/START MACHINE
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLEA;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(1)=0;
                    SIGNAL:  26; SPINDLE AVAILABLE
                    ALTER:  DECOUPLER1A, +1;
                    SIGNAL:  21:DISPOSE; D1A REPL. SIGNAL

PT2                 QUEUE, CARTB;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
                    DELAY: RN(LOADT);
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLEB;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(1)=0;
```

```
                    SIGNAL:   26; SPINDLE AVAILABLE
                    ALTER:   DECOUPLER1B, +1;
                    SIGNAL:   22:DISPOSE; D1B REPL. SIGNAL

PT3                 QUEUE, CARTC;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(PAPART);
                    ASSIGN:   X(1)=1;
                    DELAY:   RN(LOADT);
                    RELEASE:   OPERATOR;
                    DELAY:   RN(PT1);
                    QUEUE, ECYCLEC;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(UNLOADT);
                    RELEASE:   OPERATOR;
                    ASSIGN:   X(1)=0;
                    SIGNAL:   26; SPINDLE AVAILABLE
                    ALTER:   DECOUPLER1C, +1;
                    SIGNAL:   23:DISPOSE; D1C REPL. SIGNAL

PT4                 QUEUE, CARTD;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(PAPART);
                    ASSIGN:   X(1)=1;
                    DELAY:   RN(LOADT);
                    RELEASE:   OPERATOR;
                    DELAY:   RN(PT1);
                    QUEUE, ECYCLED;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(UNLOADT);
                    RELEASE:   OPERATOR;
                    ASSIGN:   X(1)=0;
                    SIGNAL:   26; SPINDLE AVAILABLE
                    ALTER:   DECOUPLER1D, +1;
                    SIGNAL:   24:DISPOSE; D1D REPL. SIGNAL
 END;
```

```
BEGIN;

PROJECT, CELLULAR MANUFACTURING, N ALTENEH;

PARAMETERS:     1,ORDERIN,8     :! TIME BETWEEN ORDERS
                2,WALKT,.5,.1   :! WALKING TIME
                3,INSPT,1,.1    :! INSPECTION TIME
                4,UNLOADT,.5,.1:! UNLOADING MACHINES
                5,PT5,3.5,.45   :! PROCESSING TIMES
                6,PT4,3.5,.45   :
                7,PT3,3.5,.45   :
                8,PT2,3.5,.45   :
                9,PT1,3.5,.45   :
                10,PAPART,.5,.1:! PICK A PART
                11,LOADT,.8,.1  :! LOADING TIME
                12,SCRAPT,2,4   ;! SCRAP TIME

ATTRIBUTES:     INSP5TA :          INSP5TB  :
                INSP5TC :          INSP5TD  :
                REWORKTA:          REWORKTB :
                REWORKTC:          REWORKTD :
                PARTTYPE                    ;

QUEUES:         UNLOAD4A:          UNLOAD4B:
                UNLOAD4C:          UNLOAD4D:
                UNLOAD3A:          UNLOAD3B:
                UNLOAD3C:          UNLOAD3D:
                UNLOAD2A:          UNLOAD2B:
                UNLOAD2C:          UNLOAD2D:
                UNLOAD1A:          UNLOAD1B:
                UNLOAD1C:          UNLOAD1D:
                REWORK5A:          REWORK5B:
                REWORK5C:          REWORK5D:
                DELAYATV:          DELAYATH:
                DELAYATL:          DELAYATD:
                DELAYATS:          HOLDAT5 :
                HOLD4A  :          HOLD4B  :
                HOLD4C  :          HOLD4D  :
                HOLD2A  :          HOLD1B  :
                HOLD3A  :          HOLD3B  :
                HOLD3C  :          HOLD3D  :
                HOLD2B  :          HOLD2C  :
                HOLD2D  :          HOLD1A  :
                HOLD1C  :          HOLD1D  :
                ECYCLEA :          ECYCLEB :
                ECYCLEC :          ECYCLED :
                CARTA   :          CARTB   :
                CARTC   :          CARTD   :
                I5A: I5B:  I5C:    I5D     :
                AC4: AC3:  AC2:    AC1     :
                UNL1:  UNL2:  UNL3:  UNL4:
                UNL5:  UNL6:  UNL7:  UNL8:
```

```
                UNL9: UNL10: UNL11: UNL12:
                UNL13: UNL14: UNL15:UNL16:
                ARRIVEBLOCK:    HOLDBLOCKS:
                HOLDBLOCKD:     HOLDBLOCKL:
                HOLDBLOCKH                ;

RESOURCES:      DECOUPLER1A,SCHED(1) :
                DECOUPLER1B,SCHED(2) :
                DECOUPLER1C,SCHED(3) :
                DECOUPLER1D,SCHED(4) :
                DECOUPLER2A,SCHED(5) :
                DECOUPLER2B,SCHED(6) :
                DECOUPLER2C,SCHED(7) :
                DECOUPLER2D,SCHED(8) :
                DECOUPLER3A,SCHED(9) :
                DECOUPLER3B,SCHED(10):
                DECOUPLER3C,SCHED(11):
                DECOUPLER3D,SCHED(12):
                DECOUPLER4A,SCHED(13):
                DECOUPLER4B,SCHED(14):
                DECOUPLER4C,SCHED(15):
                DECOUPLER4D,SCHED(16):
                DECOUPLER5A,SCHED(17):
                DECOUPLER5B,SCHED(18):
                DECOUPLER5C,SCHED(19):
                DECOUPLER5D,SCHED(20):
                OPERATOR:        INSP5;

SCHEDULES:      1,1 :   2,1 :   3,1 :   4,1 :  5,1 :
                6,1 :   7,1 :   8,1 :   9,1 :  10,1:
                11,1:  12,1 :  13,1 :  14,1 :  15,1:
                16,1:  17,1 :  18,1 :  19,1 :  20,1;

DSTAT:          NR(OPERATOR) * 100, OPERATOR UTIL:
                X(1)  * 100, SPINDLE UTIL:
                X(2)  * 100, DRILL UTIL  :
                X(3)  * 100, LATHE UTIL  :
                X(4) * 100, HMILL UTIL   :
                X(5) * 100, VMILL UTIL   :
                (X(1) + X(2) + X(3)      +
                X(4) + X(5))* 100/5, CELL UTIL :
                NQ(DELAYATV), NO. DELAYED AT M5:
                NQ(DELAYATH), NO. DELAYED AT M4:
                NQ(DELAYATL), NO. DELAYED AT M3:
                NQ(DELAYATD), NO. DELAYED AT M2:
                NQ(DELAYATS), NO. DELAYED AT M1:
                MR(DECOUPLER4A), NO. IN D4A    :
                MR(DECOUPLER4B), NO. IN D4B    :
                MR(DECOUPLER4C), NO. IN D4C    :
                MR(DECOUPLER4D), NO. IN D4D    :
                MR(DECOUPLER3A), NO. IN D3A    :
                MR(DECOUPLER3B), NO. IN D3B    :
```

```
                    MR(DECOUPLER3C), NO. IN D3C    :
                    MR(DECOUPLER3D), NO. IN D3D    :
                    MR(DECOUPLER2A), NO. IN D2A    :
                    MR(DECOUPLER2B), NO. IN D2B    :
                    MR(DECOUPLER2C), NO. IN D2C    :
                    MR(DECOUPLER2D), NO. IN D2D    :
                    MR(DECOUPLER1A), NO. IN D1A    :
                    MR(DECOUPLER1B), NO. IN D1B    :
                    MR(DECOUPLER1C), NO. IN D1C    :
                    MR(DECOUPLER1D), NO. IN D1D    ;

COUNTERS:           NO. OF REQUESTS,,     NO:
                    A PARTS REWORKED,,    NO:
                    A PARTS SCRAPED,,     NO:
                    B PARTS REWORKED,,    NO:
                    B PARTS SCRAPED,,     NO:
                    C PARTS REWORKED,,    NO:
                    C PARTS SCRAPED,,     NO:
                    D PARTS REWORKED,,    NO:
                    D PARTS SCRAPED,,     NO;

TALLIES:            A CYCLE TIME :
                    A INSPECTION TIME:
                    A REWORK TIME:
                    A SCRAP TIME :
                    B CYCLE TIME :
                    B INSPECTION TIME:
                    B REWORK TIME:
                    B SCRAP TIME :
                    C CYCLE TIME :
                    C INSPECTION TIME:
                    C REWORK TIME:
                    C SCRAP TIME :
                    D CYCLE TIME :
                    D INSPECTION TIME:
                    D REWORK TIME:
                    D SCRAP TIME :
                    DECOUPLER4A REPL. TIME:
                    DECOUPLER4B REPL. TIME:
                    DECOUPLER4C REPL. TIME:
                    DECOUPLER4D REPL. TIME;

REPLICATE,,0,480;

END;
```

APPENDIX B

Simulation Model With Changeovers

```
BEGIN;

;              SIMULATION BASE MODEL WITH CHANGEOVERS

;       CONTRIBUTION OF CELLULAR MANUFACTURING DESIGN TO

;              THE REDUCTION OF PRODUCTION LEAD TIME

;                   UTILIZING SIMULATION MODELING

;                              **********

                  CREATE: EX(ORDERIN);
                  COUNT:NO. OF REQUESTS;
                  ASSIGN:ASETUP=1;
                  ASSIGN:BSETUP=2;
                  ASSIGN:CSETUP=3;
                  ASSIGN:DSETUP=4;
                  ASSIGN:ISETUP=DISCRETE
                  (.25,ASETUP,.5,BSETUP,.75,
                  CSETUP,1.,DSETUP);
                  ASSIGN:PARTTYPE=
                  DISCRETE(.3,1,.55,2,.8,3,1.,4);
                  BRANCH,1:
                           IF,PARTTYPE.NE.ISETUP,CHANGEOVER:
                           ELSE,ICYCLE;
CHANGEOVER        COUNT:NO. OF
                  CHANGEOVERS:NEXT(BEGINSETUP);
ICYCLE            COUNT:NO. OF CYCLES;
BEGINCYCLE        ASSIGN:X(6)=TNOW;
                  SIGNAL:1;              REQUEST FOR PARTS
                  QUEUE,ARRIVEBLOCK:DETACH;

                  CREATE;
BR1               BRANCH,1:
                           IF,NQ(ARRIVEBLOCK).GT.0,RMV1:
                           ELSE,WAIT1;
WAIT1             QUEUE,HOLDAT5;
                  WAIT:1,1:NEXT(BR1);
RMV1              REMOVE:1,ARRIVEBLOCK,BR2:NEXT(BR1);

;                 AC5 (VERTICAL MACHINE & DECOUPLER 4)

BR2               BRANCH,1:
                           IF,X(5).GT.0,BUSYVMILL:
                           ELSE,IDLEVMILL;
BUSYVMILL         QUEUE,DELAYATV;
                  WAIT:2,1;
IDLEVMILL         BRANCH,1:
                           IF,PARTTYPE==1,TYPE1:
                           IF,PARTTYPE==2,TYPE2:
                           IF,PARTTYPE==3,TYPE3:
```

