SELECTION OF AN OPTIMAL SET OF ASSEMBLY PART DELIVERY DATES IN A STOCHASTIC ASSEMBLY SYSTEM

by

Sanchoy K. Das

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Industrial Engineering and Operations Research

APPROVED:

S.C.Sarin, Chairman

C.J.Malmborg

H.D.Sherali

June, 1985

Blacksburg, Virginia

SELECTION OF AN OPTIMAL SET OF ASSEMBLY PART DELIVERY DATES IN A STOCHASTIC ASSEMBLY SYSTEM

by

Sanchoy K. Das

(ABSTRACT)

The scheduling of material requirements at a factory to maximize profits or productivity is a difficult mathematical problem. The stochastic nature of most production setups introduces additional complications as a result of the uncertainity involved in vendor reliability and processing times. But in developing the descriptive model for a system, a true representation can only be attained if the variability of these elements is considered.

Here we present the development of a normative model based on a new type of descriptive model which considers the element of stochasticity. The arrival time of an assembly part from a vendor is considered to be a normally distributed random variable. We attempt to optimize the system with regard to work-in-process inventory using a dynamic programming algorithm in combination with a heuristic procedure. The decision variable is the prescribed assembly part delivery date. The model is particularly suitable for application in low volume assembly lines, where products are manufactured in discrete batches.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my adviser Dr. Subhash C. Sarin, for his valuable support, patience, advice. and enthusiasm throughout the course of this study. I also thank Dr. Hanif D. Sherali and Dr. Charles J. Malmborg for serving on my graduate committee.

I would also like to express my thanks to my many friends, for their inspiration and support at all times.

Finally my sincere gratitude to my parents for their continous support and guidance.

TABLE OF CONTENTS

Appendix Page

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

1.1 INTRODUCTION TO THE PROBLEM

A typical characteristic of many industries is that they always have a succession of subcontractors. In these industries the final assembly parts are not only fabricated inside the factories but also made by many outside subcontractors. Further, the assembly parts themselves are in turn composed of many component parts made by sub-subcontractors. In general many sources will be involved in this complex multi-stage assembly system. The stages being either in a series arrangement or in a converging tree arrangement.

Industrial engineers are concerned with the problem of managing manufacturing operations in these complex systems. Our objective is to optimize the system with regard to certain performance measures. Clearly our continued thrust towards the introduction of robots and other automated processes will improve the efficiency of a factory. But the most significant effect on the output of a factory and it's product economy comes from other activities. By other activities we mean the development of a master schedule detailing in time phased format the requirements for finished products, assembly components and raw materials among other things.

The scheduling of these resources to maximize profits or productivity is a difficult mathematical problem. Various

techniques in operations research may be applied to obtain an optimum production plan in various situations. A method of planning would cut down on long waiting lines and hence inventory costs. Inventory is a non performing asset that is susceptible to damage, theft or obsolescence. Therefore reduction of inventory costs will be the major criterion in any policy to optimize production costs. In order to analyze a multi-stage production-assembly system, it would be necessary to build large and complex mathamatical models embracing the salient features of the system.

New technology and modern computation techniques provide us with several ways to model the behaviour of a production system. Probably the most widely used method is to simulate the production system to determine how such operating factors as inventory levels, downtime, variable processing times and lead time effect the production rate and the efficiency of a system. Computer simulation is a very powerful and versatile tool in the analysis of technical systems, and for determing whether a system can be improved. Models are usually simple to construct and may be employed to analyze different situations. But the simulation models are very costly to construct and validate. In general a different program must be constructed for each separate system. Mize (1979) and Shannon (1979) review the fundamental concepts of simulation modelling. According to Emshoff (1970) in a simulation experiment with a large number of continous variables only half the

battle is over with model development, since the search for the optimal result can be tedious. The running of the simulation program, once constructed, can involve a great deal of computer time, which could be uneconomical. Special purpose simulation languages have helped to reduce this factor, but there is still a formidable disadvantage. Three such languages GPSS, SLAM and SIMAN are reviewed by Davis (1984).

A simulation model that accurately describes the physical system enables us to analyse and evaluate different operating strategies. But simulation experiments that are undertaken to investigate the model as an interrelated system yield no additional information about the validity of the model. Also the determination of the combination of factor levels that provides the best overall response poses a difficult problem.

In many cases the performance of a system is evaluated through a quantitative model which is a mathematical representation of the system under study. Often this model involves an equation or equations which vary in their degree of complexity, with the degree of complexity represented by the model. At times, in attempting to devolop a mathematical model for a given system, we find the system so complex that the description of the system by a mathamatical model is beyond our capability or the system is amenable to description mathamatically, but correct analysis of the model is beyond the level of our mathamatical sophistication. But when the

system is amenable to both description and analysis by a math model, this may be the best method for decision making.

1.2 LOW VOLUME MULTI-STAGE ASSEMBLY SYSTEM

The development of a descriptive model and it's subsequent mathematical analysis provides an efficient and reliable optimization procedure. Wilhelm and Ahmadi-Marandi (1982) have developed and tested a methodology to describe the operating characteristics of a particular type of multi-stage assembly system. They investigated the problem of scheduling a low volume (small lot) assembly line to manufacture large and costly products. Examples of products manufactured in an assembly line of this nature are, aircrafts, sophisticated weapons systems, electrical-power and heavy agricultural machinery.

The operation of large product assembly lines is complex. It's operating characteristics are different from those of the high volume automatic assembly line. Long and greatly varying processing times make it impossible to balance such a line. Usually a significant number of component parts are supplied by external vendors. Hence, the intervals at which mainframes are launched into production, and rules by which delivery dates for assembly parts are set, have a significant influence on system performance. A good master schedule would therefore be required to coordinate vendor programs, customer due date requirements and production capacity.

In such systems the delivery of assembly parts is almost always uncertain, consequently variables that describe system performance are random. It is possible for a queue of assembly parts to form at any station due to these variations, and the schedule of delivery dates for the parts becomes an important factor in determining system performance. Hence to truly capture the system behavior it would be necessary to use a stochastic model of the assembly system.

In an actual manufacturing envoirment , the risks related to earliness or tardiness present many problems in managing assembly systems on a day to day basis. These risks are non-existent in the deterministic case. But master schedules based on a deterministic analysis appear to be overly optimistic, the probalistic nature of the system making them so. The cumulative effects of processing time-random variations tends to invalidate planned schedules over long periods of time, even if the schedule is planned using the correctly assesed expected job processing times. This is due to the long sojurn times between succesive returns to zero of a random walk.

Figure (1.1) depicts an assembly system of this type. The system operates according to the following scenario.

1. A schedule for producing J end products of the same type is considered.

2. After being launched (released) into production, each mainframe progresses through a series of N stations and finally into a finished product inventory from which it is shipped to satisfy customer orders.
3. A different part (or subassembly) is assembled to the mainframe at each station, and the resulting subassembly is transferred to the next station with zero delay.
4. Each station is a single server queue that provides unrestricted queueing space for both parts and subassemblies.

The model is easily extended to a converging tree type of line, where some assembly parts are supplied by branch lines. Such a scheme is shown in Figure (1.2).

Typically in such a system we would apply a material requirements planning (MRP) technique. But in a slow moving line the effect of uncertainity makes it difficult to estimate the part lead time reliably. Using the above model it is possible to minimize the operating cost of the system. The operating cost is in turn a function of the inventory level of assembly parts. Clearly the inventory in such a system will be measured in discrete units. We are thus interested in extending the concept of just-in-time (JIT), so successfully applied in mass production, to attain what may be called a just-in-time requirements planning system.

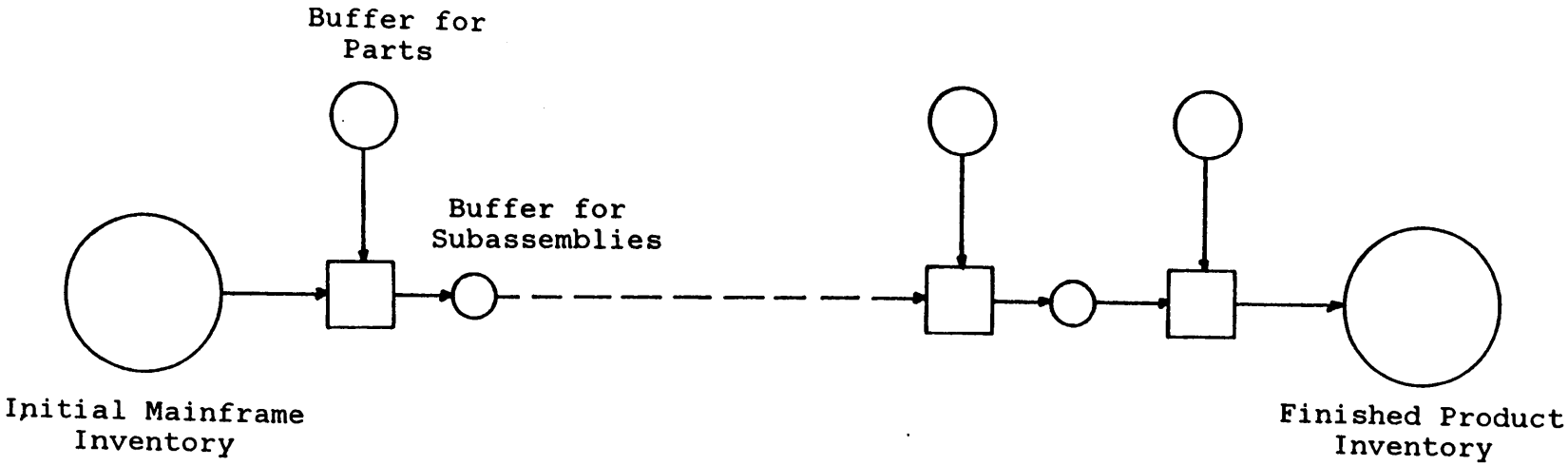


Fig. 1.1 Unbalanced Multi-Stage Assembly Line.