```
                        ELSE,TYPE4;

TYPE1           BRANCH,1:
                        IF,MR(DECOUPLER4A).EQ.0,WAIT4A:
                        ELSE,GET4A;
WAIT4A          QUEUE,HOLD4A;
                WAIT:3,1;
GET4A           QUEUE, UNLOAD4A;
                SEIZE: OPERATOR;
                DELAY: RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER: DECOUPLER4A,-1;
                SIGNAL:  4; D4A SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5A;
PROCESS5A       ASSIGN:  X(5)=1; BUSY VMILL
                DELAY: RN(PT5);
                QUEUE, UNL1;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);  UNLOAD PART FOR
INSPECTION
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(7)=TNOW-X(6);
                TALLY:  A CYCLE TIME,X(7);
                BRANCH,1:
                        ALWAYS, INSPECT5A;
INSPECT5A       QUEUE, I5A:  MARK(INSP5TA);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:A INSPECTION TIME,INT(INSP5TA);
                BRANCH,1:
                        WITH, 1., GOOD5A:
                        ELSE,RW5A;

RW5A            COUNT:  A PARTS REWORKED:MARK(REWORKTA);


                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5A;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);
                RELEASE:  OPERATOR;
                TALLY:A REWORK TIME,INT(REWORKTA);
                ASSIGN:  X(5)=0;
                BRANCH,1:
                        WITH,1.,GOOD5A:
                        ELSE,SCRAP5A;
SCRAP5A         COUNT:  A PARTS SCRAPED;
```

```
                ASSIGN:X(8)=TNOW;
                DELAY:  UN(SCRAPT);
                ASSIGN:X(9)=TNOW-X(8);
                TALLY:A SCRAP TIME,X(9);
GOOD5A          ALTER: DECOUPLER5A, +1;
                SIGNAL: 2:  DISPOSE; VMILL AVAILABLE

;               PART TYPE 2

TYPE2           BRANCH,1:
                        IF, MR(DECOUPLER4B)   .EQ.0,
WAIT4B:
                        ELSE, GET4B;
WAIT4B          QUEUE, HOLD4B;
                WAIT: 5,1;
GET4B           QUEUE, UNLOAD4B;
                SEIZE:  OPERATOR;
                DELAY: RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER: DECOUPLER4B, -1;
                SIGNAL: 4; D4B SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5B;
PROCESS5B       ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, UNL2;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(10)=TNOW-X(6);
                TALLY:  B CYCLE TIME,X(10);
                BRANCH,1:
                        ALWAYS, INSPECT5B;
INSPECT5B       QUEUE, I5B:  MARK(INSP5TB);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:B INSPECTION TIME,INT(INSP5TB);
                BRANCH,1:
                        WITH, 1., GOOD5B:
                        ELSE, RW5B;
RW5B            COUNT:  B PARTS REWORKED:MARK(REWORKTB);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5B;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(5)=0;
                TALLY:B REWORK TIME,INT(REWORKTB);
```

```
                BRANCH,1:
                        WITH,1.,GOOD5B:
                        ELSE,SCRAP5B;
SCRAP5B         COUNT:  B PARTS SCRAPED;
                ASSIGN:X(11)=TNOW;
                DELAY: UN(SCRAPT);
                ASSIGN:X(12)=TNOW-X(11);
                TALLY:B SCRAP TIME,X(12);
GOOD5B          ALTER:  DECOUPLER5B,+1;
                SIGNAL:  2:  DISPOSE; VMILL AVAILABLE

;               PART TYPE 3

TYPE3           BRANCH,1:
                        IF, MR(DECOUPLER4C) .EQ.0,
WAIT4C:
                        ELSE, GET4C;
WAIT4C          QUEUE, HOLD4C;
                WAIT: 6,1;
GET4C           QUEUE, UNLOAD4C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER4C, -1;
                SIGNAL:  4; D4C SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5C;
PROCESS5C       ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, UNL3;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(13)=TNOW-X(6);
                TALLY:  C CYCLE TIME,X(13);
                BRANCH,1:
                        ALWAYS, INSPECT5C;
INSPECT5C       QUEUE, I5C:  MARK(INSP5TC);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:C INSPECTION TIME,INT(INSP5TC);
                BRANCH,1:
                        WITH, 1., GOOD5C:
                        ELSE,RW5C;
RW5C            COUNT:  C PARTS REWORKED:MARK(REWORKTC);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5C;
                SEIZE:  OPERATOR;
```

```
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(5)=0;
                    TALLY:C REWORK TIME,INT(REWORKTC);
                    BRANCH,1:
                            WITH,1.,GOOD5C:
                            ELSE,SCRAP5C;
SCRAP5C             COUNT:  C PARTS SCRAPED;
                    ASSIGN:X(14)=TNOW;
                    DELAY:UN(SCRAPT);
                    ASSIGN:X(15)=TNOW-X(14);
                    TALLY:C SCRAP TIME,X(15);
GOOD5C              ALTER:  DECOUPLER5C, +1;
                    SIGNAL:  2:  DISPOSE; VMILL AVAILABLE

;                   PART TYPE 4

TYPE4               BRANCH,1:
                            IF, MR(DECOUPLER4D) .EQ.0,
WAIT4D:
                            ELSE, GET4D;
WAIT4D              QUEUE, HOLD4D;
                    WAIT:  7,1;
GET4D               QUEUE, UNLOAD4D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER4D, -1;
                    SIGNAL:  4; D4D SIGNAL FOR REPL.
                    BRANCH:
                            ALWAYS, HMILL:
                            ALWAYS, PROCESS5D;
PROCESS5D           ASSIGN:  X(5)=1;
                    DELAY:  RN(PT5);
                    QUEUE, UNL4;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(5)=0;
                    ASSIGN:X(16)=TNOW-X(6);
                    TALLY:  D CYCLE TIME,X(16);
                    BRANCH,1:
                            ALWAYS, INSPECT5D;
INSPECT5D           QUEUE, I5D:  MARK(INSP5TD);
                    SEIZE:  INSP5:  OPERATOR;
                    DELAY:  RN(INSPT);
                    RELEASE:  INSP5:  OPERATOR;
                    TALLY:D INSPECTION TIME,INT(INSP5TD);
                    BRANCH,1:
                            WITH, 1., GOOD5D:
                            ELSE, RW5D;
```

```
RW5D            COUNT:  D PARTS REWORKED:MARK(REWORKTD);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5D;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(5)=0;
                TALLY:D REWORK TIME,INT(REWORKTD);
                BRANCH,1:
                        WITH,1.,GOOD5D:
                        ELSE,SCRAP5D;
SCRAP5D         COUNT:  D PARTS SCRAPED;
                ASSIGN:X(17)=TNOW;
                DELAY:UN(SCRAPT);
                ASSIGN:X(18)=TNOW-X(17);
                TALLY:D SCRAP TIME,X(18);
GOOD5D          ALTER:  DECOUPLER5D, +1;
                SIGNAL:  2:  DISPOSE; VMILL AVAILABLE

;               AC4 (HORIZONTAL MILL & DECOUPLER 3)

HMILL           ASSIGN:X(19)=TNOW;
                QUEUE, AC4:  DETACH;
                CREATE;
BR3             BRANCH,1:
                       IF,NQ(AC4).GT.0,GETORDER:
                       ELSE,WAIT2;
WAIT2           QUEUE,HOLDBLOCKH;
                WAIT:4,1:NEXT(BR3);
GETORDER        REMOVE:1,AC4,BR4:NEXT(BR3);
BR4             BRANCH,1:
                        IF, X(4) .GT.0, BUSYHMILL:
                        ELSE, IDLEHMILL;
BUSYHMILL       QUEUE, DELAYATH;
                WAIT:  8,1;
IDLEHMILL       BRANCH,1:
                        IF, PARTTYPE==1, TYP1:
                        IF, PARTTYPE==2, TYP2:
                        IF, PARTTYPE==3, TYP3:
                        ELSE, TYP4;

TYP1            BRANCH,1:
                        IF, MR(DECOUPLER3A) .EQ.0,
WAITD3A:
                        ELSE, GETAFD3;
WAITD3A         QUEUE, HOLD3A;
                WAIT:  9,1;
GETAFD3         QUEUE, UNLOAD3A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
```

```
                    ALTER: DECOUPLER3A, -1;
                    SIGNAL:  10; D3A SIGNAL FOR REPL.
                    BRANCH:
                            ALWAYS, LATHE:
                            ALWAYS, PROCESS4A;
PROCESS4A           ASSIGN:  X(4)=1; BUSY HMILL
                    DELAY:  RN(PT4);
                    QUEUE, UNL5;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(4)=0;
                    SIGNAL:  8; HMILL AVAILABLE
                    ALTER:  DECOUPLER4A, +1;
                    ASSIGN:X(20)=TNOW-X(19);
                    TALLY:DECOUPLER4A REPL. TIME,X(20);
                    SIGNAL:  3:  DISPOSE; D4A REPL. SIGNAL

;                   PART TYPE 2

TYP2                BRANCH,1:
                            IF, MR(DECOUPLER3B) .EQ.0,
WAITD3B:
                            ELSE, GETBFD3;
WAITD3B             QUEUE, HOLD3B;
                    WAIT:  11,1;
GETBFD3             QUEUE, UNLOAD3B;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER3B, -1;
                    SIGNAL:  10; D3B SIGNAL FOR REPL.
                    BRANCH:
                            ALWAYS, LATHE:
                            ALWAYS, PROCESS4B;
PROCESS4B           ASSIGN:  X(4)=1;
                    DELAY:  RN(PT4);
                    QUEUE, UNL6;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(4)=0;
                    SIGNAL:  8; HMILL AVAILABLE
                    ALTER:  DECOUPLER4B, +1;
                    ASSIGN:X(21)=TNOW-X(19);
                    TALLY:DECOUPLER4B REPL. TIME,X(21);
                    SIGNAL:  5:  DISPOSE; D4B REPL. SIGNAL

;                   PART TYPE 3

TYP3                BRANCH,1:
                            IF, MR(DECOUPLER3C) .EQ.0,
```

```
WAITD3C:
                        ELSE, GETCFD3;
WAITD3C         QUEUE, HOLD3C;
                WAIT:  12,1;
GETCFD3         QUEUE, UNLOAD3C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER3C, -1;
                SIGNAL:  10; D3C SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4C;
PROCESS4C       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL7;
                SEIZE:  OPERATOR;
                DELAY:  UNLOADT;
                RELEASE: OPERATOR;
                ASSIGN:  X(4)=0;
                SIGNAL:  8; HMILL AVAILABLE
                ALTER:  DECOUPLER4C, +1;
                ASSIGN:X(22)=TNOW-X(19);
                TALLY:DECOUPLER4C REPL. TIME,X(22);
                SIGNAL:  6:  DISPOSE; D4C REPL. SIGNAL

;               PART TYPE 4

TYP4            BRANCH,1:
                        IF, MR(DECOUPLER3D) .EQ.0,
WAITD3D:
                        ELSE, GETDFD3;
WAITD3D         QUEUE, HOLD3D;
                WAIT:  13,1;
GETDFD3         QUEUE, UNLOAD3D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER3D, -1;
                SIGNAL:  10; D3D SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4D;
PROCESS4D       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL8;
                SEIZE:  OPERATOR;
                RELEASE:   OPERATOR;
                ASSIGN: X(4)=0;
                SIGNAL:8; HMILL AVAILABLE
                ALTER:  DECOUPLER4D, +1;
                ASSIGN:X(23)=TNOW-X(19);
```

```
                    TALLY:DECOUPLER4D REPL. TIME,X(23);
                    SIGNAL:  7:  DISPOSE; D4D REPL. SIGNAL

;                   AC3 (LATHE & DECOUPLER 2)

LATHE               QUEUE, AC3:  DETACH;
                    CREATE;
BR5                 BRANCH,1:
                             IF, NQ(AC3) .GT.0, NEXTORDER:
                             ELSE, WAIT3;
WAIT3               QUEUE, HOLDBLOCKL;
                    WAIT:  10:  NEXT(BR5);
NEXTORDER           REMOVE:  1, AC3, BR6:  NEXT (BR5);
BR6                 BRANCH,1:
                             IF, X(3) .GT.0, BUSYLATHE:
                             ELSE, IDLELATHE;
BUSYLATHE           QUEUE, DELAYATL;
                    WAIT:  14;
IDLELATHE           BRANCH,1:
                             IF, PARTTYPE==1, TY1:
                             IF, PARTTYPE==2, TY2:
                             IF, PARTTYPE==3, TY3:
                             ELSE, TY4;

TY1                 BRANCH,1:
                             IF, MR(DECOUPLER2A) .EQ.0,
WAITD2A:
                             ELSE, GETAFD2;
WAITD2A             QUEUE, HOLD2A;
                    WAIT:  15,1;
GETAFD2             QUEUE, UNLOAD2A;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER2A, -1;
                    SIGNAL:  19; D2A SIGNAL FOR REPL.
                    BRANCH:
                             ALWAYS, DRILL:
                             ALWAYS, PROCESS3A;
PROCESS3A           ASSIGN:  X(3)=1;
                    DELAY:  RN(PT3);
                    QUEUE, UNL9;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(3)=0;
                    SIGNAL:  14; LATHE AVAILABLE
                    ALTER:  DECOUPLER3A, +1;
                    SIGNAL:  9:  DISPOSE; D3A REPL. SIGNAL

;                   PART TYPE 2
```

```
TY2             BRANCH,1:
                        IF, MR(DECOUPLER2B) .EQ.0,
WAITD2B:
                        ELSE, GETBFD2;
WAITD2B         QUEUE, HOLD2B;
                WAIT:  16,1;
GETBFD2         QUEUE, UNLOAD2B;
                SEIZE:  OPERATOR;
                DELAY: RN( WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2B, -1;
                SIGNAL:  19; D2B SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3B;
PROCESS3B       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL10;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14; LATHE AVAILABLE
                ALTER:  DECOUPLER3B, +1;
                SIGNAL:  11:  DISPOSE; D3B REPL. SIGNAL

;               PART TYPE 3

TY3             BRANCH,1:
                        IF, MR(DECOUPLER2C) .EQ.0,
WAITD2C:
                        ELSE, GETCFD2;
WAITD2C         QUEUE, HOLD2C;
                WAIT:  17,1;
GETCFD2         QUEUE, UNLOAD2C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2C, -1;
                SIGNAL:  19; D2C SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3C;
PROCESS3C       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL11;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14; LATHE AVAILABLE
                ALTER:  DECOUPLER3C, +1;
```

```
                    SIGNAL:  12:  DISPOSE; D3C REPL. SIGNAL

;                   PART TYPE 4

TY4                 BRANCH,1:
                            IF, MR(DECOUPLER2D) .EQ.0,
WAITD2D:
                            ELSE, GETDFD2;
WAITD2D             QUEUE, HOLD2D;
                    WAIT:  18,1;
GETDFD2             QUEUE, UNLOAD2D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER2D, -1;
                    SIGNAL:  19; D2D SIGNAL FOR REPL.
                    BRANCH:
                            ALWAYS, DRILL:
                            ALWAYS, PROCESS3D;
PROCESS3D           ASSIGN:  X(3)=1;
                    DELAY:RN(PT3);
                    QUEUE, UNL12;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(3)=0;
                    SIGNAL:  14; LATHE AVAILABLE
                    ALTER:  DECOUPLER3D, +1;
                    SIGNAL:  13:  DISPOSE; D3D REPL. SIGNAL

;                   AC2 (DRILL & DECOUPLER 1)

DRILL               QUEUE, AC2:  DETACH;
                    CREATE;
BR7                 BRANCH,1:
                            IF, NQ(AC2) .GT.0, NEXTENTITY:
                            ELSE, WAIT4;
WAIT4               QUEUE, HOLDBLOCKD;
                    WAIT:  19,1:  NEXT(BR7);
NEXTENTITY          REMOVE:  1,AC2, BR8:  NEXT(BR7);
BR8                 BRANCH,1:
                            IF, X(2) .GT.0, BUSYDRILL:
                            ELSE, IDLEDRILL;
BUSYDRILL           QUEUE, DELAYATD;
                    WAIT:  20,1;
IDLEDRILL           BRANCH,1:
                            IF, PARTTYPE==1, PART1:
                            IF, PARTTYPE==2, PART2:
                            IF, PARTTYPE==3, PART3:
                            ELSE, PART4;

PART1               BRANCH,1:
```

```
                        IF, MR(DECOUPLER1A) .EQ.0,
WAITD1A:
                        ELSE, GETAFD1;
WAITD1A         QUEUE, HOLD1A;
                WAIT:  21,1;
GETAFD1         QUEUE, UNLOAD1A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1A, -1;
                SIGNAL:  25; D1A SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2A;
PROCESS2A       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL13;
                SEIZE:  OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
                ALTER:  DECOUPLER2A, +1;
                SIGNAL:  15:  DISPOSE; D2A REPL. SIGNAL

;               PART TYPE 2

PART2           BRANCH,1:
                        IF, MR(DECOUPLER1B) .EQ.0,
WAITD1B:
                        ELSE, GETBFD1;
WAITD1B         QUEUE, HOLD1B;
                WAIT:  22,1;
GETBFD1         QUEUE, UNLOAD1B;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1B, -1;
                SIGNAL:  25; D1B SIGNAL FOR REPL.


                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2B;
PROCESS2B       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL14;
                SEIZE:  OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
```

```
                ALTER:  DECOUPLER2B, +1;
                SIGNAL:  16:  DISPOSE; D2A REPL. SIGNAL


;               PART TYPE 3

PART3           BRANCH,1:
                        IF, MR(DECOUPLER1C) .EQ.0,
WAITD1C:
                        ELSE, GETCFD1;
WAITD1C         QUEUE, HOLD1C;
                WAIT:  23,1;
GETCFD1         QUEUE, UNLOAD1C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1C, -1;
                SIGNAL:  25; D1C SIGNAL FOR REPL.
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2C;
PROCESS2C       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL15;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20; DRILL AVAILABLE
                ALTER:  DECOUPLER2C, +1;
                SIGNAL:  17:  DISPOSE; D2A REPL. SIGNAL


;               PART TYPE 4

PART4           BRANCH,1:
                        IF, MR(DECOUPLER1D) .EQ.0,
WAITD1D:
                        ELSE, GETDFD1;
WAITD1D         QUEUE, HOLD1D;
                WAIT:  24,1;
GETDFD1         QUEUE, UNLOAD1D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1D, -1;
                SIGNAL:  25; D1D SIGNAL FOR REPLACEMENT
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2D;
PROCESS2D       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
```

```
                    QUEUE, UNL16;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(2)=0;
                    SIGNAL:  20; DRILL AVAILABLE
                    ALTER:  DECOUPLER2D, +1;
                    SIGNAL:  18:  DISPOSE; D2A REPL. SIGNAL


;                   AC1 (SPINDLE MACHINE)

SPINDLE             QUEUE, AC1:  DETACH;
                    CREATE;
BR9                 BRANCH,1:
                            IF, NQ(AC1) .GT.0, M1:
                            ELSE, WAIT5;
WAIT5               QUEUE, HOLDBLOCKS;
                    WAIT:  25,1:  NEXT (BR9);
M1                  REMOVE:  1, AC1, BR10:  NEXT (BR9);
BR10                BRANCH,1:
                            IF, X(1) .GT.0, BUSYSPINDLE:
                            ELSE, IDLESPINDLE;
BUSYSPINDLE         QUEUE, DELAYATS;
                    WAIT:  26,1;
IDLESPINDLE         BRANCH,1:
                            IF, PARTTYPE==1, PT1:
                            IF, PARTTYPE==2, PT2:
                            IF, PARTTYPE==3, PT3:
                            ELSE, PT4;

PT1                 QUEUE, CARTA;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
                    DELAY:  LOADT; LOAD/POSITION/START
MACHINE
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLEA;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(1)=0;
                    SIGNAL:  26; SPINDLE AVAILABLE
                    ALTER:  DECOUPLER1A, +1;
                    SIGNAL:  21:DISPOSE; D1A REPL. SIGNAL

PT2                 QUEUE, CARTB;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
```

```
                    DELAY: RN(LOADT);
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLEB;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(1)=0;
                    SIGNAL:  26; SPINDLE AVAILABLE
                    ALTER:  DECOUPLER1B, +1;
                    SIGNAL:  22:DISPOSE; D1B REPL. SIGNAL

PT3                 QUEUE, CARTC;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
                    DELAY:  RN(LOADT);
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLEC;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(1)=0;
                    SIGNAL:  26; SPINDLE AVAILABLE
                    ALTER:  DECOUPLER1C, +1;
                    SIGNAL:  23:DISPOSE; D1C REPL. SIGNAL

PT4                 QUEUE, CARTD;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
                    DELAY:  RN(LOADT);
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLED;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(1)=0;
                    SIGNAL:  26; SPINDLE AVAILABLE
                    ALTER:  DECOUPLER1D, +1;
                    SIGNAL:  24:DISPOSE; D1D REPL. SIGNAL

 BEGINSETUP         ASSIGN:X(24)=TNOW;
                    QUEUE,Q0;
                    SEIZE:OPERATOR;
                    ASSIGN:X(25)=TNOW;
                    ASSIGN:X(1)=1;
                    DELAY:UN(ST1);
                    ASSIGN:X(1)=0;
                    ASSIGN:X(26)=TNOW-X(25);
```