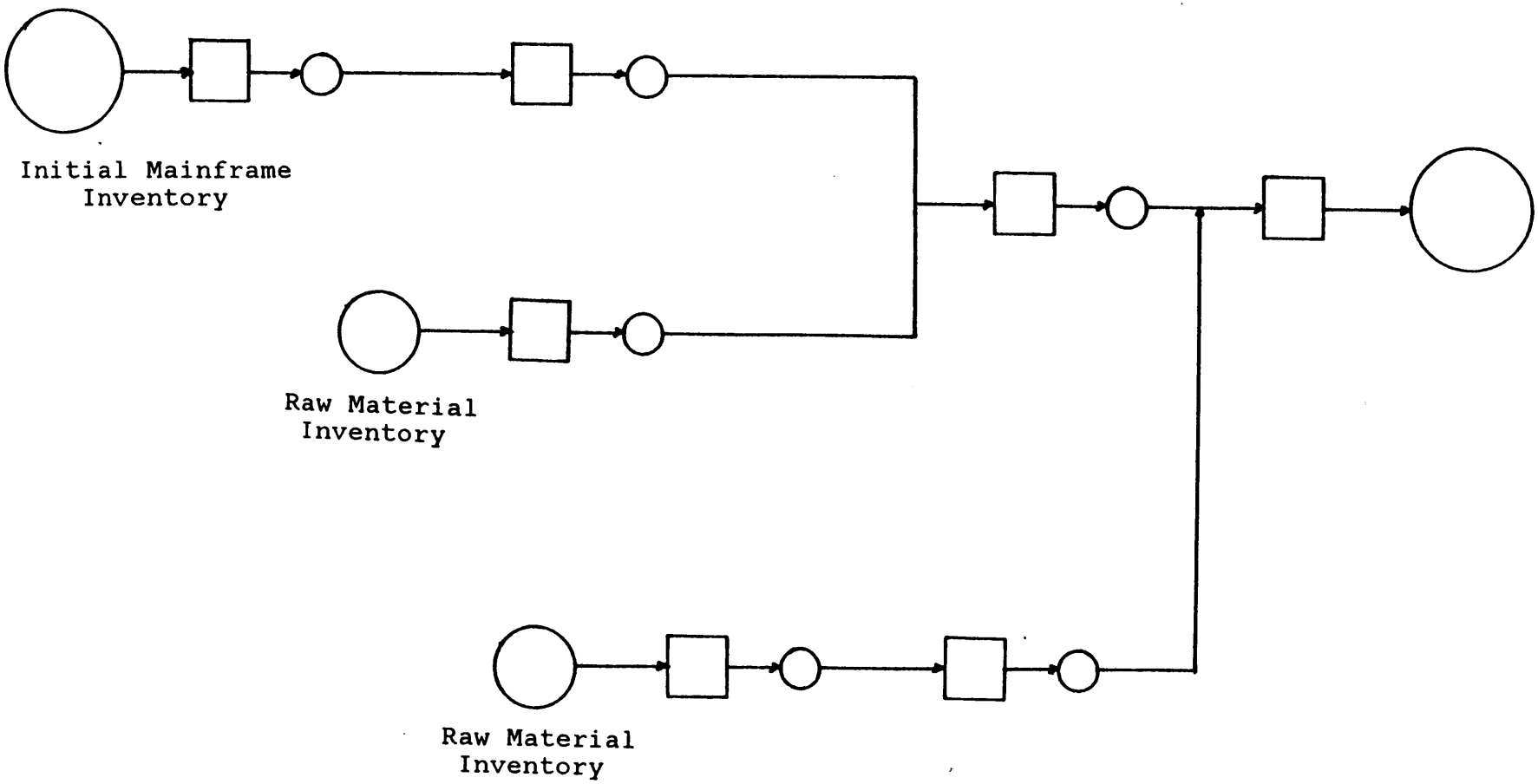


Fig. 1.2 Converging Tree Assembly System.

1.3 DESCRIPTION OF TECHNICAL TERMS

During the study various terms will be used to describe the operating characteristics and physical features of the model. To avoid confusion the following is a list of definitions of some of the important terms which will be used in the study.

- Mainframe - The initial frame which is launched into the assembly line.
- Assembly Parts - The part to be assembled to the mainframe at a workstation. Normally supplied by an external vendor.
- Part Delivery Date - The prescribed date on which an external vendor is supposed to deliver an assembly part.
- Sub-Assembly - The incomplete final product as it moves through the assembly stages.

1.4 PROBLEM STATEMENT

In a manfacturing assembly line, a variety of parts have to be provided at the various workstations. These parts are normally supplied by vendors or an in-house facility. Scheduling the arrival of these parts is the concern of industrial engineers. If vendors are completely reliable and a deterministic system can be assumed, then the solution is trivial. But in most situations, the part arrival time is a random variable. In such situations, scheduling the part ar-

rival is a difficult problem, since a delay in part arrival can have a significant impact on the system operating cost. Conversely, early arrival implies additional inventory cost.

The objective of this study is to develop a methodology which, would enable us to determine the optimal set of assembly parts delivery dates in a stochastic multi-stage assembly system. The part arrival time at each station, is assumed to be a normally distributed random variable, and the prescribed delivery date is the mean of this distribution. Hence, the decision variables in the prescriptive model, will be the mean delivery dates of the parts. The objective is to minimize the total cost of assembly part queue time, subassembly queue time and makespan.

1.5 THESIS ORGANIZATION

The thesis is sectioned into five chapters. Chapter two describes some of the literature on the scheduling of materials in an assembly system. In chapter three the development of the prescriptive model to be used for scheduling a single job multi-station problem is described. The descriptive model of Wilhelm and Ahmadi-Marandi (1982), which forms the basis of the model, is also presented. The solution methodolgy used is a heuristic version of a stochastic dynamic programming (DP) procedure using backward recursion. The heuristic nature arises due to a simplification of the derived state transformation function using non-linear regression analysis. The

DP solution was compared with other scheduling methods and shown to be superior.

In chapter four the DP solution is extended to the multi-job, multi-station case. Problems of dimensionality hindered the continued application of DP methods. Hence, a heuristic procedure was used in conjunction with the DP solution to determine the operating schedule. The heuristic procedure involved selecting paths, based on their criticality in the network, and then optimizing using the DP technique. The DP heuristic procedure was favorably tested on a wide range of randomly generated problems.

Chapter five summarizes the results of these investigations. The benefits of implementing the DP heuristic procedure are listed. Areas of future research, which not only improve the goodness of the procedure, but also extend its applicability to a larger variety of problems is also presented.

Chapter II

LITERATURE REVIEW

The study focuses on an area that has not been researched extensively. Search for pertinent literature revealed few papers on the subject. This is because of the difficult analysis required to model the complex problems caused by stochasticity. The assembly line we are studying is an obvious extension of the flow shop. Models describing the behavior of both these systems explain some of the operating characteristics of multistage systems.

2.1 ASSEMBLY LINES

This study is a departure from the traditional line balancing problem. Assembly line balancing techniques enable us to determine an optimal number of stations and the division of tasks among the stations, but they do not assist in developing an operating schedule for the line.

Harrison (1973) introduced a queueing theoretic model of an asembly line. His model consisted of a multi-input process of different components, and a single server who assembled the final product. Both the arrival of parts and assembly time were independently distributed random variables. Limit theorems were developed for the waiting time for the nth arriving item of type k. Harrison's research is limited

to one station but can be generalized into an assembly like network of sequential operations.

A new methodology to describe the operating characteristics of low volume (small lot) assembly systems was presented by Wilhelm and Ahmadi-Marandi (1982). They modelled the assembly line as a stochastic system in which the arrival of parts was a normally distributed random variable. They used a method developed by Clark (1961) to recursively estimate the mean and variance of operation start times at each station. Their methodology does provide estimates which are acceptable for master scheduling for production systems of the type we are studying. The work of Wilhelm and Ahmadi-Marandi thus form the basis of this study.

Wilhelm and Johnson (1983) extended the earlier descriptive model to incorporate parallel machines at a single station and to model operation times as random variables. The random operation time permits the system analyst to analyse the effect of breakdowns on system performance.

2.2 MULTI-STAGE FLOW SHOPS

Muth (1976) devoloped a unique modelling approach which may incorporate a variety of service time distributions. His paper also discussed inherent operating characteristics and provided a number of elementary models which yield considerable insight into flowline operation. Using this approach it

is possible to describe the relationship between line output rate and the service time distribution involved.

The imbalance of processing time in the assembly line under study causes in-process inventory to form. Buffer stocks maybe located between stations to reduce station dependency. Though in this study we assume there is an infinite amount of buffer space, buffer capacity can easily be introduced as a design parameter. Buzacott (1967) devolops models applicable to several configurations and describes how they might be used to specify optimal buffer stock levels. Kraemer and Love (1976) devoloped a model for optimizing the buffer inventory storage level in a sequence production system. It is a queuing type model but emphasizes the economics involved in flowline design. Altiok and Stidham (1983) also used a queueing model to analyse the allocation of buffer capacity in a flowshop type production line where stages are subject to breakdown.

Jensen and Khan (1972) considered the scheduling of a multi-stage production system with setup and inventory costs. They considered only one input of raw material at the first stage. All system variables were considered to be deterministic. Dynamic programming was used to obtain the startup and shut down schedule of each stage which minimizes the sum of inventory and startup costs. Each stage was operated in a periodic manner with a fixed cycle time. The time

between subsequent orders of raw materials was a decision variable.

The system of Jensen and Khan (1972) was extended by Bigham and Mogg (1981) for treatment of multi-stage scheduling with stochastic lead time. They considered lead time for raw materials to be a random variable with known probability density function. There was an input of raw material at each stage, but the raw material for all the stages in a single cycle was assumed to arrive simultaneously, which makes it equivalent to the former model. They assumed there would be no motivation for early startup even if raw material arrived early and all machines were available. This seems rational, but the very existence of such an enforced idle time reflects on the optimality of their schedule. The objective function of Jensen and Khan was extended to include the expected cost of both early and late deliveries of raw materials. Bigham and Mogg used an iterative procedure to determine the optimal lead time for raw materials. The applicability of their model is limited, but can probably be enriched.

2.3 MATERIAL REQUIREMENTS PLANNING (MRP)

MRP is rapidly becoming a widely used method of materials ordering, and has evolved into manufacturing resources planning or MRP II. From a given master schedule which shows the expected demand of parts, MRP will calculate the demand quantities for items dependent on the first item. In a MRP

II system one of the executing steps is purchasing and ordering. This step links the manufacturer with supplying vendors. Wight (1984) discusses in detail this relationship along with complete details of an MRP II system. Fox (1984) presents some future oriented ideas on MRP and it's relation with computer integrated manufacturing. Walker and Wysk (1983) present a framework by which organizations can identify the least cost purchased-part lead time strategy, for items in an MRP system. The methodology used is simulation, and primarily deals with job shops. The strategy recommended by them, prescribes a purchasing lead time of one standard deviation.

No earlier research considered the input of raw materials or assembly parts at more than one station. The successive arrival of subassemblies at a work station was only considered in the papers by Wilhelm (1982 & 1983). Project scheduling techniques may be used to model assembly line operations for a single product, but cannot be used for continous production.

Chapter III

DEVELOPMENT OF THE PRESCRIPTIVE MODEL

This study will be based on the descriptive model developed by Wilhelm and Ahmadi-Marandi (1982) and the extensions developed by Wilhelm and Johnson (1983). Their work forms the basis for describing the assembly line being studied. Here we present the descriptive model, and develop the prescriptive model and the methodology for it's implementation in the optimization process.

3.1 THE DESCRIPTIVE MODEL

The model is based on a fundamental relationship among the variables that collectively define the time at which an assembly operation can start. This model indicates that processing time of a subassembly at any station is limited by three random variables.

- The finish time of the previous subassembly at the station
- The finish time of the subassembly at the previous station
- The time at which the assembly part is available at the station

The start of the operation is determined by the maximum of these random variables. Therefore the operation time at any station is completely specified by the start and finish time.

This is shown schematically in figure (3.1). System behavior is defined by the following equations.

$$S_{ij} = \text{Max}\ \{F_{i,j-1},\ F_{i-1,j},\ A_{ij}\ \} \tag{3.1}$$

$$F_{ij} = S_{ij} + P_{ij} \tag{3.2}$$

where

i= 1,2,----,N is the index for assembly stations.

j= 1,2,----,J is the index for subassemblies.

S_{ij} is the start time of operation j at station i

F_{ij} is the finish time of operation j at station i

A_{ij} is the time at which assembly part j is available at station i

P_{ij} is the time to perform operation j at station i.

Equations (3.1) and (3.2) indicate the recursive relationship between assembly operations. This can be used as a model of a master schedule to coordinate the materials and resources required for production. It is possible to incorporate into the model variations indicative of actual operating conditions. These include arrival of a second part, machine breakdown, buffer capacity and the availability of materials handling equipment.