```
                    TALLY:SPINDLE SETUP,X(26);

                    DELAY:RN(WALKT);

                    ASSIGN:X(27)=TNOW;
                    ASSIGN:X(2)=1;
                    DELAY:UN(ST2);
                    ASSIGN:X(2)=0;
                    ASSIGN:X(28)=TNOW-X(27);
                    TALLY:DRILL SETUP,X(28);

                    DELAY:RN(WALKT);

                    ASSIGN:X(29)=TNOW;
                    ASSIGN:X(3)=1;
                    DELAY:UN(ST3);
                    ASSIGN:X(3)=0;
                    ASSIGN:X(30)=TNOW-X(29);
                    TALLY:LATHE SETUP,X(30);

                    DELAY:RN(WALKT);

                    ASSIGN:X(31)=TNOW;
                    ASSIGN:X(4)=1;
                    DELAY:UN(ST4);
                    ASSIGN:X(4)=0;
                    ASSIGN:X(32)=TNOW-X(31);
                    TALLY:HMILL SETUP,X(32);

                    DELAY:RN(WALKT);

                    ASSIGN:X(33)=TNOW;
                    ASSIGN:X(5)=1;
                    DELAY:UN(ST5);
                    ASSIGN:X(5)=0;
                    ASSIGN:X(34)=TNOW-X(33);
                    TALLY:VMILL SETUP,X(34);
                    DELAY:RN(WALKT);
                    ASSIGN:X(35)=TNOW-X(24);
                    TALLY:TOTAL SETUP,X(35);
                    RELEASE:OPERATOR:NEXT(BEGINCYCLE);

END;
```

```
BEGIN;

PROJECT, CELLULAR MANUFACTURING, N ALTENEH;

PARAMETERS:     1,ORDERIN,8     :! TIME BETWEEN ARRIVALS
                2,WALKT,.5,.1   :! WALKING TIME
                3,INSPT,1,.1    :! INSPECTION TIME
                4,UNLOADT,.5,.1:! UNLOADING MACHINES
                5,PT5,3.5,.45   :! PROCESSING TIMES
                6,PT4,3.5,.45   :
                7,PT3,3.5,.45   :
                8,PT2,3.5,.45   :
                9,PT1,3.5,.45   :
                10,PAPART,.5,.1:! PICK A PART
                11,LOADT,.8,.1  :! LOADING TIME
                12,SCRAPT,2,4   :! SCRAP TIME
                13,ST1,2,3      :! SETUP TIMES
                14,ST2,1,2      :
                16,ST3,4,5      :
                17,ST4,1,1.5    :
                18,ST5,0,.5     ;

ATTRIBUTES:     INSP5TA :         INSP5TB  :
                INSP5TC :         INSP5TD  :
                REWORKTA:         REWORKTB :
                REWORKTC:         REWORKTD :
                PARTTYPE:         ISETUP   :
                ASETUP  :         BSETUP   :
                CSETUP  :         DSETUP   ;

QUEUES:         UNLOAD4A:         UNLOAD4B:
                UNLOAD4C:         UNLOAD4D:
                UNLOAD3A:         UNLOAD3B:
                UNLOAD3C:         UNLOAD3D:
                UNLOAD2A:         UNLOAD2B:
                UNLOAD2C:         UNLOAD2D:
                UNLOAD1A:         UNLOAD1B:
                UNLOAD1C:         UNLOAD1D:
                REWORK5A:         REWORK5B:
                REWORK5C:         REWORK5D:
                DELAYATV:         DELAYATH:
                DELAYATL:         DELAYATD:
                DELAYATS:         HOLDAT5 :
                HOLD4A  :         HOLD4B  :
                HOLD4C  :         HOLD4D  :
                HOLD2A  :         HOLD1B  :
                HOLD3A  :         HOLD3B  :
                HOLD3C  :         HOLD3D  :
                HOLD2B  :         HOLD2C  :
                HOLD2D  :         HOLD1A  :
                HOLD1C  :         HOLD1D  :
                ECYCLEA :         ECYCLEB :
```

```
                ECYCLEC :          ECYCLED :
                CARTA   :          CARTB   :
                CARTC   :          CARTD   :
                I5A: I5B:  I5C:    I5D     :
                AC4: AC3:  AC2:    AC1     :
                UNL1:  UNL2:  UNL3:  UNL4:
                UNL5:  UNL6:  UNL7:  UNL8:
                UNL9: UNL10: UNL11: UNL12:
                UNL13: UNL14: UNL15:UNL16:
                ARRIVEBLOCK:Q0:HOLDBLOCKS:
                HOLDBLOCKD:    HOLDBLOCKL:
                HOLDBLOCKH                ;

RESOURCES:      DECOUPLER1A,SCHED(1) :
                DECOUPLER1B,SCHED(2) :
                DECOUPLER1C,SCHED(3) :
                DECOUPLER1D,SCHED(4) :
                DECOUPLER2A,SCHED(5) :
                DECOUPLER2B,SCHED(6) :
                DECOUPLER2C,SCHED(7) :
                DECOUPLER2D,SCHED(8) :
                DECOUPLER3A,SCHED(9) :
                DECOUPLER3B,SCHED(10):
                DECOUPLER3C,SCHED(11):
                DECOUPLER3D,SCHED(12):
                DECOUPLER4A,SCHED(13):
                DECOUPLER4B,SCHED(14):
                DECOUPLER4C,SCHED(15):
                DECOUPLER4D,SCHED(16):
                DECOUPLER5A,SCHED(17):
                DECOUPLER5B,SCHED(18):
                DECOUPLER5C,SCHED(19):
                DECOUPLER5D,SCHED(20):
                OPERATOR:       INSP5;

SCHEDULES:      1,1 :   2,1 :   3,1 :   4,1 :  5,1 :
                6,1 :   7,1 :   8,1 :   9,1 :  10,1:
                11,1:  12,1 :  13,1 :  14,1 :  15,1:
                16,1:  17,1 :  18,1 :  19,1 :  20,1;

DSTAT:          NR(OPERATOR) * 100, OPERATOR UTIL:
                X(1)  * 100, SPINDLE UTIL:
                X(2)  * 100, DRILL UTIL  :
                X(3)  * 100, LATHE UTIL  :
                X(4) * 100, HMILL UTIL   :
                X(5) * 100, VMILL UTIL   :
                (X(1) + X(2) + X(3)      +
                X(4) + X(5))* 100/5, CELL UTIL :
                NQ(DELAYATV), NO. DELAYED AT M5:
                NQ(DELAYATH), NO. DELAYED AT M4:
                NQ(DELAYATL), NO. DELAYED AT M3:
                NQ(DELAYATD), NO. DELAYED AT M2:
```

```
                NQ(DELAYATS), NO. DELAYED AT M1:
                MR(DECOUPLER4A), NO. IN D4A    :
                MR(DECOUPLER4B), NO. IN D4B    :
                MR(DECOUPLER4C), NO. IN D4C    :
                MR(DECOUPLER4D), NO. IN D4D    :
                MR(DECOUPLER3A), NO. IN D3A    :
                MR(DECOUPLER3B), NO. IN D3B    :
                MR(DECOUPLER3C), NO. IN D3C    :
                MR(DECOUPLER3D), NO. IN D3D    :
                MR(DECOUPLER2A), NO. IN D2A    :
                MR(DECOUPLER2B), NO. IN D2B    :
                MR(DECOUPLER2C), NO. IN D2C    :
                MR(DECOUPLER2D), NO. IN D2D    :
                MR(DECOUPLER1A), NO. IN D1A    :
                MR(DECOUPLER1B), NO. IN D1B    :
                MR(DECOUPLER1C), NO. IN D1C    :
                MR(DECOUPLER1D), NO. IN D1D    ;

COUNTERS:       NO. OF REQUESTS,,    NO:
                NO. OF CYCLES,,      NO:
                NO. OF CHANGEOVERS,, NO:
                A PARTS REWORKED,,   NO:
                A PARTS SCRAPED,,    NO:
                B PARTS REWORKED,,   NO:
                B PARTS SCRAPED,,    NO:
                C PARTS REWORKED,,   NO:
                C PARTS SCRAPED,,    NO:
                D PARTS REWORKED,,   NO:
                D PARTS SCRAPED,,    NO;

TALLIES:        SPINDLE SETUP:
                DRILL SETUP  :
                LATHE SETUP  :
                HMILL SETUP  :
                VMILL SETUP  :
                TOTAL SETUP  :
                A CYCLE TIME :
                A INSPECTION TIME:
                A REWORK TIME:
                A SCRAP TIME :
                B CYCLE TIME :
                B INSPECTION TIME:
                B REWORK TIME:
                B SCRAP TIME :
                C CYCLE TIME :
                C INSPECTION TIME:
                C REWORK TIME:
                C SCRAP TIME :
                D CYCLE TIME :
                D INSPECTION TIME:
                D REWORK TIME:
                D SCRAP TIME :
```

```
                    DECOUPLER4A REPL. TIME:
                    DECOUPLER4B REPL. TIME:
                    DECOUPLER4C REPL. TIME:
                    DECOUPLER4D REPL. TIME;

REPLICATE,,0,480;
END;
```

APPENDIX C

Simulation Model With Increased Product Mix

```
BEGIN;
;                       SIMULATION BASE MODEL

;                          PRODUCT MIX = 6

;          CONTRIBUTION OF CELLULAR MANUFACTURING DESIGN TO

;                  THE REDUCTION OF PRODUCTION LEAD TIME

;                       UTILIZING SIMULATION MODELING

;                                  **********

                   CREATE: EX(ORDERIN);
                   COUNT:NO. OF REQUESTS;
                   ASSIGN:PARTTYPE=DISCRETE
                   (.17,1,.31,2,.45,3,.56,4,.78,5,1.0,6);
ICYCLE             ASSIGN:X(6)=TNOW;
                   SIGNAL:1;
                   QUEUE,ARRIVEBLOCK:DETACH;

                   CREATE;
BR1                BRANCH,1:
                            IF,NQ(ARRIVEBLOCK).GT.0,RMV1:
                            ELSE,WAIT1;
WAIT1              QUEUE,HOLDAT5;
                   WAIT:1,1:NEXT(BR1);
RMV1               REMOVE:1,ARRIVEBLOCK,BR2:NEXT(BR1);

;                  AC5 (VERTICAL MACHINE & DECOUPLER 4)

BR2                BRANCH,1:
                            IF,X(5).GT.0,BUSYVMILL:
                            ELSE,IDLEVMILL;
BUSYVMILL          QUEUE,DELAYATV;
                   WAIT:2,1;
IDLEVMILL          BRANCH,1:
                            IF,PARTTYPE==1,TYPE1:
                            IF,PARTTYPE==2,TYPE2:
                            IF,PARTTYPE==3,TYPE3:
                            IF,PARTTYPE==4,TYPE4:
                            IF,PARTTYPE==5,TYPE5:
                            ELSE,TYPE6;

TYPE1              BRANCH,1:
                            IF,MR(DECOUPLER4A).EQ.0,WAIT4A:
                            ELSE,GET4A;
WAIT4A             QUEUE,HOLD4A;
                   WAIT:3,1;
GET4A              QUEUE, UNLOAD4A;
                   SEIZE: OPERATOR;
```

```
                    DELAY: RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER: DECOUPLER4A,-1;
                    SIGNAL:  4;
                    BRANCH:
                            ALWAYS, HMILL:
                            ALWAYS, PROCESS5A;
PROCESS5A           ASSIGN:  X(5)=1; BUSY VMILL
                    DELAY: RN(PT5);
                    QUEUE, UNL1;
                    SEIZE:  OPERATOR;
                    DELAY: RN(UNLOADT);UNLOAD PART FOR INSP.
                    RELEASE: OPERATOR;
                    ASSIGN:  X(5)=0;
                    ASSIGN:X(7)=TNOW-X(6);
                    TALLY:  A CYCLE TIME,X(7);

                    BRANCH,1:
                            ALWAYS, INSPECT5A;

INSPECT5A           QUEUE, I5A:  MARK(INSP5TA);
                    SEIZE:  INSP5:  OPERATOR;
                    DELAY:  RN(INSPT);
                    RELEASE:  INSP5:  OPERATOR;
                    TALLY:A INSPECTION TIME,INT(INSP5TA);
                    BRANCH,1:
                            WITH, 1., GOOD5A:
                            ELSE,RW5A;
RW5A                COUNT:  A PARTS REWORKED:MARK(REWORKTA);
                    ASSIGN:  X(5)=1;
                    DELAY:  RN(PT5);
                    QUEUE, REWORK5A;
                    SEIZE:  OPERATOR;
                    DELAY: RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    TALLY:A REWORK TIME,INT(REWORKTA);
                    ASSIGN:  X(5)=0;
                    BRANCH,1:
                            WITH,1.,GOOD5A:
                            ELSE,SCRAP5A;
SCRAP5A             COUNT:  A PARTS SCRAPED;
                    ASSIGN:X(8)=TNOW;
                    DELAY:  UN(SCRAPT);
                    ASSIGN:X(9)=TNOW-X(8);
                    TALLY:A SCRAP TIME,X(9);
GOOD5A              ALTER: DECOUPLER5A, +1;
                    SIGNAL: 2:  DISPOSE;

;                   PART TYPE 2

TYPE2               BRANCH,1:
                            IF, MR(DECOUPLER4B)  .EQ.0,
```

```
WAIT4B:
                        ELSE, GET4B;
WAIT4B          QUEUE, HOLD4B;
                WAIT: 5,1;
GET4B           QUEUE, UNLOAD4B;
                SEIZE:  OPERATOR;
                DELAY: RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER: DECOUPLER4B, -1;
                SIGNAL: 4;
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5B;
PROCESS5B       ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, UNL2;
                SEIZE:  OPERATOR;
                DELAY: RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(10)=TNOW-X(6);
                TALLY:  B CYCLE TIME,X(10);
                BRANCH,1:
                        ALWAYS, INSPECT5B;
INSPECT5B       QUEUE, I5B:  MARK(INSP5TB);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:B INSPECTION TIME,INT(INSP5TB);
                BRANCH,1:
                        WITH, 1., GOOD5B:
                        ELSE, RW5B;

RW5B            COUNT:  B PARTS REWORKED:MARK(REWORKTB);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5B;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(5)=0;
                TALLY:B REWORK TIME,INT(REWORKTB);
                BRANCH,1:
                        WITH,1.,GOOD5B:
                        ELSE,SCRAP5B;

SCRAP5B         COUNT:  B PARTS SCRAPED;
                ASSIGN:X(11)=TNOW;
                DELAY: UN(SCRAPT);
                ASSIGN:X(12)=TNOW-X(11);
                TALLY:B SCRAP TIME,X(12);
GOOD5B          ALTER:  DECOUPLER5B,+1;
```

```
                    SIGNAL:  2:  DISPOSE;

;                   PART TYPE 3

TYPE3               BRANCH,1:
                            IF, MR(DECOUPLER4C) .EQ.0,
WAIT4C:
                            ELSE, GET4C;
WAIT4C              QUEUE, HOLD4C;
                    WAIT: 6,1;
GET4C               QUEUE, UNLOAD4C;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER4C, -1;
                    SIGNAL:  4;
                    BRANCH:
                            ALWAYS, HMILL:
                            ALWAYS, PROCESS5C;
PROCESS5C           ASSIGN:  X(5)=1;
                    DELAY:  RN(PT5);
                    QUEUE, UNL3;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(5)=0;
                    ASSIGN:X(13)=TNOW-X(6);
                    TALLY:  C CYCLE TIME,X(13);
                    BRANCH,1:
                            ALWAYS, INSPECT5C;
INSPECT5C           QUEUE, I5C:  MARK(INSP5TC);
                    SEIZE:  INSP5:  OPERATOR;
                    DELAY:  RN(INSPT);
                    RELEASE:  INSP5:  OPERATOR;
                    TALLY:C INSPECTION TIME,INT(INSP5TC);
                    BRANCH,1:
                            WITH, 1., GOOD5C:
                            ELSE,RW5C;
RW5C                COUNT:  C PARTS REWORKED:MARK(REWORKTC);
                    ASSIGN:  X(5)=1;
                    DELAY:  RN(PT5);
                    QUEUE, REWORK5C;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(5)=0;
                    TALLY:C REWORK TIME,INT(REWORKTC);
                    BRANCH,1:
                            WITH,1.,GOOD5C:
                            ELSE,SCRAP5C;
SCRAP5C             COUNT:  C PARTS SCRAPED;
                    ASSIGN:X(14)=TNOW;
```

```
                DELAY:UN(SCRAPT);
                ASSIGN:X(15)=TNOW-X(14);
                TALLY:C SCRAP TIME,X(15);
GOOD5C          ALTER:  DECOUPLER5C, +1;
                SIGNAL:  2:  DISPOSE;

;               PART TYPE 4

TYPE4           BRANCH,1:
                        IF, MR(DECOUPLER4D) .EQ.0,
WAIT4D:
                        ELSE, GET4D;
WAIT4D          QUEUE, HOLD4D;
                WAIT:  7,1;
GET4D           QUEUE, UNLOAD4D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER4D, -1;
                SIGNAL:  4;
                BRANCH:
                        ALWAYS, HMILL:
                        ALWAYS, PROCESS5D;
PROCESS5D       ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, UNL4;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(5)=0;
                ASSIGN:X(16)=TNOW-X(6);
                TALLY:  D CYCLE TIME,X(16);
                BRANCH,1:
                        ALWAYS, INSPECT5D;
INSPECT5D       QUEUE, I5D:  MARK(INSP5TD);
                SEIZE:  INSP5:  OPERATOR;
                DELAY:  RN(INSPT);
                RELEASE:  INSP5:  OPERATOR;
                TALLY:D INSPECTION TIME,INT(INSP5TD);
                BRANCH,1:
                        WITH, 1., GOOD5D:
                        ELSE, RW5D;
RW5D            COUNT:  D PARTS REWORKED:MARK(REWORKTD);
                ASSIGN:  X(5)=1;
                DELAY:  RN(PT5);
                QUEUE, REWORK5D;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(5)=0;
                TALLY:D REWORK TIME,INT(REWORKTD);
                BRANCH,1:
```

```
                        WITH,1.,GOOD5D:
                        ELSE,SCRAP5D;
SCRAP5D         COUNT:  D PARTS SCRAPED;
                ASSIGN:X(17)=TNOW;
                DELAY:UN(SCRAPT);
                ASSIGN:X(18)=TNOW-X(17);
                TALLY:D SCRAP TIME,X(18);
GOOD5D          ALTER:  DECOUPLER5D, +1;
                SIGNAL:  2:  DISPOSE;

;               PART TYPE 5

TYPE5           BRANCH,1:
                       IF,MR(DECOUPLER4E).EQ.0,WAIT4E:
                       ELSE,GET4E;
WAIT4E          QUEUE,HOLD4E;
                WAIT:30,1;
GET4E           QUEUE,UNLOAD4E;
                SEIZE:OPERATOR;
                DELAY:RN(WALKT);
                RELEASE:OPERATOR;
                ALTER:DECOUPLER4E,-1;
                SIGNAL:4;
                BRANCH:
                       ALWAYS,HMILL:
                       ALWAYS,PROCESS5E;
PROCESS5E       ASSIGN:X(5)=1;
                DELAY:RN(PT5);
                QUEUE,UNL16;
                SEIZE:OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:OPERATOR;
                ASSIGN:X(5)=0;
                BRANCH,1:
                        ALWAYS,INSPECT5E;
INSPECT5E       QUEUE,I5E:MARK(INSP5TE);
                SEIZE:INSP5:OPERATOR;
                DELAY:RN(INSPT);
                RELEASE:INSP5:OPERATOR;
                TALLY:E INSPECTION TIME,INT(INSP5TE);
                BRANCH,1:
                        WITH,1.,GOOD5E:
                        ELSE,RW5E;
RW5E            COUNT: E PARTS REWORKED:MARK(REWORKTE);
                ASSIGN:X(5)=1;
                DELAY:RN(PT5);
                QUEUE,REWORK5E;
                SEIZE:OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:OPERATOR;
                ASSIGN:X(5)=0;
                TALLY:E REWORK TIME,INT(REWORKTE);
```

```
                BRANCH,1:
                        WITH,1.,GOOD5E:
                        ELSE,SCRAP5E;
SCRAP5E         COUNT:E PARTS SCRAPED;
                ASSIGN:X(37)=TNOW;
                DELAY:UN(SCRAPT);
                ASSIGN:X(38)=TNOW-X(37);
                TALLY:E SCRAP TIME,X(38);
GOOD5E          ALTER:DECOUPLER5E,+1;
                SIGNAL:2:DISPOSE;

;               PART TYPE 6

TYPE6           BRANCH,1:
                       IF,MR(DECOUPLER4F).EQ.0,WAIT4F:
                       ELSE,GET4F;
WAIT4F          QUEUE,HOLD4F;
                WAIT:31,1;
GET4F           QUEUE,UNLOAD4F;
                SEIZE:OPERATOR;
                DELAY:RN(WALKT);
                RELEASE:OPERATOR;
                ALTER:DECOUPLER4F,-1;
                SIGNAL:4;
                BRANCH:
                       ALWAYS,HMILL:
                       ALWAYS,PROCESS5F;
PROCESS5F       ASSIGN:X(5)=1;
                DELAY:RN(PT5);
                QUEUE,UNL17;
                SEIZE:OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:OPERATOR;
                ASSIGN:X(5)=0;
                BRANCH,1:
                        ALWAYS,INSPECT5F;
INSPECT5F       QUEUE,I5F:MARK(INSP5TF);
                SEIZE:INSP5:OPERATOR;
                DELAY:RN(INSPT);
                RELEASE:INSP5:OPERATOR;
                TALLY:F INSPECTION TIME,INT(INSP5TF);
                BRANCH,1:
                        WITH,1.,GOOD5F:
                        ELSE,RW5F;
RW5F            COUNT: F PARTS REWORKED:MARK(REWORKTF);
                ASSIGN:X(5)=1;
                DELAY:RN(PT5);
                QUEUE,REWORK5F;
                SEIZE:OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:OPERATOR;
                ASSIGN:X(5)=0;
```