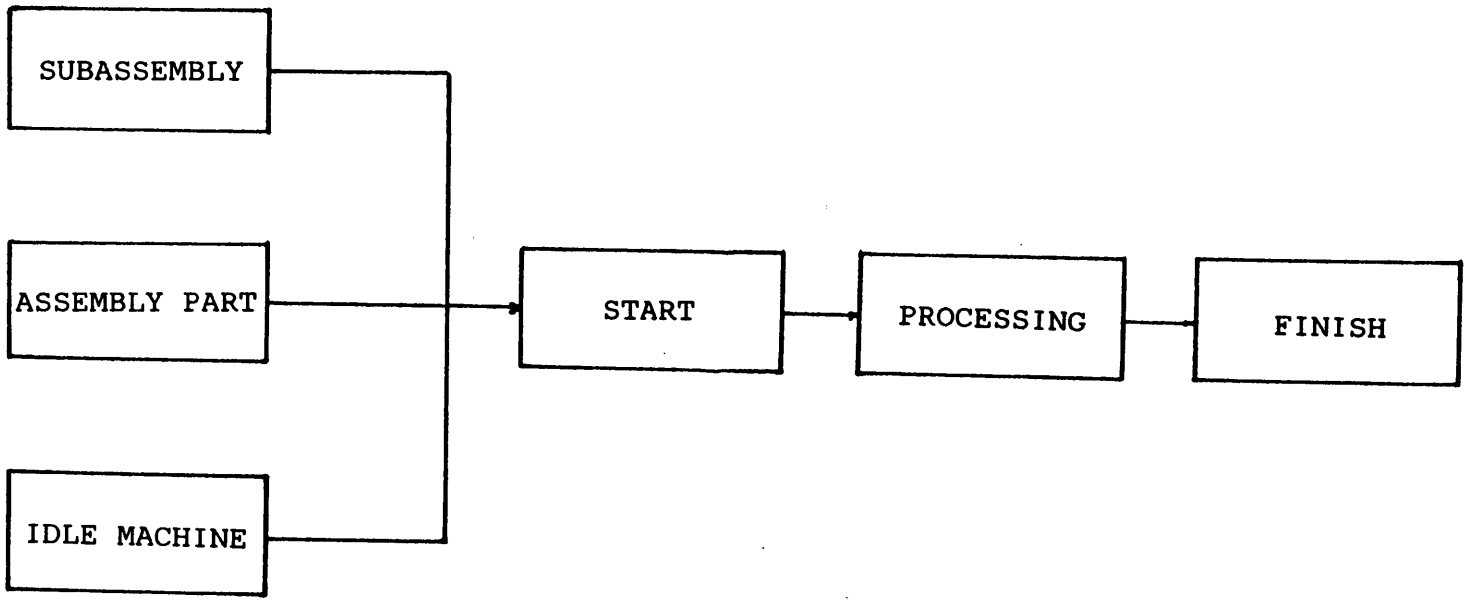


Fig. 3.1 Scheme of Operations at a Station.

3.1.1 Objective Function

We are interested in minimizing the total system idleness. There are three types of controllable idleness inherent in the system. The objective function thus consists of the following costs.

1. Expected total cost of workstation idle time:

$$\sum_{i=1}^{N} \sum_{j=1}^{J} W_i \{S_{ij} - F_{i,j-1}\}$$

2. Expected total cost of subassembly queue time:

$$\sum_{i=1}^{N} \sum_{j=1}^{J} M_i \{S_{ij} - F_{i-1,j}\}$$

3. Expected total cost of component part queue time:

$$\sum_{i=1}^{N} \sum_{j=1}^{J} C_i \{S_{ij} - A_{i,j}\}$$

Here, W_i is the unit time cost for an idle workstation, and C_i and M_i are the unit time queue costs for component parts and subassemblies respectively.

3.1.2 Estimation of Assembly Start Time

The above descriptive model is applicable to both the deterministic and stochastic cases. In this study we are primarily concerned with the stochastic case, and will

henceforth assume both part arrival time and processing time to be normally distributed independent random variables.

When equation (3.1) is applied recursively in the stochastic case, first processing the first mainframe through all the stages sequentially, and then the subsequent mainframes serially, it is possible to obtain estimates of the assembly start and finish times at each stage for each mainframe. Wilhelm and Ahmadi-Marandi (1982) used a method developed by Clark (1961) in the recursion to calculate estimates of the mean and variance of operation start times. Clark's method enables us to calculate the moments of the greatest of a finite set of random variables. Since the start times are inter-related by equation (3.1) a series of correlation coefficients relating $F_{i,j-1}$ and $F_{i-1,j}$, have to be determined at each recursion. To do this Wilhelm and Ahmadi-Marandi developed a 'seven-step procedure', which is summarized in Table (3.1). Clark's equations which form the basis of this procedure are presented below.

Let,

$$\tau = \text{Max}\ \{\eta,\ \nu\} \tag{3.3}$$

Where η and ν are normally distributed with means μ_1, μ_2 and with variances σ_1, σ_2 respectively. Then the first and second moments of τ are given by:

$$V_1 = \mu_1\Phi(\alpha) + \mu_2\Phi(-\alpha) + a\psi(\alpha) \tag{3.4}$$

$$V_2 = (\mu_1{}^2+\sigma_1{}^2)\phi(\alpha) + (\mu_2{}^2+\sigma_2{}^2)\phi(-\alpha) + (\mu_1+\mu_2)a\psi(\alpha) \tag{3.5}$$

where,

$a^2 = \sigma_1{}^2+\sigma_2{}^2-2\sigma_1\sigma_2\rho$

$\alpha = (\mu_1-\mu_2)/a$

$\psi(x) = (2\pi)^{1/2} \exp(-x^2/2)$

$\phi(t) = \int\psi(t)\, dt$

$r(x,y)$ = coefficient of linear correlation between x,y.

$\rho = r(\eta,\nu)$

Equation (3.3) is easily extended to any finite number of normal variables using the following logic.

$$\beta = \mathrm{Max}\{\eta,\nu,\varepsilon\} = \mathrm{Max}\{\mathrm{Max}(\eta,\nu),\varepsilon\} \tag{3.6}$$

$$r\{\mathrm{Max}(\nu,\eta),\varepsilon\} = [\sigma_1\rho_1\phi(\alpha)+\sigma_2\rho_2\phi(-\alpha)]/\sqrt{(V_1{}^2-V_2{}^2)} \tag{3.7}$$

where,

$\rho_1 = r(\nu,\varepsilon)$

$\rho_2 = r(\eta,\varepsilon)$

Estimates of the moments of β can be determined by using equations similar to (3.4) and (3.5). But this calculation would be inaccurate because the distribution of $\max(\eta,\nu)$ is not normal. Tippet (1925) showed it to be a positively skewed distribution. Clark (1961) compared the results of his approximation with those of Tippet, to get an error of less

than one percent. The error involved in the repeated application of equation (3.1) may be small, but will accumulate as the number of stations and subassemblies increase. Wilhelm and Johnson (1983) investigated the error of approximation introduced by the above computational procedure. They compared the estimates of makespan (F_{NJ}) obtained from the recursive procedure with that resulting from a simulation experiment. In their results, the null hypothesis that the true makespan is equal to the recursive estimate is rejected with (α =0.05) for J>50, but cannot be rejected for J<40. These results suggest, that the recursive estimates can be reliably used in scheduling a certain size of assembly lines.

The seven-step procedure requires a recursive use of equation (3.7) to calculate the coefficient of correlation between the variables in equation (3.1) which define the start time at a station. S_{ij} is dependent on all the start and finish times upstream of it, but the procedure considers the dependency only of the previous two stations and jobs.

STEP	T	I	N	P_1	P_2	P	RESULT
1	F(i-1, j-1)	F(i-1, j-1)	F(i,j-2)	$\tau\begin{bmatrix}F(i-1,\ j-1),\\ F(i-1,\ j-1)\end{bmatrix}$	$\tau\begin{bmatrix}F(i-1,\ j-1),\\ F(i,\ j-2)\end{bmatrix}$	$\tau\begin{bmatrix}F(i-1,\ j-1),\\ F(i,\ j-2)\end{bmatrix}$	P_2^{12}
2	F(i-1, j-1)	A(i,j-1)	$\max\begin{bmatrix}F(i-1,\ j-1),\\ F(i,\ j-2)\end{bmatrix}$	0	P_2^{12}	0	P_2^{25}
3	F(i-2,j)	F(i-1, j-1)	F(i, j-2)	$\tau\begin{bmatrix}F(i-2,\ j),\\ F(i-1,\ j-1)\end{bmatrix}$	$\tau\begin{bmatrix}F(i-2,\ j)\\ F(i,\ j-2)\end{bmatrix}$	$\tau\begin{bmatrix}F(i-1,\ j-1),\\ F(i,\ j-2)\end{bmatrix}$	P_2^{34}
4	F(i-2,j)	A(i,j-1)	$\max\begin{bmatrix}F(i-1,\ j-1),\\ F(i,\ j-2)\end{bmatrix}$	0	ROS P_2^{34}	0	P_1^{45}
5	S(i,j-1)	F(i-2,j)	F(i-1, j-1)	P_1^{45}	P_2^{25}	$\tau\begin{bmatrix}F(i-2,\ j),\\ F(i-1,\ j-1)\end{bmatrix}$	P_2^{56}
6	S(i,j-1)	A(i-1,j)	$\max\begin{bmatrix}F(i-2,\ j),\\ F(i-1,\ j-1)\end{bmatrix}$	0	P_2^{56}	0	P^{67}
7	A(i,j)	F(i-1,j)	F(i, j-1)	0	0	P^{67}	P

Table 3.1 Wilhelm's Seven Step Procedure.

3.2 DYNAMIC PROGRAMMING FORMULATION

Dynamic programming (DP) is a computational method that allows us to break up a complex problem into a sequence of easier subproblems by means of a recursive relation, which can be evaluated by stages.

Since our descriptive model is sequential in nature, and the seven-step procedure is also recursive, we can conveniently apply stochastic DP for the optimization of the assembly line. But DP algorithms become extremely complex when applied to multi-dimensional problems. To avoid the problem of dimensionality we shall for the present, limit the model to a single-job multi-station case, and later attempt to extend it to the multi-job case.

The decision to limit the present model to the single job case, greatly improves the computational aspect of our problem. Firstly, the necessity to apply the seven step procedure is avoided, since workstation availability is assumed not to limit operation start time. Further, it is possible to apply both a forward and backward recursion solution to the DP. Initial investigations with the forward recursion approach were not successful . Problems due to dimensionality and the inverse transformation function could not be overcome. Consequently, a backward approach was adopted for this problem.

In developing the DP formulation we shall be building a unique model appropriate to our needs. But generally the

methodologies presented by Rao (1979) and Cooper (1981) in their books will be used as guidelines. The notation used will be the same as that used for the descriptive model in section 3.1 . A variable is prefixed by μ or σ to denote it's mean or standard deviation respectively. The subscript 'j' will be dropped from all variables for the rest of this chapter. The following new variables are also introduced.

μD_i = Expected time of arrival of assembly parts for processing at station i from a subcontractor $= E(A_i)$.

X_i = A random variable representing variation in the actual arrival of the assembly part for processing at station i.

3.2.1 Single-Job Multi-Station DP Model

In the proposed DP model each stage is defined to represent a workstation in the assembly line. These stages are interlinked by the process finish times. The finish time at the last stage being the job flow time. Stages are numbered similar to workstations. That is, the first workstation is stage one, and the last stage 'N'.

The primary objective of this research is to determine the optimal set of prescribed part delivery dates. Consequently the decision variable associated with each stage is μD_i. This is the date on which the subcontractor may be ex-

pected to deliver the assembly part. μD_i is related to the actual part arrival time by the following expression.

$$A_i = \mu D_i + \sigma D_i X_i \tag{3.8}$$

where,

σD_i = Known standard deviation of part delivery date.

Figure (3.2) shows a general stagewise DP model for a single subassembly. Here F_{i-1} is the input state variable to stage i, μD_i is the decision variable and X_i is a random variable. In this initial model the processing time is considered to be invariant. Therefore since assembly part delivery date is assumed to be normally distributed X_i will have a standard normal distribution.

Here we shall define the state of the system at any stage, by the variables required to make a decision at that stage. Hence the proposed DP model has a two dimensional state space. The state into any intermediate stage 'i' being given by (F_{i-1}). Since F_{i-1} is a random variable the state space can also be written as $(\mu F_{i-1}, \sigma F_{i-1})$. The initial state is F_0 which is the expected date of availability of subassembly for launch into production. Note that it is assumed F_0 is not known with certainity, otherwise the assumption of normality will be violated, making Clark's equations inapplicable.

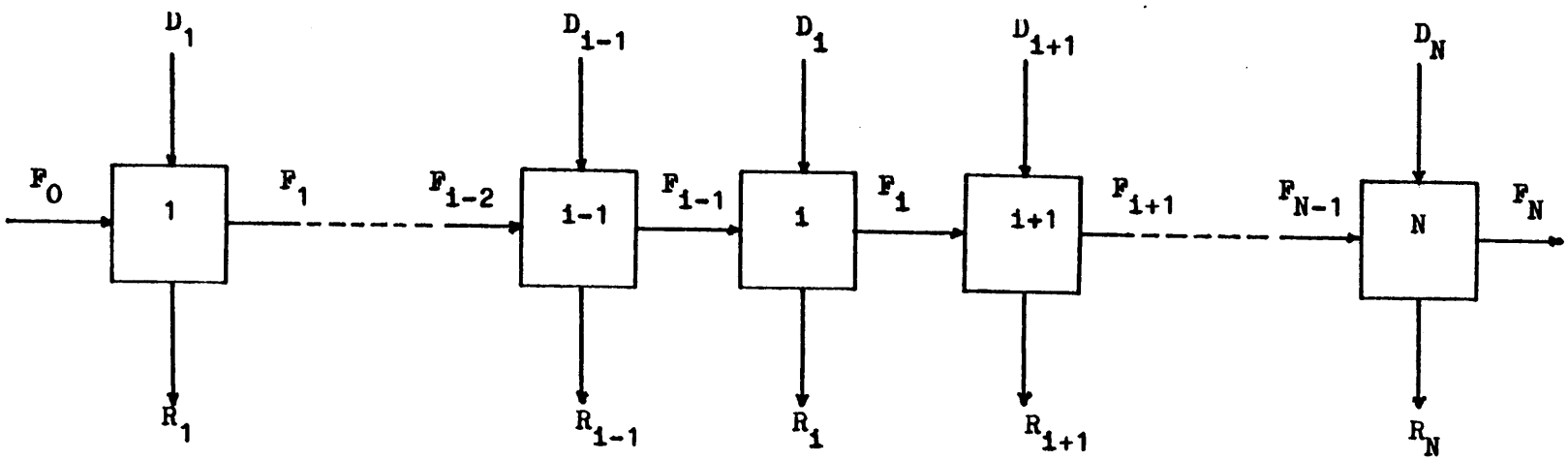


Fig. 3.2 Single Job multi-Station DP Model.

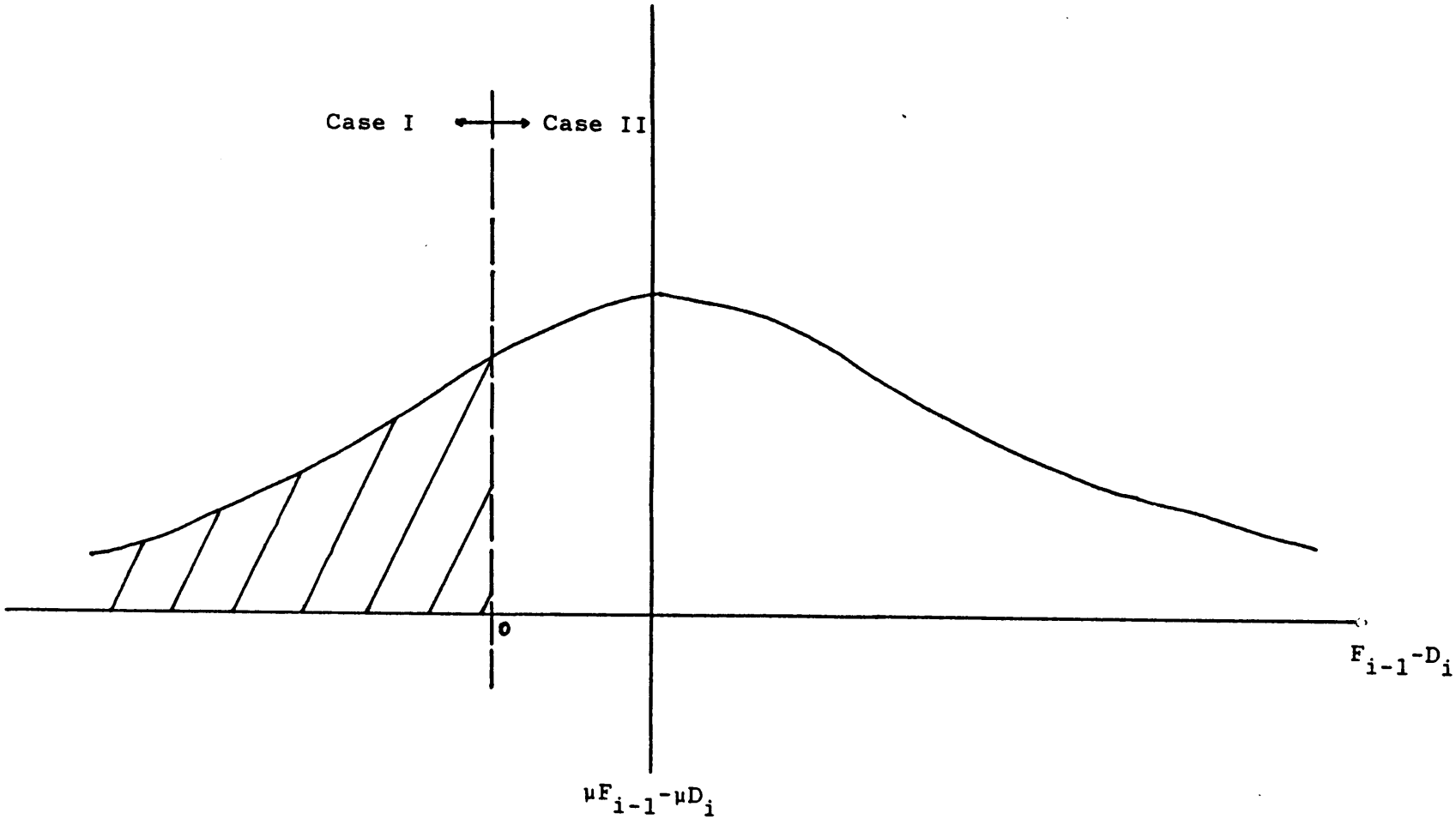


Fig. 3.3 Case Occurence Probabilities at Stage 'i'.

3.2.2 The Individual Stage Return Function

The stage return function reflects the optimality criterion of our objective function as specified previously in section 3.1.1. We shall define this function for stage 'i' to be, $G_i\{\mu D_i, \sigma D_i, F_{i-1}, X_i\}$. And the expected value of G_i is represented by R_i. Then for fixed values of the first three parameters of G_i, we would expect on an average a stage return of,

$$R_i\{\mu D_i, \sigma D_i, F_{i-1}\} = \int \phi(X_i)\ G_i\{\mu D_i, \sigma D_i, F_{i-1}, X_i\}\ dX_i \qquad (3.9)$$

An important property of the expected stage return is that it statistically represents an estimate of the average return for any one trial, even though it may not be possible to incur a cost R_i in practice.

Generally the outcome of any decision policy, which schedules due dates for the parts at every stage in the system, is uncertain. This gives rise to the problem of deciding how to compare the results of the various possible decision policies. There are many ways by which this comparison can be done and these are referred to as stochastic orderings. Suresh (1984) defines some of these methods. In our model we shall use what he terms as expected value orderings. This is defined as ordering of two random variables to be $X_i > X_j$ if $E[X_i] > E[X_j]$. Thus we would like to get a schedule in which the sum of the stage returns is stochastically min-

imized on the basis of expected value ordering. That is, $E[R^*]<E[R]$ where R^* is the cumulative expected return for the optimal schedule and R that for any other schedule of delivery dates.

The return function is a sum of the waiting time costs associated with the queueing of subassemblies, and assembly parts. Assuming that these costs are fixed at each station for all subassemblies and parts, the function G_i would take the following form in the deterministic case.

$$G_i = C_i\{F_i-P_i-A_i\} + M_i\{F_i-P_i-F_{i-1}\} \tag{3.10}$$

C_i = Unit waiting time cost for assembly parts (components) at workstation i.

M_i = Unit waiting time cost for subassemblies (mainframes) at workstation i.

In the stochastic case A_i would be substituted by equation (3.8), and G_i by it's expectation R_i. Further, one of two possible cases can occur during a state transformation at each stage. These cases are,

Case I: Arrival of assembly part limits start time.

Case II: Arrival of the subassembly from the previous station limits start time.

For each case a corresponding cost expression in equation (3.10) equates to zero. The return function is thus rewritten to be,

$$G_i = \begin{cases} C_i\{F_i - P_i - A_i\} = C_i\{F_{i-1} - A_i\} & \text{|Case II} \\ M_i\{F_i - P_i - F_{i-1}\} = M_i\{A_i - F_{i-1}\} & \text{|Case I} \end{cases} \tag{3.11}$$

The above equation is a function of the waiting times. But, the difference between two independent normal random variables x_1 and x_2, is known to be normally distributed with mean $\mu_1 - \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$. Using this property the following distribution is obtained to describe waiting time.

$$(F_{i-1} - A_i) = N\{(\mu F_{i-1} - \mu D_i), (\sigma F_{i-1}^2 + \sigma D_i^2)\}$$

Fig 3.3 displays the relationship between the above distribution, and the case occurence probabilities. Using this relationship, and the expression for the expectation of a truncated normal distribution [Johnson, 1970], the expected stage return is derived.

$$R_i = -M_i \int_{-\infty}^{0} \frac{x}{\sqrt{2\pi} a_i} \exp(-((x/\alpha_i) - \alpha_i)^2/2)\, dx$$

$$+ C_i \int_{0}^{\infty} \frac{x}{\sqrt{2\pi} a_i} \exp(-((x/\alpha_i) - \alpha_i)^2/2)\, dx \tag{3.12}$$

where,

$$\alpha_i = \{\mu F_{i-1} - \mu D_i\}/a_i$$
$$a_i^2 = \sigma F_{i-1}^2 + \sigma D_i^2$$

Further simplification reduces equation (3.12) to,

$$R_i = (M_i+C_i)(a_i/\sqrt{2\pi})\exp(-\alpha_i^2/2) + C_i(\mu F_{i-1}-\mu D_i) - (\mu F_{i-1}-\mu D_i)(M_i+C_i)\Phi(-\alpha_i) \qquad (3.13)$$

3.2.3 Single Stage Optimization

To minimize the return function , equation (3.13) is differentiated with respect to μD_i. Simplification of this differential results in,

$$\frac{dR_i}{d\mu D_i} = (M_i+C_i)\ \Phi\left[\frac{\mu D_i-\mu F_{i-1}}{\sqrt{(\sigma F_{i-1}^2+\sigma D_i^2)}}\right] - C_i \qquad (3.14)$$

The second derivative of equation (3.14) is,

$$\frac{d^2R_i}{d\mu D_i^2} = \frac{(M_i+C_i)}{\sqrt{2\pi}\ a_i}\ \exp\{-0.5((\mu D_i-\mu F_{i-1})/a_i)^2\}$$

Clearly the above expression is always going to be positive, implying convexity. Hence, equating the r.h.s of equation (3.14) to zero gives the optimal value of μD_i.

$$\mu D_i^* = \mu F_{i-1} - K_1(\sigma F_{i-1}^2+\sigma D_i^2)^{\frac{1}{2}} \qquad (3.15)$$

where,

$$K_1 = \Phi^{-1}[M_i/(M_i+C_i)]$$

Substitution of μD_i^* in equation (3.13) gives the optimal stage return.

$$R_i^* = K_2(\sigma F_{i-1}^2+\sigma D_i^2)^{\frac{1}{2}} \tag{3.16}$$

where,

$$K_2 = (M_i+C_i)\{K_1\phi[K_1]+[\exp(-K_1^2/2)/\sqrt{2\pi}]\} - C_iK_1$$

Equations (3.15) and (3.16) give the optimal values for a single stage studied in isolation. These equations have to be modified when considering recursive returns. An important property realized from these results, is that the only state variable to affect the optimal stage return is σF_{i-1}. This is to be expected, since the optimal decision is translated to adapt to the other state variable μF_{i-1}.