```
                    TALLY:F REWORK TIME,INT(REWORKTF);
                    BRANCH,1:
                            WITH,1.,GOOD5F:
                            ELSE,SCRAP5F;
SCRAP5F             COUNT:F PARTS SCRAPED;
                    ASSIGN:X(40)=TNOW;
                    DELAY:UN(SCRAPT);
                    ASSIGN:X(41)=TNOW-X(40);
                    TALLY:F SCRAP TIME,X(41);
GOOD5F              ALTER:DECOUPLER5F,+1;
                    SIGNAL:2:DISPOSE;


;                   AC4 (HORIZONTAL MILL & DECOUPLER 3)

HMILL               ASSIGN:X(19)=TNOW;
                    QUEUE, AC4:  DETACH;
                    CREATE;
BR3                 BRANCH,1:
                           IF,NQ(AC4).GT.0,GETORDER:
                           ELSE,WAIT2;
WAIT2               QUEUE,HOLDBLOCKH;
                    WAIT:4,1:NEXT(BR3);
GETORDER            REMOVE:1,AC4,BR4:NEXT(BR3);
BR4                 BRANCH,1:
                            IF, X(4) .GT.0, BUSYHMILL:
                            ELSE, IDLEHMILL;
BUSYHMILL           QUEUE, DELAYATH;
                    WAIT:  8,1;
IDLEHMILL           BRANCH,1:
                            IF, PARTTYPE==1, TYP1:
                            IF, PARTTYPE==2, TYP2:
                            IF, PARTTYPE==3, TYP3:
                            IF, PARTTYPE==4, TYP4:
                            IF, PARTTYPE==5, TYP5:
                            ELSE,TYPE6;

TYP1                BRANCH,1:
                            IF, MR(DECOUPLER3A) .EQ.0,
WAITD3A:
                            ELSE, GETAFD3;
WAITD3A             QUEUE, HOLD3A;
                    WAIT:  9,1;
GETAFD3             QUEUE, UNLOAD3A;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER: DECOUPLER3A, -1;
                    SIGNAL:  10;
                    BRANCH:
                            ALWAYS, LATHE:
                            ALWAYS, PROCESS4A;
```

```
PROCESS4A       ASSIGN:  X(4)=1; BUSY HMILL
                DELAY:  RN(PT4);
                QUEUE, UNL5;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(4)=0;
                SIGNAL:  8;
                ALTER:  DECOUPLER4A, +1;
                ASSIGN:X(20)=TNOW-X(19);
                TALLY:DECOUPLER4A REPL. TIME,X(20);
                SIGNAL:  3:  DISPOSE;

;               PART TYPE 2

TYP2            BRANCH,1:
                        IF, MR(DECOUPLER3B) .EQ.0,
WAITD3B:
                        ELSE, GETBFD3;
WAITD3B         QUEUE, HOLD3B;
                WAIT:  11,1;
GETBFD3         QUEUE, UNLOAD3B;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER3B, -1;
                SIGNAL:  10;
                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4B;
PROCESS4B       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL6;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(4)=0;
                SIGNAL:  8;
                ALTER:  DECOUPLER4B, +1;
                ASSIGN:X(21)=TNOW-X(19);
                TALLY:DECOUPLER4B REPL. TIME,X(21);
                SIGNAL:  5:  DISPOSE;

;               PART TYPE 3

TYP3            BRANCH,1:
                        IF, MR(DECOUPLER3C) .EQ.0,
WAITD3C:
                        ELSE, GETCFD3;
WAITD3C         QUEUE, HOLD3C;
                WAIT:  12,1;
GETCFD3         QUEUE, UNLOAD3C;
```

```
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER3C, -1;
                    SIGNAL:  10;
                    BRANCH:
                            ALWAYS, LATHE:
                            ALWAYS, PROCESS4C;
PROCESS4C           ASSIGN:  X(4)=1;
                    DELAY:  RN(PT4);
                    QUEUE, UNL7;
                    SEIZE:  OPERATOR;
                    DELAY:  UNLOADT;
                    RELEASE: OPERATOR;
                    ASSIGN:  X(4)=0;
                    SIGNAL:  8;
                    ALTER:  DECOUPLER4C, +1;
                    ASSIGN:X(22)=TNOW-X(19);
                    TALLY:DECOUPLER4C REPL. TIME,X(22);
                    SIGNAL:  6:  DISPOSE;

;                   PART TYPE 4

TYP4                BRANCH,1:
                            IF, MR(DECOUPLER3D) .EQ.0,
WAITD3D:
                            ELSE, GETDFD3;
WAITD3D             QUEUE, HOLD3D;
                    WAIT:  13,1;
GETDFD3             QUEUE, UNLOAD3D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER3D, -1;
                    SIGNAL:  10;
                    BRANCH:
                            ALWAYS, LATHE:
                            ALWAYS, PROCESS4D;
PROCESS4D           ASSIGN:  X(4)=1;
                    DELAY:  RN(PT4);
                    QUEUE, UNL8;
                    SEIZE:  OPERATOR;
                    RELEASE:  OPERATOR;
                    ASSIGN: X(4)=0;
                    SIGNAL:8;
                    ALTER:  DECOUPLER4D, +1;
                    ASSIGN:X(23)=TNOW-X(19);
                    TALLY:DECOUPLER4D REPL. TIME,X(23);
                    SIGNAL:  7:  DISPOSE;

;                   PART TYPE 5
```

```
TYP5            BRANCH,1:
                        IF, MR(DECOUPLER3E) .EQ.0,
WAITD3E:
                        ELSE, GETEFD3;
WAITD3E         QUEUE, HOLD3E;
                WAIT:  32,1;
GETEFD3         QUEUE, UNLOAD3E;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER3E, -1;
                SIGNAL:  10;
                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4E;
PROCESS4E       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL18;
                SEIZE:  OPERATOR;
                RELEASE:   OPERATOR;
                ASSIGN: X(4)=0;
                SIGNAL: 8;
                ALTER:  DECOUPLER4E, +1;
                ASSIGN:X(42)=TNOW-X(19);
                TALLY:DECOUPLER4E REPL. TIME,X(42);
                SIGNAL: 30:  DISPOSE;

;               PART TYPE 6

TYP6            BRANCH,1:
                        IF, MR(DECOUPLER3F) .EQ.0,
WAITD3F:
                        ELSE, GETFFD3;
WAITD3F         QUEUE, HOLD3F;
                WAIT:  33,1;
GETFFD3         QUEUE, UNLOAD3F;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER3F, -1;
                SIGNAL:  10;
                BRANCH:
                        ALWAYS, LATHE:
                        ALWAYS, PROCESS4F;
PROCESS4F       ASSIGN:  X(4)=1;
                DELAY:  RN(PT4);
                QUEUE, UNL19;
                SEIZE:  OPERATOR;
                RELEASE:   OPERATOR;
                ASSIGN: X(4)=0;
                SIGNAL: 8;
                ALTER:  DECOUPLER4F, +1;
```

```
                ASSIGN:X(43)=TNOW-X(19);
                TALLY:DECOUPLER4F REPL. TIME,X(43);
                SIGNAL: 31:  DISPOSE;

;               AC3 (LATHE & DECOUPLER 2)

LATHE           QUEUE, AC3:  DETACH;
                CREATE;
BR5             BRANCH,1:
                        IF, NQ(AC3) .GT.0, NEXTORDER:
                        ELSE, WAIT3;
WAIT3           QUEUE, HOLDBLOCKL;
                WAIT:  10:  NEXT(BR5);
NEXTORDER       REMOVE:  1, AC3, BR6:  NEXT (BR5);
BR6             BRANCH,1:
                        IF, X(3) .GT.0, BUSYLATHE:
                        ELSE, IDLELATHE;
BUSYLATHE       QUEUE, DELAYATL;
                WAIT:  14;
IDLELATHE       BRANCH,1:
                        IF, PARTTYPE==1, TY1:
                        IF, PARTTYPE==2, TY2:
                        IF, PARTTYPE==3, TY3:
                        IF, PARTTYPE==4, TY4:
                        IF, PARTTYPE==5, TY5:
                        ELSE, TY6;

;               PART TYPE 1

TY1             BRANCH,1:
                        IF, MR(DECOUPLER2A) .EQ.0,
WAITD2A:
                        ELSE, GETAFD2;
WAITD2A         QUEUE, HOLD2A;
                WAIT:  15,1;
GETAFD2         QUEUE, UNLOAD2A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2A, -1;
                SIGNAL:  19;
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3A;
PROCESS3A       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL9;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14;
```

```
                ALTER:  DECOUPLER3A, +1;
                SIGNAL:  9:  DISPOSE;

;               PART TYPE 2

TY2             BRANCH,1:
                        IF, MR(DECOUPLER2B) .EQ.0,
WAITD2B:
                        ELSE, GETBFD2;
WAITD2B         QUEUE, HOLD2B;
                WAIT:  16,1;
GETBFD2         QUEUE, UNLOAD2B;
                SEIZE:  OPERATOR;
                DELAY: RN( WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2B, -1;
                SIGNAL:  19;
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3B;
PROCESS3B       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL10;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14;
                ALTER:  DECOUPLER3B, +1;
                SIGNAL:  11:  DISPOSE;

;               PART TYPE 3

TY3             BRANCH,1:
                        IF, MR(DECOUPLER2C) .EQ.0,
WAITD2C:
                        ELSE, GETCFD2;
WAITD2C         QUEUE, HOLD2C;
                WAIT:  17,1;
GETCFD2         QUEUE, UNLOAD2C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2C, -1;
                SIGNAL:  19;
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3C;
PROCESS3C       ASSIGN:  X(3)=1;
                DELAY:  RN(PT3);
                QUEUE, UNL11;
                SEIZE:  OPERATOR;
```

```
                    DELAY:  RN(UNLOADT);
                    RELEASE:  OPERATOR;
                    ASSIGN:  X(3)=0;
                    SIGNAL:  14;
                    ALTER:  DECOUPLER3C, +1;
                    SIGNAL:  12:  DISPOSE;