3.2.4 State Transformation Function

In the previous section the optimal stage return was shown to be a function of the state variable σF_{i-1}. In the recursive solution of the DP it is required to transfer the returns at any stage, to the stage preceding it before further optimization can be done. The recursive return is then differentiated with respect to the decision variable at that stage. For instance, if the return of stage 'i+1' is transferred to stage 'i' (see figure 3.2) and differentiated with respect to μD_i, then using equation (3.16),

$$\frac{\delta R_{i+1}}{\delta \mu D_i} = K_2 \frac{\delta \sigma F_i^{\,2}/\delta \mu D_i}{2\sqrt{\sigma F_i^{\,2}+\sigma D_{i+1}^{\,2}}} \tag{3.17}$$

It is required the r.h.s of the above equation be a function only of the input state variables of stage 'i'. Hence, σF_i should be expressed as a function of the input state variables of stage 'i'. This is the transformation function of the DP solution, and is derived as follows.

The variance of a random variable is defined by equation (3.18). Which is,

$$\sigma F_i^{\,2} = \{\text{2nd Moment of } F_i\} - \{\text{1st Moment of } F_i\}^2 \tag{3.18}$$

Clark's equations define the moments. Hence, substituting (3.4) and (3.5) in (3.18) gives the state transformation function. Since P_i is invariant it does not appear.

$$\begin{aligned}\sigma F_i^{\,2} = {} & (\mu F_{i-1}^{\,2}+\sigma F_{i-1}^{\,2})\Phi(\alpha_i) + (\mu D_i^{\,2}+\sigma D_i^{\,2})\Phi(-\alpha_i) \\ & + a_i(\mu F_{i-1}+\mu D_i)\psi(\alpha_i) - \mu F_{i-1}^{\,2}\Phi^2(\alpha) - \mu D_i^{\,2}(1-\Phi(\alpha_i))^2 \\ & - 2a_i\mu F_{i-1}\Phi(\alpha_i)\psi(\alpha_i) - 2a_i\mu D_i(1-\Phi(\alpha_i))\psi(\alpha_i) \\ & - a_i^{\,2}(\psi(\alpha_i))^2 - 2\mu F_{i-1}\mu D_i\Phi(\alpha_i)(1-\Phi(\alpha_i))\end{aligned} \tag{3.19}$$

The above equation is fairly complex and would be cumbersome when applied to the DP model. But, it is a true mathematical representation of the stage transformation. Differentiation of the above function is lengthy and tedious, and only the simplified result is presented here.

$$\frac{\delta\sigma F_i^{\,2}}{\delta\mu D_i} = (\sigma D_i^{\,2}-\sigma F_{i-1}^{\,2}-a_i^{\,2})\psi(\alpha_i)/a_i + 2\Phi(\alpha_i)\psi(\alpha_i)a_i$$

$$+ 2(\mu F_{i-1}-\mu D_i)[\Phi^2(\alpha_i)-\Phi(\alpha_i)] \qquad (3.20)$$

Substitution of Equations (3.19) and (3.20) in (3.17) results in an extremely complex equation (fig. 3.6), which is difficult to solve and is incompatible with the DP model. A more convenient method of applying these equations in an approximate fashion is derived in the following sections.

3.2.5 Two-Stage Problem

To simplify further derivations, a two stage problem is introduced. This problem is illustrated in Figure 3.4. The recurrence function H_i is also introduced, H_i being the total return from stage 'i' to 'N'. Then H_1 is given by equation (3.21), and it's derivative by (3.22) using (3.14).

$$H_1(\mu D_1) = R_1(\mu D_1) + R_2^{\,*}(\mu D_1) \qquad (3.21)$$

$$\frac{\delta H_1}{\delta\mu D_1} = -(M_1+C_1)\Phi(\alpha_1) + M_1 + \frac{\delta R_2^{\,*}}{\delta\mu D_1} \qquad (3.22)$$

In the previous section the last part of (3.22) was derived. But as seen, the final form was complex and posed severe computational difficulties. Hence, it is neccessary to determine an equivalent and simpler form of this equation, which can be conveniently applied to the DP model.

3.2.6 Convexity and Unimodality

Before proceeding further it is necessary to determine the nature of the function $H^*_{i+1}(\mu D_i)$, or specifically the relationship of R^*_{i+1} to μD_i after state transformation.

The function $\sigma F_i^2(\mu D_i)$ is defined by equation (3.19), and the general nature of this function is described in figure 3.5a. Since σF_i^2 is the variance of $\mathrm{Max}\{F_{i-1}, A_i\}$, it becomes asymptotic to the argument which dominates. That is when $\mu F_{i-1} > \mu D_i$ then σF_i^2 assumes a value closer to σF_{i-1}^2 than σD_i^2, and vice-versa.

The function $R^*_{i+1}(\sigma F_i^2)$ is defined by equation (3.16). The general nature of this function is also sketched in figure 3.5b. Since the function $\sigma F_i^2(\mu D_i)$ is asymptotic, the function $R^*_{i+1}(\mu D_i)$ will also be asymptotic. Hence, $R^*_{i+1}(\mu D_i)$ will have both a finite global maximum and minimum. The function $R_i(\mu D_i)$ has been proved to be convex. Hence, in the two-stage problem, H_1 is the summation of a convex function, and the function $R_2(\mu D_1)$. Consequently $H_1(\mu D_1)$ will not have a finite global maximum, but will have a global minimum since both $R_2^*(\mu D_1)$ and $R_1(\mu D_1)$ have lower bounds. However the unimodality of $H_1(\mu D_1)$ cannot be shown. But should $H_1(\mu D_1)$ unimodal, then it can be concluded that the function $H_1(\mu D_1)$, and generally $H_i(\mu D_i)$ is convex. We discuss this later in section 3.4.

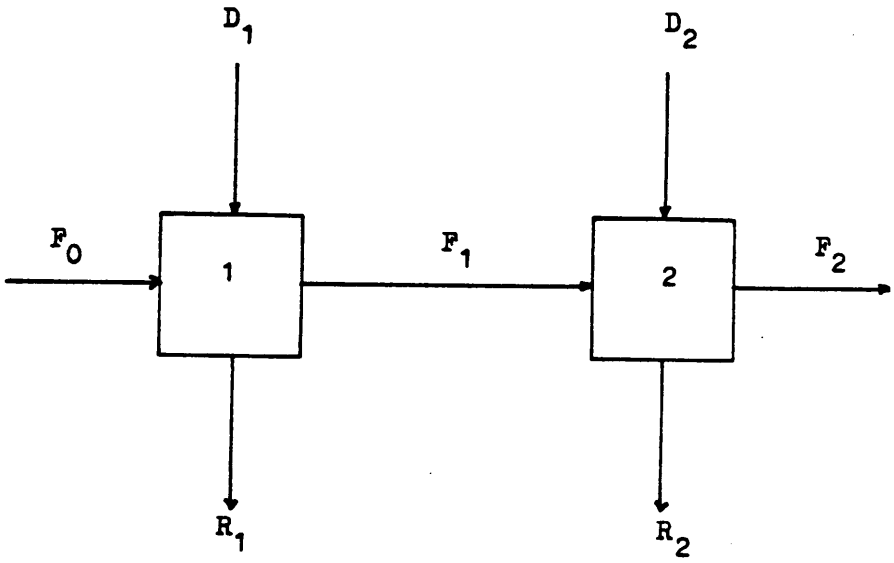


Fig. 3.4 Two Stage Problem.

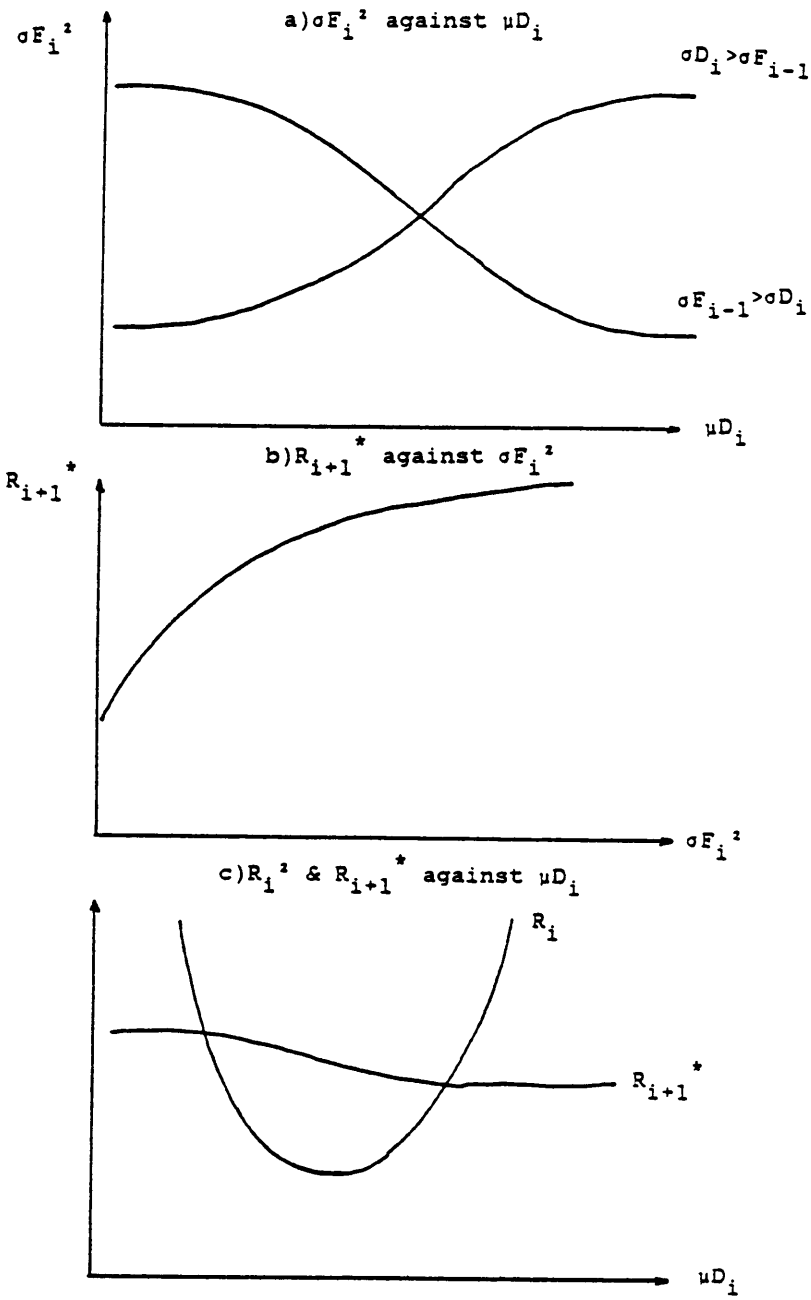


Figure 3.5 Nature of Transformation Curves.

$$\boxed{GRAD(\mu F_{i-1}, \sigma F_{i-1}, \mu D_i, \sigma D_i, K_2, K_3) = K_2(A)/\{2(B^2+K_3)^{\frac{1}{2}}\}}$$

where,

$$A = (\sigma D_i^{\,2} - \sigma F_{i-1}^{\,2} - a_i^{\,2})\psi(\alpha_i)/a_i + 2\Phi(\alpha_i)\psi(\alpha_i)a_i + 2(\mu F_{i-1} - \mu D_i)[\Phi^2(\alpha_i) - \Phi(\alpha_i)]$$

$$\begin{aligned} B = {} & (\mu F_{i-1}^{\,2} + \sigma F_{i-1}^{\,2})\Phi(\alpha_i) + (\mu D_i^{\,2} + \sigma D_i^{\,2})\Phi(-\alpha_i) \\ & + a_i(\mu F_{i-1} + \mu D_i)\psi(\alpha_i) - \mu F_{i-1}^{\,2}\Phi^2(\alpha) - \mu D_i^{\,2}(1-\Phi(\alpha_i))^2 \\ & - 2a_i\mu F_{i-1}\Phi(\alpha_i)\psi(\alpha_i) - 2a_i\mu D_i(1-\Phi(\alpha_i))\psi(\alpha_i) \\ & - a_i^{\,2}(\psi(\alpha_i))^2 - 2\mu F_{i-1}\mu D_i\Phi(\alpha_i)(1-\Phi(\alpha_i)) \end{aligned}$$

Figure 3.6 True Form of Function GRAD.