;                   PART TYPE 4

TY4                 BRANCH,1:
                            IF, MR(DECOUPLER2D) .EQ.0,
WAITD2D:
                            ELSE, GETDFD2;
WAITD2D             QUEUE, HOLD2D;
                    WAIT:  18,1;
GETDFD2             QUEUE, UNLOAD2D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER2D, -1;
                    SIGNAL:  19;
                    BRANCH:
                            ALWAYS, DRILL:
                            ALWAYS, PROCESS3D;
PROCESS3D           ASSIGN:  X(3)=1;
                    DELAY:RN(PT3);
                    QUEUE, OPERATION3D;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(3)=0;
                    SIGNAL:  14;
                    ALTER:  DECOUPLER3D, +1;
                    SIGNAL:  13:  DISPOSE;

;                   PART TYPE 5

TY5                 BRANCH,1:
                            IF, MR(DECOUPLER2E) .EQ.0,
WAITD2E:
                            ELSE, GETEFD2;
WAITD2E             QUEUE, HOLD2E;
                    WAIT:  34,1;
GETEFD2             QUEUE, UNLOAD2E;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER2E, -1;
                    SIGNAL:  19;
                    BRANCH:
                            ALWAYS, DRILL:
                            ALWAYS, PROCESS3E;
```

```
PROCESS3E       ASSIGN:  X(3)=1;
                DELAY:RN(PT3);
                QUEUE, OPERATION3E;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14;
                ALTER:  DECOUPLER3E, +1;
                SIGNAL:  32:  DISPOSE;

;               PART TYPE 6

TY6             BRANCH,1:
                        IF, MR(DECOUPLER2F) .EQ.0,
WAITD2F:
                        ELSE, GETFFD2;
WAITD2F         QUEUE, HOLD2F;
                WAIT:  35,1;
GETFFD2         QUEUE, UNLOAD2F;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER2F, -1;
                SIGNAL:  19;
                BRANCH:
                        ALWAYS, DRILL:
                        ALWAYS, PROCESS3F;
PROCESS3F       ASSIGN:  X(3)=1;
                DELAY:RN(PT3);
                QUEUE, OPERATION3F;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(3)=0;
                SIGNAL:  14;
                ALTER:  DECOUPLER3F, +1;
                SIGNAL:  33:  DISPOSE;

;               AC2 (DRILL & DECOUPLER 1)

DRILL           QUEUE, AC2:  DETACH;
                CREATE;
BR7             BRANCH,1:
                        IF, NQ(AC2) .GT.0, NEXTENTITY:
                        ELSE, WAIT4;
WAIT4           QUEUE, HOLDBLOCKD;
                WAIT:  19,1:  NEXT(BR7);
NEXTENTITY      REMOVE:  1,AC2, BR8:  NEXT(BR7);
BR8             BRANCH,1:
                        IF, X(2) .GT.0, BUSYDRILL:
                        ELSE, IDLEDRILL;
```

```
BUSYDRILL       QUEUE, DELAYATD;
                WAIT:  20,1;
IDLEDRILL       BRANCH,1:
                        IF, PARTTYPE==1, PART1:
                        IF, PARTTYPE==2, PART2:
                        IF, PARTTYPE==3, PART3:
                        IF, PARTTYPE==4, PART4:
                        IF, PARTTYPE==5, PART5:
                        ELSE, PART6;
PART1           BRANCH,1:
                        IF, MR(DECOUPLER1A) .EQ.0,
WAITD1A:
                        ELSE, GETAFD1;
WAITD1A         QUEUE, HOLD1A;
                WAIT:  21,1;
GETAFD1         QUEUE, UNLOAD1A;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1A, -1;
                SIGNAL:  25;
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2A;
PROCESS2A       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL12;
                SEIZE:  OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20;
                ALTER:  DECOUPLER2A, +1;
                SIGNAL:  15:  DISPOSE;

;               PART TYPE 2

PART2           BRANCH,1:
                        IF, MR(DECOUPLER1B) .EQ.0,
WAITD1B:
                        ELSE, GETBFD1;
WAITD1B         QUEUE, HOLD1B;
                WAIT:  22,1;
GETBFD1         QUEUE, UNLOAD1B;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1B, -1;
                SIGNAL:  25;
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2B;
```

```
PROCESS2B       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL13;
                SEIZE:  OPERATOR;
                DELAY:RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20;
                ALTER:  DECOUPLER2B, +1;
                SIGNAL:  16:  DISPOSE;

;               PART TYPE 3

PART3           BRANCH,1:
                        IF, MR(DECOUPLER1C) .EQ.0,
WAITD1C:
                        ELSE, GETCFD1;
WAITD1C         QUEUE, HOLD1C;
                WAIT:  23,1;
GETCFD1         QUEUE, UNLOAD1C;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1C, -1;
                SIGNAL:  25;
                BRANCH:
                        ALWAYS, SPINDLE:
                        ALWAYS, PROCESS2C;
PROCESS2C       ASSIGN:  X(2)=1;
                DELAY:  RN(PT2);
                QUEUE, UNL14;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE: OPERATOR;
                ASSIGN:  X(2)=0;
                SIGNAL:  20;
                ALTER:  DECOUPLER2C, +1;
                SIGNAL:  17:  DISPOSE;

;               PART TYPE 4

PART4           BRANCH,1:
                        IF, MR(DECOUPLER1D) .EQ.0,
WAITD1D:
                        ELSE, GETDFD1;
WAITD1D         QUEUE, HOLD1D;
                WAIT:  24,1;
GETDFD1         QUEUE, UNLOAD1D;
                SEIZE:  OPERATOR;
                DELAY:  RN(WALKT);
                RELEASE:  OPERATOR;
                ALTER:  DECOUPLER1D, -1;
```

```
                    SIGNAL:  25;
                    BRANCH:
                             ALWAYS, SPINDLE:
                             ALWAYS, PROCESS2D;
PROCESS2D           ASSIGN:  X(2)=1;
                    DELAY:  RN(PT2);
                    QUEUE, UNL15;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(2)=0;
                    SIGNAL:  20;
                    ALTER:  DECOUPLER2D, +1;
                    SIGNAL:  18:  DISPOSE;

;                   PART TYPE 5

PART5               BRANCH,1:
                             IF, MR(DECOUPLER1E) .EQ.0,
WAITD1E:
                             ELSE, GETEFD1;
WAITD1E             QUEUE, HOLD1E;
                    WAIT:  36,1;
GETEFD1             QUEUE, UNLOAD1E;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER1E, -1;
                    SIGNAL:  25;
                    BRANCH:
                             ALWAYS, SPINDLE:
                             ALWAYS, PROCESS2E;
PROCESS2E           ASSIGN:  X(2)=1;
                    DELAY:  RN(PT2);
                    QUEUE, UNL20;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(2)=0;
                    SIGNAL:  20;
                    ALTER:  DECOUPLER2E, +1;
                    SIGNAL:  34:  DISPOSE;

;                   PART TYPE 6

PART6               BRANCH,1:
                             IF, MR(DECOUPLER1F) .EQ.0,
WAITD1F:
                             ELSE, GETFFD1;
WAITD1F             QUEUE, HOLD1F;
                    WAIT:  37,1;
GETFFD1             QUEUE, UNLOAD1F;
```

```
                    SEIZE:  OPERATOR;
                    DELAY:  RN(WALKT);
                    RELEASE:  OPERATOR;
                    ALTER:  DECOUPLER1F, -1;
                    SIGNAL:  25;
                    BRANCH:
                            ALWAYS, SPINDLE:
                            ALWAYS, PROCESS2F;
PROCESS2F           ASSIGN:  X(2)=1;
                    DELAY:  RN(PT2);
                    QUEUE, UNL21;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(UNLOADT);
                    RELEASE: OPERATOR;
                    ASSIGN:  X(2)=0;
                    SIGNAL:  20;
                    ALTER:  DECOUPLER2F, +1;
                    SIGNAL:  35:  DISPOSE;


;                   AC1 (SPINDLE MACHINE)

SPINDLE             QUEUE, AC1:  DETACH;
                    CREATE;
BR9                 BRANCH,1:
                            IF, NQ(AC1) .GT.0, M1:
                            ELSE, WAIT5;
WAIT5               QUEUE, HOLDBLOCKS;
                    WAIT:  25,1:  NEXT (BR9);
M1                  REMOVE:  1, AC1, BR10:  NEXT (BR9);
BR10                BRANCH,1:
                            IF, X(1) .GT.0, BUSYSPINDLE:
                            ELSE, IDLESPINDLE;
BUSYSPINDLE         QUEUE, DELAYATS;
                    WAIT:  26,1;
IDLESPINDLE         BRANCH,1:
                            IF, PARTTYPE==1, PT1:
                            IF, PARTTYPE==2, PT2:
                            IF, PARTTYPE==3, PT3:
                            IF, PARTTYPE==4, PT4:
                            IF, PARTTYPE==5, PT5:
                            ELSE, PT6;

PT1                 QUEUE, CARTA;
                    SEIZE:  OPERATOR;
                    DELAY:  RN(PAPART);
                    ASSIGN:  X(1)=1;
                    DELAY:  LOADT; LOAD/POSITION/START
MACHINE
                    RELEASE:  OPERATOR;
                    DELAY:  RN(PT1);
                    QUEUE, ECYCLEA;
```

```
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(1)=0;
                SIGNAL:  26;
                ALTER:  DECOUPLER1A, +1;
                SIGNAL:  21:DISPOSE;

PT2             QUEUE, CARTB;
                SEIZE:  OPERATOR;
                DELAY:  RN(PAPART);
                ASSIGN:  X(1)=1;
                DELAY: RN(LOADT);
                RELEASE:  OPERATOR;
                DELAY:  RN(PT1);
                QUEUE, ECYCLEB;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(1)=0;
                SIGNAL:  26;
                ALTER:  DECOUPLER1B, +1;
                SIGNAL:  22:DISPOSE;

PT3             QUEUE, CARTC;
                SEIZE:  OPERATOR;
                DELAY:  RN(PAPART);
                ASSIGN:  X(1)=1;
                DELAY:  RN(LOADT);
                RELEASE:  OPERATOR;
                DELAY:  RN(PT1);
                QUEUE, ECYCLEC;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(1)=0;
                SIGNAL:  26;
                ALTER:  DECOUPLER1C, +1;
                SIGNAL:  23:DISPOSE;

PT4             QUEUE, CARTD;
                SEIZE:  OPERATOR;
                DELAY:  RN(PAPART);
                ASSIGN:  X(1)=1;
                DELAY:  RN(LOADT);
                RELEASE:  OPERATOR;
                DELAY:  RN(PT1);
                QUEUE, ECYCLED;
                SEIZE:  OPERATOR;
                DELAY:  RN(UNLOADT);
                RELEASE:  OPERATOR;
                ASSIGN:  X(1)=0;
```

```
                    SIGNAL:   26;
                    ALTER:   DECOUPLER1D, +1;
                    SIGNAL:   24:DISPOSE;