3.3 REGRESSION ANALYSIS

In this section it is intended to use regression analysis to find a model which closely predicts the value of $\delta R_{i+1}/\delta \mu D_i$ as defined by (3.17). The procedure consists of first generating a randomized data set using equations (3.17), (3.19) and (3.20). This data set will define the value of the dependent variable ($\delta R_{i+1}/\delta \mu D_i$) for numerous combinations of the independent variables (K_2, K_3, σD_i, σF_{i-1}, $\mu F_{i-1}-\mu D_i$). Where K_3 substitutes σD_i in equation (3.16).

In creating the data set, lower and upper bounds were placed on the independent variables. These bounds are shown in Table 3.2. These are required to increase the accuracy and reliability of any model developed. But the bounds do not detract from the generality of the model, since model parameters can be scaled down to be within bounds. The bound on $\mu F_{i-1}-\mu D_i$ is explained by the reasoning that in all probability $\mu D_i^* < \mu F_{i-1}$, since subassembly queue cost is generally higher than component cost.

3.3.1 Regression Model Development

In the two stage problem of section 3.2.4, second step optimization is achieved by equating the r.h.s of (3.22) to zero. For computational ease it is desirable $\delta R_2/\delta \mu D_1$, be of the same form as the remaining portion of the equation. That is, if GRAD is defined as the regression model, then

$$GRAD = fn\{\Phi[(\mu F_0 - \mu D_1)/a_1],\ \sigma D_1,\ K_2,\ K_3\}$$

For a given situation only the first argument of GRAD is unknown. And the sensitivity of GRAD to changes in this argument are especially important. Graphs of R_i and it's gradient are shown in Figures 3.7 and 3.8. From Figure 3.8 it is apparent there is a closely linear relationship between argument one and the gradient of R_i for a certain range of values. Plots for several other cases were determined, and observed to be of a similar shape. Hence, though the assumption of linearity is not proved here, it was decided to have GRAD linearly dependent on argument one for all proposed regression models.

3.3.2 Model Selection

Based on the above discussions several models were proposed and experimented with. Regression analysis was performed on these models to select the one with the best fit. The analysis was done by using the NLIN procedure of the SAS computer package. This procedure implements iterative methods that attempt to find least-squares estimates for non-linear models. A modified Gauss-Newton method is used for iteration, this involves regressing the residuals of the partial derivatives of the model with respect to the parameters until the iterations converge. Each model was studied using four different data sets, each of 1024 data records,

giving a total of 4096 records for comparision. All models were tested using the same data sets.

The SAS program listing gives estimates of the model parameters $\{C_1, C_2, ---\}$ and the mean residual square. Aggregate results over all data sets are shown for some models in Table 3.3. Clearly, model #3 is best, and was selected for further study. The selected model and parameter values are,

$$GRAD = C_1(\sigma D_1)^{C_2} K_2 \Phi(\alpha_i) + C_3(\sigma D_1)^{C_4} K_2 + C_6 K_2 K_3^{C_7} \qquad (3.23)$$

$C_1 = 0.04577093$	$C_5 = 2.3500$
$C_2 = -0.47805193$	$C_6 = 0.02354503$
$C_3 = -2.39312763$	$C_7 = -0.58349074$
$C_4 = -0.03059004$	

Further testing was done on the model to check for bias in the error direction. To do this, the predicted values of GRAD were compared with the actual values as given by (3.17). This was done for a large number of random data records and the frequency of different error sizes recorded. The results of the test are tabulated in table 3.4. GRAD was also observed to be on the lower side 60% of the time. On an average an absolute error of about 0.10 was observed. From these results, it can be concluded the model is a good estimator of the dependent variable, and can satisfactorily be applied to the DP solution.

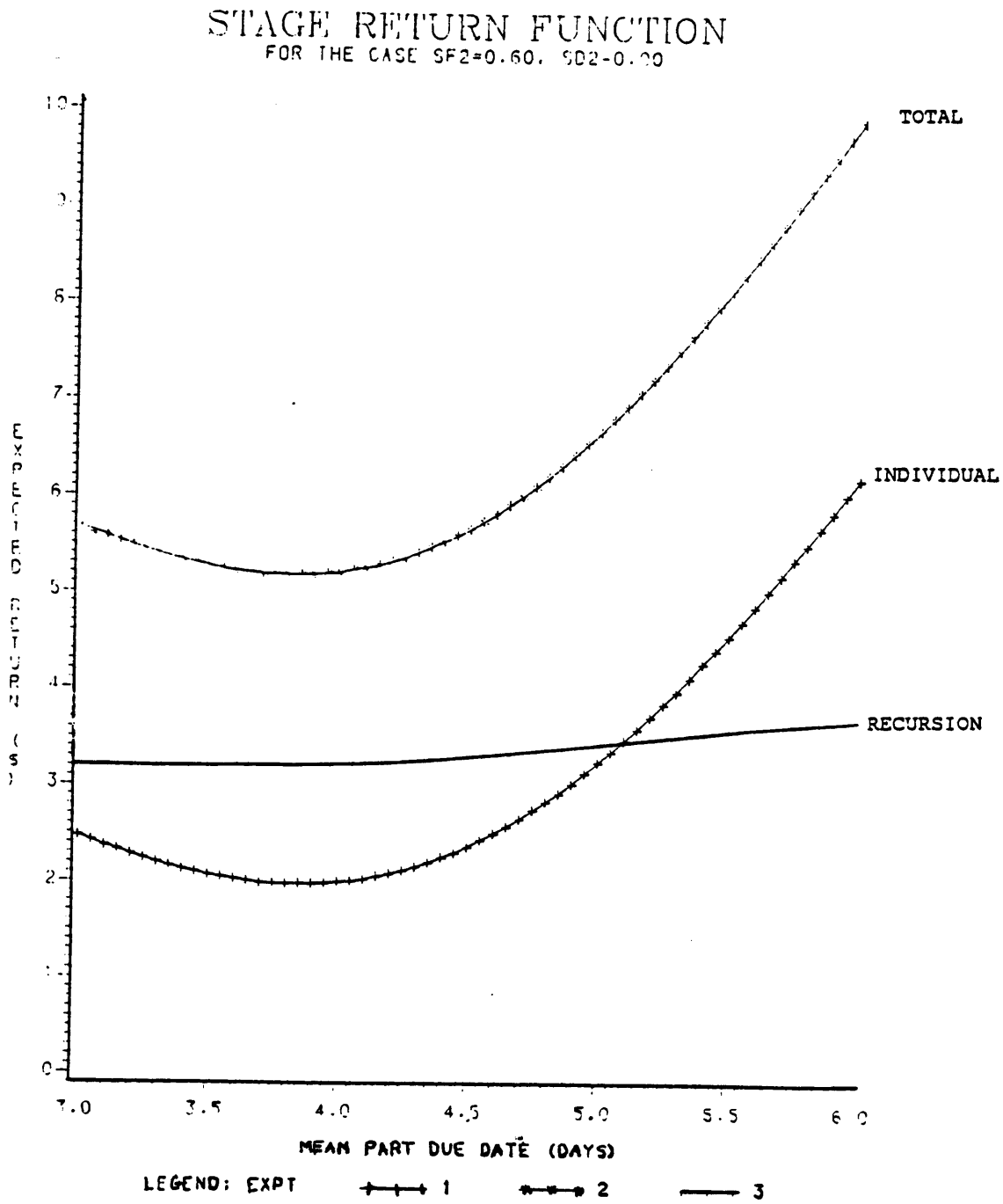


Fig. 3.7 Graph of Stage Return Function.

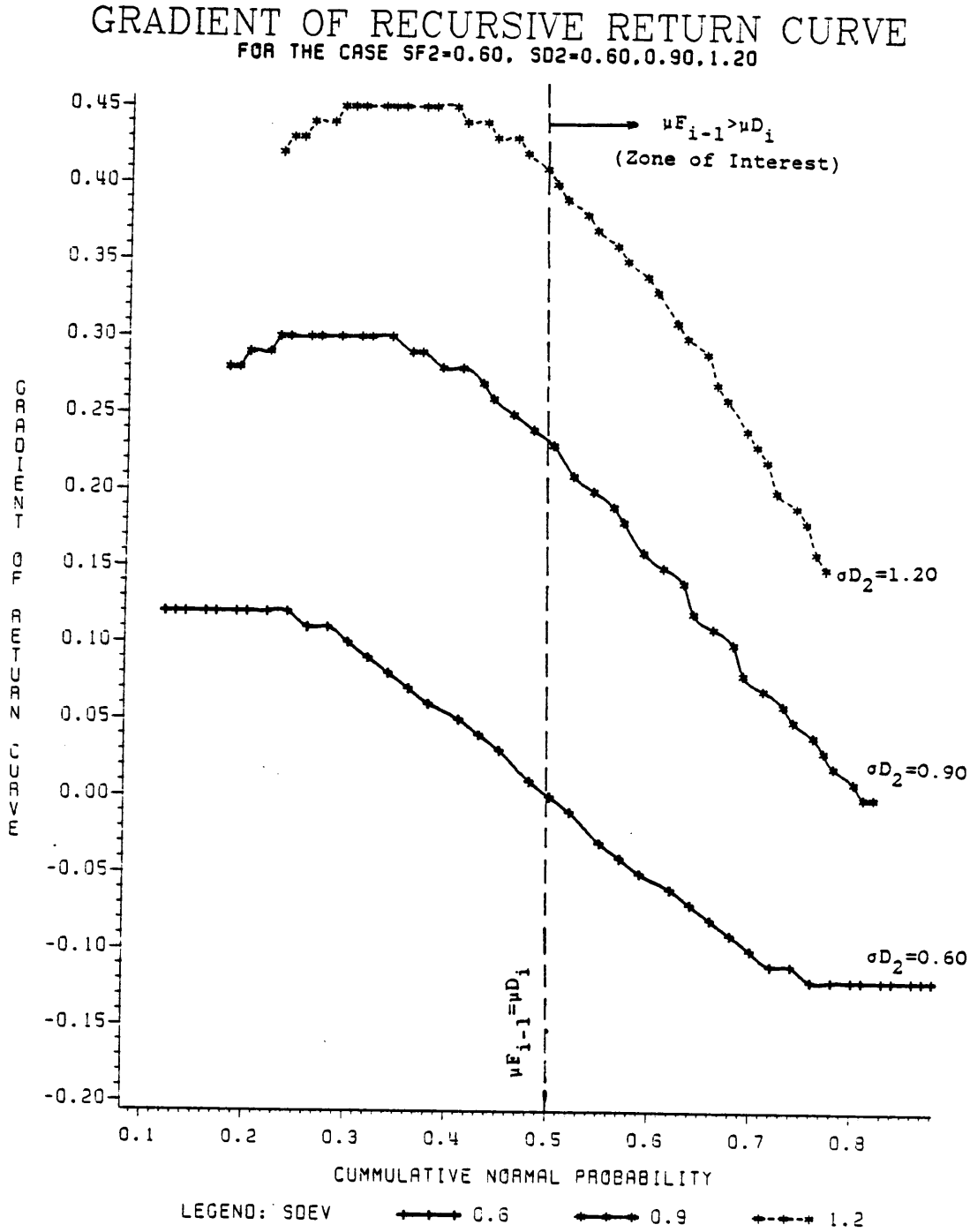


Fig. 3.8 Graph of Gradient of Recurrsive Return Curve.

Table 3.2 Bounds on Randomized Data for Regression.

Independent Variable	Lower Bound	Upper Bound
K_2	0	5
K_3	0	10
σD_i	0	5
σF_{i-1}	0	5
$\mu F_{i-1} - \mu D_i$	$\sigma/2$	1.5σ

Table 3.3 NLIN Regression Results.

Model #	Regression Model	# of. Itera.	Mean Res. Error
1	$C_1 \sigma D_1^{C^2} K_2 \Phi(\alpha_i)/K_3 + C_3 \sigma D_1 K_2 + C_4$	10	63.7
2	$C_1 \sigma D_1^{C^2} K_2 \Phi(\alpha_i) + C_3 \sigma D_1^{C_4} + C_5 K_3$	8	0.5321
3	$C_1 \sigma D_1^{C^2} K_2 \Phi(\alpha_i) + C_3 \sigma D_1^{C_4} K_2 + C_6 K_2 K_3^{C_7}$	17	0.0542

Table 3.4 Error Range Frequency of Regression Model.

Error Range	Frequency
-0.6 to -0.7	0
-0.5 to -0.6	20
-0.4 to -0.5	10
-0.3 to -0.4	23
-0.2 to -0.3	61
-0.1 to -0.2	55
0.0 to -0.1	314
0.1 to 0.0	657
0.2 to 0.1	297
0.3 to 0.2	59
0.4 to 0.3	33
0.5 to 0.4	23
0.6 to 0.4	22
0.7 to 0.6	19

Error = $(\delta R/\delta D)$ - GRAD

Total Number of Iterations = 1600

3.4 SOLUTION OF TWO-STAGE PROBLEM

It is now possible to solve the two-stage problem of section 3.2.5 to completion. The first decision in this solution, is selection of the optimal date for stage two, which is adapted from (3.15).

$$\mu D_2^* = \mu F_1 - K_1(\sigma F_1^2 + \sigma D_2^2)^{\frac{1}{2}} \tag{3.24}$$

Further, using equations (3.22) and (3.23) we get,

$$\frac{\delta H_1}{\delta \mu D_1} = (Q_1 - M_1 - C_1)\Phi(\alpha_1) + Q_2 + M_1 \tag{3.25}$$

where,

$$Q_1 = C_1(\sigma D_1)^{C_2} K_2$$

$$Q_2 = K_2\{C_3(\sigma D_1)^{C_4} + C_5 + C_6 K_3^{C_7}\}$$

Equating the r.h.s of (3.25) to zero and solving for μD_1^* defines the modal points of $H_1(\mu D_1)$.

$$\mu D_1^* = \mu F_0 - K_4(\sigma F_0^2 + \sigma D_1^2)^{\frac{1}{2}} \tag{3.26}$$

where,

$$K_4 = \Phi^{-1}\left[\frac{M_1 + Q_2}{M_1 + C_1 - Q_1}\right]$$

Since a unique value of μD_1^* is defined by equation (3.26), the function $H_1(\mu D_1)$ is unimodal. The second derivative of $H_1(\mu D_1)$ is,

$$\frac{\delta^2 H_1}{\delta \mu D_1^2} = \frac{(M_i + C_i - Q_1)}{\sqrt{2\pi}\, a_i} \quad \exp\{-0.5((\mu D_i - \mu F_{i-1})/a_i)^2\}$$

In section 3.2.6 it was shown that should $H_1(\mu D_1)$ be unimodal, then the modal point is a global minmum. But due to the introduction of the regression model, some system features may have been lost during state transformation. For μD_1^* to be a global minimum the second derivative of $H_1(\mu D_1)$ must be positive. This implies that M_1+C_1 should be greater than Q_1 In the event this does not hold the DP procedure cannot be continued. From the tests later conducted on the DP procedure, it was concluded that it is highly unlikely such a situation occurs. Given the condition is satisfied then equation (3.26) defines the optimal decision.

Substitution of equation (3.26) in (3.19) gives the variance of the finish time at stage one for an optimal decision policy.

$$\sigma F_1^2 = Z_1 \sigma F_0^2 + Z_2 \tag{3.27}$$

where,

$$Z_1 = K_4^2\{\Phi(K_4) - \Phi^2(K_4)\} + \Phi(K_4) - \psi^2(K_4) + K_4\psi(K_4)\{1-2\Phi(K_4)\}$$

$$Z_2 = Z_1 \sigma D_1^2 + \sigma D_1^2\{1-2\Phi(K_4)\}$$

Using the above equations the recurrence function for an optimal decision policy is obtained.

$$H_1^* = K_5(\sigma F_0^2+\sigma D_1^2)^{\frac{1}{2}} + K_6(\sigma F_0^2+K_7^2)^{\frac{1}{2}} \qquad (3.28)$$

where,

$$K_5 = (M_1+C_1)\{K_4\phi[K_4]+[\exp(-K_4^2/2)/\sqrt{2\pi}]\} - C_1K_4$$

$$K_6 = K_2\sqrt{Z_1}$$

$$K_7 = \{(Z_2+\sigma D_1^2)/Z_1\}^{\frac{1}{2}}$$

The solution procedure is completed by, numerically determing the values of Q_1, Q_2 and K_4 to solve for μD_1^*. Clark's first equation and (3.27) are then employed to estimate μF_1 and σF_1 respectively, which are in turn used to solve for μD_2^*. Total system return can be determined by using (3.28). But, a more accurate estimate of the total return is got by calculating individual stage return separately and summing. This eliminates the errors introduced by the regression model in calculation of the objective function. A numerical example is presented to illustrate the working of the DP solution.

3.4.1 Numerical Example

A hypothetical example of the two-stage problem is solved here. Suppose the following set of system parameters describes the example.

$\sigma D_2=1.75$	$M_2=3.5$	$C_2=0.40$
$\sigma D_1=2.00$	$M_1=3.0$	$C_1=0.50$
$P_1=6.0$	$P_2=5.0$	
$\mu F_0=10.0$	$\sigma F_0=0.50$	

The solution procedure involves determing numerical values stage wise, using expressions previously derived.

Stage 2

$M_2/(M_2+C_2) = 0.897$

$K_1 = \Phi^{-1}(0.897) = 1.267$

$K_2 = 4.6164$

$K_3 = \sigma D_2 = 1.75$

Stage 1

$Q_1 = 0.023$

$Q_2 = 0.0168$

$K_4 = \Phi^{-1}(0.910) = 1.115$

$\mu D_1^* = 10-1.115(2^2+0.5^2)^{\frac{1}{2}} = 7.7$

$K_5 = 0.7906$

$\mu F_1 = 16.14$

$\sigma F_1 = 0.64$

$\mu D_2^* = 16.14-1.267(0.64^2+1.75^2)^{\frac{1}{2}} = 13.78$

$R_2 = 1.30$

$R_1 = 1.63$

$H_1^* = 2.93$

This completes the solution for the two-stage problem. The solution methodology has to be modified for the multi-stage case.

3.4.2 Generalization to N-Stages

The solution for the two-stage problem is easily generalized to the multi-stage case. Since, there will be several

state transformations in the N-stage case, the subscript 'i' is added to certain parameters of the previous sections. The following set of equations form the the multi-stage DP solution. All expressions are derived from equations already presented in previous sections, and j varies from 2 to N-i+1.

$$H_i^* = K_{i,1,1}(\sigma F_{i-1}^2 + K_{i,1,2})^{\frac{1}{2}} + - -$$
$$- - + K_{i,N-i,1}(\sigma F_{i-1}^2 + K_{i,N-i,2})^{\frac{1}{2}}$$

$$\mu D_i^* = \mu F_i - OPTK_i(\sigma F_{i-1}^2 + \sigma D_i^2)^{\frac{1}{2}}$$

$$OPTK_i = T_i = \Phi^{-1}\left[\frac{M_i + SUMQ2_i}{M_i + C_i - SUMQ1_i}\right]$$

$$K_{i,1,1} = (M_i + C_i)\{T_i\Phi[T_i] + [\exp(-T_i^2/2)/\sqrt{2\pi}]\} - C_i T_i$$

$$K_{i,1,2} = \sigma D_i^2$$

$$K_{i,j,1} = K_{i-1,j+1,1}(Z_{i,1})^{\frac{1}{2}}$$

$$K_{i,j,2} = \{(Z_{i,2} + (K_{i-1,j+1,2})^2)/Z_{i,1}\}^{\frac{1}{2}}$$

$$Q_{i,j,1} = C_1(\sigma D_i)^{C_2} K_{i-1,j+1,1}$$

$$Q_{i,j,2} = K_{i-1,j+1,1}\{C_3(\sigma D_i)^{C_4} + C_5 + C_6(K_{i-1,j+1,2})^{C_7}\}$$

$$SUMQ1_i = \sum_{j=2}^{N-i+1} Q_{i,N-j,1}$$

$$SUMQ2_i = \sum_{j=2}^{N-i+1} Q_{i,N-j,2}$$

$$Z_{i,1} = T_i^2\{\Phi(T_i) - \Phi^2(T_i)\} + \Phi(T_i) - \psi^2(T_i)$$
$$+ T_i\psi(T_i)\{1 - 2\Phi(T_i)\}$$

$$Z_{i,2} = Z_{i,1}\sigma D_i^2 + \sigma D_i^2\{1 - 2\Phi(T_i)\}$$

At each stage, all of the above except H_i^* and μD_i^* can be evaluated numerically. Once backward recursion is completed, optimal decisions are computed from stage one to 'N'.