PT5                 QUEUE, CARTE;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(PAPART);
                    ASSIGN:   X(1)=1;
                    DELAY:   RN(LOADT);
                    RELEASE:   OPERATOR;
                    DELAY:   RN(PT1);
                    QUEUE, ECYCLEE;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(UNLOADT);
                    RELEASE:   OPERATOR;
                    ASSIGN:   X(1)=0;
                    SIGNAL:   26;
                    ALTER:   DECOUPLER1E, +1;
                    SIGNAL:   36:DISPOSE;

PT6                 QUEUE, CARTF;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(PAPART);
                    ASSIGN:   X(1)=1;
                    DELAY:   RN(LOADT);
                    RELEASE:   OPERATOR;
                    DELAY:   RN(PT1);
                    QUEUE, ECYCLEF;
                    SEIZE:   OPERATOR;
                    DELAY:   RN(UNLOADT);
                    RELEASE:   OPERATOR;
                    ASSIGN:   X(1)=0;
                    SIGNAL:   26;
                    ALTER:   DECOUPLER1F, +1;
                    SIGNAL:   37:DISPOSE;
END;
```

```
BEGIN;

PROJECT, CELLULAR MANUFACTURING, N ALTENEH;

PARAMETERS:     1,ORDERIN,8     :! TIME BETWEEN ARRIVALS
                2,WALKT,.5,.1   :! WALKING TIME
                3,INSPT,1,.1    :! INSPECTION TIME
                4,UNLOADT,.5,.1:! UNLOADING MACHINES
                5,PT5,3.5,.45   :! PROCESSING TIMES
                6,PT4,3.5,.45   :
                7,PT3,3.5,.45   :
                8,PT2,3.5,.45   :
                9,PT1,3.5,.45   :
                10,PAPART,.5,.1:! PICK A PART
                11,LOADT,.8,.1  :! LOADING TIME
                17,SCRAPT,2,4   ;! SCRAP TIME

ATTRIBUTES:     INSP5TA :         INSP5TB :
                INSP5TC :         INSP5TD :
                INSP5TE :         INSP5TF :
                REWORKTA:         REWORKTB:
                REWORKTC:         REWORKTD:
                REWORKTE:         REWORKTF:
                PARTTYPE                  ;

QUEUES:         UNLOAD4A:         UNLOAD4B:
                UNLOAD4C:         UNLOAD4D:
                UNLOAD4E:         UNLOAD4F:
                UNLOAD3A:         UNLOAD3B:
                UNLOAD3C:         UNLOAD3D:
                UNLOAD3E:         UNLOAD3F:
                UNLOAD2A:         UNLOAD2B:
                UNLOAD2C:         UNLOAD2D:
                UNLOAD2E:         UNLOAD2F:
                UNLOAD1A:         UNLOAD1B:
                UNLOAD1C:         UNLOAD1D:
                UNLOAD1E:         UNLOAD1F:
                REWORK5A:         REWORK5B:
                REWORK5C:         REWORK5D:
                REWORK5E:         REWORK5F:
                DELAYATV:         DELAYATH:
                DELAYATL:         DELAYATD:
                DELAYATS:         HOLDAT5 :
                HOLD4A  :         HOLD4B  :
                HOLD4C  :         HOLD4D  :
                HOLD4E  :         HOLD4F  :
                HOLD3A  :         HOLD3B  :
                HOLD3C  :         HOLD3D  :
                HOLD3E  :         HOLD3F  :
                HOLD2A  :         HOLD2B  :
                HOLD2C  :         HOLD2D  :
                HOLD2E  :         HOLD2F  :
```

```
                HOLD1A  :          HOLD1B  :
                HOLD1C  :          HOLD1D  :
                HOLD1E  :          HOLD1F  :
                ECYCLEA :          ECYCLEB :
                ECYCLEC :          ECYCLED :
                ECYCLEE :          ECYCLEF :
                CARTA   :          CARTB   :
                CARTC   :          CARTD   :
                CARTE   :          CARTF   :
                I5A:I5B:I5C:I5D:I5E:I5F  :
                AC4: AC3:  AC2:  AC1:UNL1:
                UNL2:  UNL3:  UNL4:UNL5  :
                UNL6:  UNL7:  UNL8:UNL21 :
                UNL9: UNL10: UNL11: UNL12:
                UNL13: UNL14: UNL15:UNL16:
                UNL17:UNL18:UNL19:UNL20  :
                ARRIVEBLOCK:    HOLDBLOCKS:
                HOLDBLOCKD:     HOLDBLOCKL:
                HOLDBLOCKH:    OPERATION3D:
                OPERATION3E:   OPERATION3F;

RESOURCES:      DECOUPLER1A,SCHED(1)  :
                DECOUPLER1B,SCHED(2)  :
                DECOUPLER1C,SCHED(3)  :
                DECOUPLER1D,SCHED(4)  :
                DECOUPLER1E,SCHED(5)  :
                DECOUPLER1F,SCHED(6)  :
                DECOUPLER2A,SCHED(7)  :
                DECOUPLER2B,SCHED(8)  :
                DECOUPLER2C,SCHED(9)  :
                DECOUPLER2D,SCHED(10):
                DECOUPLER2E,SCHED(11):
                DECOUPLER2F,SCHED(12):
                DECOUPLER3A,SCHED(13):
                DECOUPLER3B,SCHED(14):
                DECOUPLER3C,SCHED(15):
                DECOUPLER3D,SCHED(16):
                DECOUPLER3E,SCHED(17):
                DECOUPLER3F,SCHED(18):
                DECOUPLER4A,SCHED(19):
                DECOUPLER4B,SCHED(20):
                DECOUPLER4C,SCHED(21):
                DECOUPLER4D,SCHED(22):
                DECOUPLER4E,SCHED(23):
                DECOUPLER4F,SCHED(24):
                DECOUPLER5A,SCHED(25):
                DECOUPLER5B,SCHED(26):
                DECOUPLER5C,SCHED(27):
                DECOUPLER5D,SCHED(28):
                DECOUPLER5E,SCHED(29):
                DECOUPLER5F,SCHED(30):
                OPERATOR:       INSP5;
```

```
SCHEDULES:      1,1 :   2,1 :   3,2 :   4,1 :  5,1 :
                6,1 :   7,1 :   8,1 :   9,1 :  10,1:
                11,1:  12,1 :  13,1 :  14,1 :  15,1:
                16,1:  17,1 :  18,1 :  19,1 :  20,1:
                21,1:  22,1 :  23,1 :  24,1 :  25,1:
                26,1:  27,1 :  28,1 :  29,1 :  30,1;

DSTAT:          NR(OPERATOR) * 100, OPERATOR UTIL:
                X(1)  * 100, SPINDLE UTIL:
                X(2)  * 100, DRILL UTIL  :
                X(3)  * 100, LATHE UTIL  :
                X(4) * 100, HMILL UTIL   :
                X(5) * 100, VMILL UTIL   :
                (X(1) + X(2) + X(3)      +
                X(4) + X(5))* 100/5, CELL UTIL :
                NQ(DELAYATV), NO. DELAYED AT M5:
                NQ(DELAYATH), NO. DELAYED AT M4:
                NQ(DELAYATL), NO. DELAYED AT M3:
                NQ(DELAYATD), NO. DELAYED AT M2:
                NQ(DELAYATS), NO. DELAYED AT M1:
                MR(DECOUPLER4A), NO. IN D4A    :
                MR(DECOUPLER4B), NO. IN D4B    :
                MR(DECOUPLER4C), NO. IN D4C    :
                MR(DECOUPLER4D), NO. IN D4D    :
                MR(DECOUPLER4E), NO. IN D4E    :
                MR(DECOUPLER4F), NO. IN D4F    :
                MR(DECOUPLER3A), NO. IN D3A    :
                MR(DECOUPLER3B), NO. IN D3B    :
                MR(DECOUPLER3C), NO. IN D3C    :
                MR(DECOUPLER3D), NO. IN D3D    :
                MR(DECOUPLER3E), NO. IN D3E    :
                MR(DECOUPLER3F), NO. IN D3F    :
                MR(DECOUPLER2A), NO. IN D2A    :
                MR(DECOUPLER2B), NO. IN D2B    :
                MR(DECOUPLER2C), NO. IN D2C    :
                MR(DECOUPLER2D), NO. IN D2D    :
                MR(DECOUPLER2E), NO. IN D2E    :
                MR(DECOUPLER2F), NO. IN D2F    :
                MR(DECOUPLER1A), NO. IN D1A    :
                MR(DECOUPLER1B), NO. IN D1B    :
                MR(DECOUPLER1C), NO. IN D1C    :
                MR(DECOUPLER1D), NO. IN D1D    :
                MR(DECOUPLER1E), NO. IN D1E    :
                MR(DECOUPLER1F), NO. IN D1F    ;

COUNTERS:       NO. OF REQUESTS,,    NO:
                A PARTS REWORKED,,   NO:
                A PARTS SCRAPED,,    NO:
                B PARTS REWORKED,,   NO:
                B PARTS SCRAPED,,    NO:
                C PARTS REWORKED,,   NO:
```

```
                    C PARTS SCRAPED,,     NO:
                    D PARTS REWORKED,,    NO:
                    D PARTS SCRAPED,,     NO:
                    E PARTS REWORKED,,    NO:
                    E PARTS SCRAPED,,     NO:
                    F PARTS REWORKED,,    NO:
                    F PARTS SCRAPED,,     NO;

TALLIES:            A CYCLE TIME :
                    A INSPECTION TIME:
                    A REWORK TIME:
                    A SCRAP TIME :
                    B CYCLE TIME :
                    B INSPECTION TIME:
                    B REWORK TIME:
                    B SCRAP TIME :
                    C CYCLE TIME :
                    C INSPECTION TIME:
                    C REWORK TIME:
                    C SCRAP TIME :
                    D CYCLE TIME :
                    D INSPECTION TIME:
                    D REWORK TIME:
                    D SCRAP TIME :
                    E INSPECTION TIME:
                    E REWORK TIME:
                    E SCRAP TIME :
                    F INSPECTION TIME:
                    F REWORK TIME:
                    F SCRAP TIME :
                    DECOUPLER4A REPL. TIME:
                    DECOUPLER4B REPL. TIME:
                    DECOUPLER4C REPL. TIME:
                    DECOUPLER4D REPL. TIME:
                    DECOUPLER4E REPL. TIME:
                    DECOUPLER4F REPL. TIME;

REPLICATE,,0,480;

END;
```

VITA

Nayef Khalil Alteneh was born in Amman, Jordan on May 23, 1965. He attended Amman High School in Amman, Jordan and graduated in 1983. He attended the University of Tennessee at Knoxville and received his Bachelor of Science degree in Industrial Engineering in May of 1993. The following August, he reentered the University of Tennessee at Knoxville and in December, 1994 received the degree of Master of Science in Industrial Engineering degree.