3.4.3 Three Stage Illustration of General Solution

The process of generalizing to N-stages is partially illustrated by the solution of the three stage problem. Given a three stage problem the optimal decisions for stages two and three are determined by modifying equations (3.24) and (3.26). Thus,

$$\mu D_3^* = \mu F_2 - OPTK_3(\sigma F_2^2+\sigma D_3^2)^{\frac{1}{2}} \tag{3.29}$$

$$\mu D_2^* = \mu F_1 - OPTK_2(\sigma F_1^2+\sigma D_2^2)^{\frac{1}{2}} \tag{3.30}$$

where,

$$OPTK_3 = \Phi^{-1}[M_3/(M_3+C_3]$$

$$OPTK_2 = \Phi^{-1}\left[\frac{M_2+Q_{2,1,2}}{M_2+C_2-Q_{2,1,1}}\right]$$

The recurrence function at stage one will be,

$$\begin{aligned} H_1(\mu D_1) &= R_1(\mu D_1) + R_2^*(\mu D_1) + R_3^*(\mu D_1) \\ &= R_1(\mu D_1) + K_{1,2,1}(\sigma F_0^2+K_{1,2,2})^{\frac{1}{2}} \\ &\qquad + K_{1,3,1}(\sigma F_0^2+K_{1,3,2})^{\frac{1}{2}} \end{aligned} \tag{3.31}$$

Using equation (3.22) and the regression model, the derivative of H_1 is determined.

$$\frac{\delta H_1}{\delta \mu D_1} = (Q_{1,2,1}+Q_{1,3,1}-M_1-C_1)\Phi(\alpha_1) + Q_{1,2,2}+Q_{1,3,2}+M_1$$

$$= (SUMQ1_1-M_1-C_1)\Phi(\alpha_1) + SUMQ2_1 + M_1 \qquad (3.32)$$

Equating the above to zero and substituting for α_1, we get

$$OPTK_1 = \Phi^{-1}\left[\frac{M_1+SUMQ2_1}{M_1+C_1-SUMQ1_1}\right]$$

and the optimal decision is,

$$\mu D_1^{*} = \mu F_0 - OPTK_1(\sigma F_0{}^2+\sigma D_1{}^2)^{\frac{1}{2}} \qquad (3.33)$$

Subject to the condition that M_1+C_1 is greater than $SUMQ1_1$. In general the following condition must hold at all stages.

$$M_i+C_i > SUMQ1_i$$

3.5 DP SOLUTION BY COMPUTER PROGRAM

The DP solution involves performing several calculations at each stage as given in section 3.4.2. A computer program was written to execute these computations, and print out the decisions. The program was written in WATFIV, and the listing is in Appendix A. IMSL subroutines MDNOR and MDNRIS were accessed to compute the normal probability and it's inverse.

Program variables are similar to those defined in section in 3.4.2. Stage returns are computed seperately and summed. This ensures an accurate estimate of system cost for the prescribed decision policy.

3.5.1 Testing of DP Solution

The optimality of the DP solution was studied by comparision with other decision policies. For, this the 10-stage problem used by Wilhelm and Ahmadi-Marandi (1983) was adopted as a test problem. The prescribed decision policy and run statistics for this problem, as per the DP solution are presented in table 3.5. The total CPU time was 0.03 secs, which is extremely economical when compared with the 2.0 secs required for a single simulation run.

A digital simulation model to describe the system behavior was developed. Five different decision policies were compared using this simulation model. The results of the tests are tabulated in table 3.6. In the simulation experiment, a 12% decrease in system cost was attained for the test problem, by using the DP solution against the deterministic scheduling rule. The shortest makespans were attained by using high safety factors with the buffer rule. Though, the results of the buffer rule for B=1 do seem comparatively good, further testing showed it to be good only for a small percentage of problems. Further testing was done with randomly generated problems.

The single stage optimzation rule does not consider stage interrelations. It analyzes each stage separately. The optimal policy is determined by a repetitive application of equation (3.15). But it is inferior only to the DP solution.

Table 3.5 DP Solution Results For Test Problem.

Stage $_i$	P_i	μD_i^*	μF_i	R_i
1	4	9.6	14.46	0.53
2	3	13.93	17.81	0.72
3	6	15.27	24.02	1.64
4	3	21.66	27.05	0.55
5	7	21.79	34.10	1.68
6	2	32.86	36.19	0.76
7	10	32.01	46.52	4.03
8	5	42.85	51.58	1.32
9	4	49.15	55.69	1.68
10	9	50.55	64.80	2.75

Computation Time = 0.03 secs

Statements Executed = 1391

Table 3.6 Comparative Results of Different Methods.

Scheduling Rule	Makespan (Days)	Cost $	Rel. Costs %
Deterministic[1]	66.77	17.42	100
B = 1.0[2]	64.31	15.85	91
B = 1.5	63.54	16.78	97
B = 2.0	63.17	19.79	114
Single Stage	65.30	15.68	90
DP Solution	64.78	15.33	88

[1] μD = deterministic estimates of start time.

[2] μD = Start Time - $B(\sigma D)$; Buffer rule.

3.6 PRESCRIBING CUSTOMER DUE-DATE

The flowtime of a job is given by F_N, which is also a random variable. Hence, the date on which a finished job is scheduled to be delivered to a customer, also forms part of the decision policy. Since a delay is subject to a tardiness cost, while early completion will incur a finished product inventory holding cost.

An additional stage (N+1) is introduced in the DP formulation, and the parameters and cost coefficients associated with this stage are,

$$\sigma D_{N+1} = 0.0$$

$$C_{N+1} = \text{Unit time tardiness cost}$$

$$M_{N+1} = \text{Unit time finished product inventory cost}$$

The optimal decision μD_{N+1}, is the optimal customer due-date. Should the customer due-date be pre-specified, then μD_{N+1} is equalized to this value, and the other decisions are translated accordingly.

Chapter IV

HEURISTIC SOLUTION OF THE MULTI-JOB PROBLEM

In our analysis of the assembly line scheduling problem till now, the mathematical model has been restricted, in that it considers only the processing of a single job (mainframe). But the descriptive model presented in section 3.1 described the behavior of a multi-job assembly line. In this chapter we shall attempt to extend the DP solution methodology presented in chapter three to the multi-job case.

Initially, it would seem rational to attempt to formulate the multi-job multi-station assembly line as a comprehensive DP model. But this approach introduces the problem of dimensionality. This arises since the input state variable of an intermediate stage includes both the finish time of the subassembly at the previous station, and the finish time of the previous subassembly at the station. Several modelling approaches were explored, in an attempt to circumvent the problem of dimensionality. Additional complications were introduced by the unusual nature of the state transformation, which is a maximization function. Consequently, no promising method was found. It would thus seem reasonable to assume that, even if the system is amenable to mathematical modelling, the resulting complexity would make the cost of optimization unjustifiable.

When dealing with complex computational problems it is often necessary to resort to the application of heuristic programming. In this method it is required to develop a set of educated "look back" and "look ahead" rules which, in conjunction with any derived procedures of suboptimization, could give solutions which are close to the optimum. Heuristic methods are often the most economical procedure of optimization, and it was decided to adopt this method for solving the problem of this thesis. The solution methodolgy will be developed from our insight and understanding of the inherent dynamics of the model.

4.1 A NETWORK MODEL OF THE ASSEMBLY LINE

An assembly line can be viewed as a channel in which products are flowing from one end to another, hence an equivalent digraph can be drawn to model the system behavior. This digraph will be of the converging tree type, with all paths leading to the finished product inventory. Such a graph is shown in figure 4.1 for the case when N=3 and J=3.

Each row in this network represents a particular subassembly. The interdependency between rows being denoted by the dashed lines, along which there is no physical flow but they represent machine availability. Since each station is visited by each subassembly once, the total number of stages in the network is given by NxJ. Therefore a subscript 'j' is added to each station number to denote a particular stage. That is,

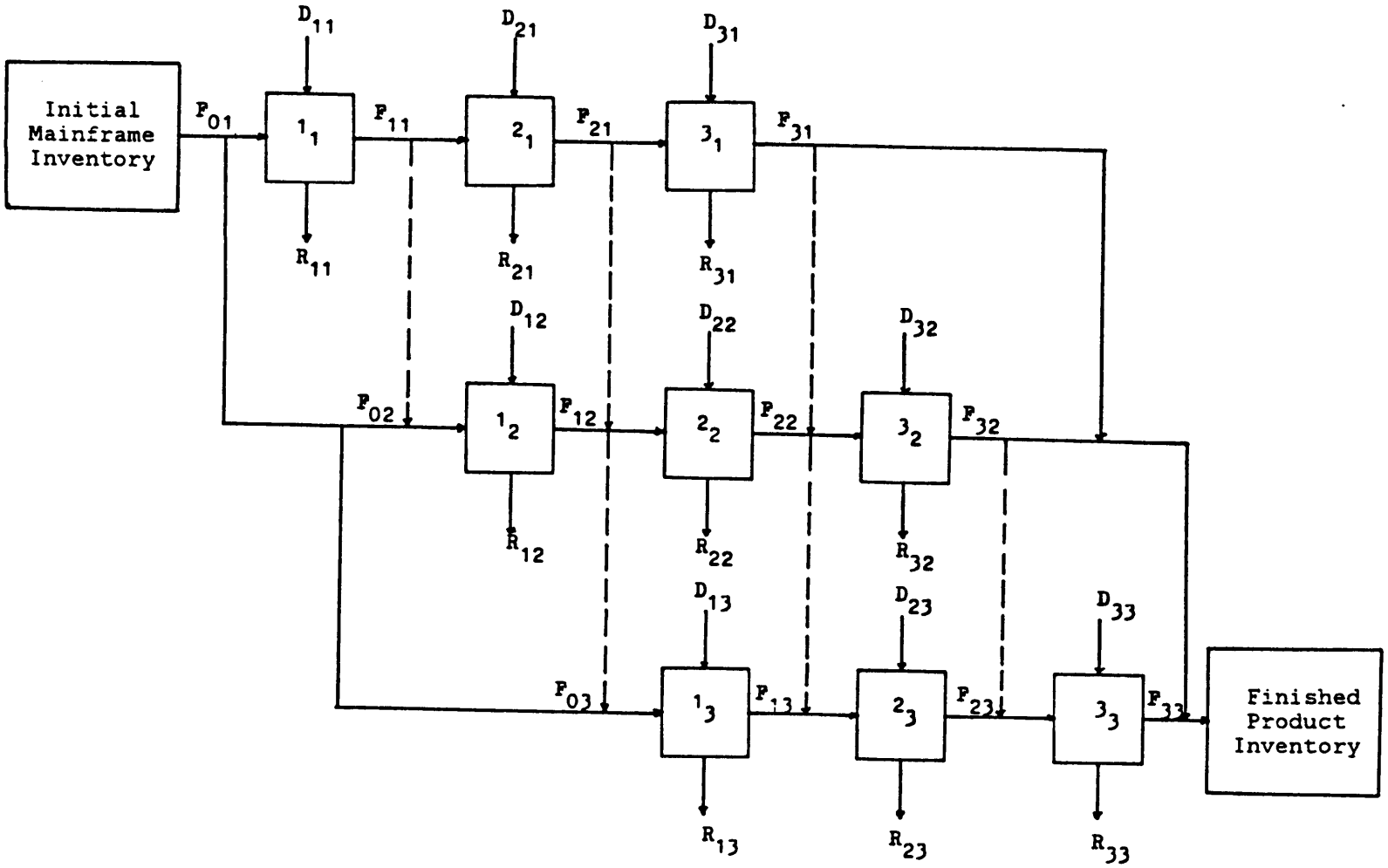


Fig. 4.1 Digraph of N=3 J=3 Assembly System.

stage i_j signifies the situation when subassembly j is at station i.[1]

The proposed network model is unique in that it is a grid in which every point is either a convergence or divergence of arcs. Clearly, there will be several paths connecting the initial mainframe inventory with the final finished product inventory. The makespan (flowtime) will be limited by the path with the longest duration. Also events occuring on any path are stochastic, consequently the makespan is also a random variable with expectation,

$$E(F_{NJ}) = E\{Max(Path\ Time)\}$$

4.1.1 Identification of the Critical Path

The path with the longest duration is normally termed the critical path (CP). Considering the dominance of the CP in the network, it can be expected that the scheduling of activities along it will be of primary importance in any process of optimization.

In a network with probabilistic time estimates, the CP can be selected by a stochastic ordering of path times to

---

[1] In strict network terminology the stages would be called nodes. Also note the system is being modelled as network simply for convenience, and there is no intent to apply any network algorithms.

determine the maximum. But in the proposed network this method is not readily applicable, since there is a set of presently unknown decisions on each path which will effect the moments of their completion times. Alternatively, the network can be assumed to be deterministic by ignoring the randomness introduced by vendor unreliability. The path completion times are then determined by summing the processing times of all stages on the path.[2] The CP can then be trivially identified.

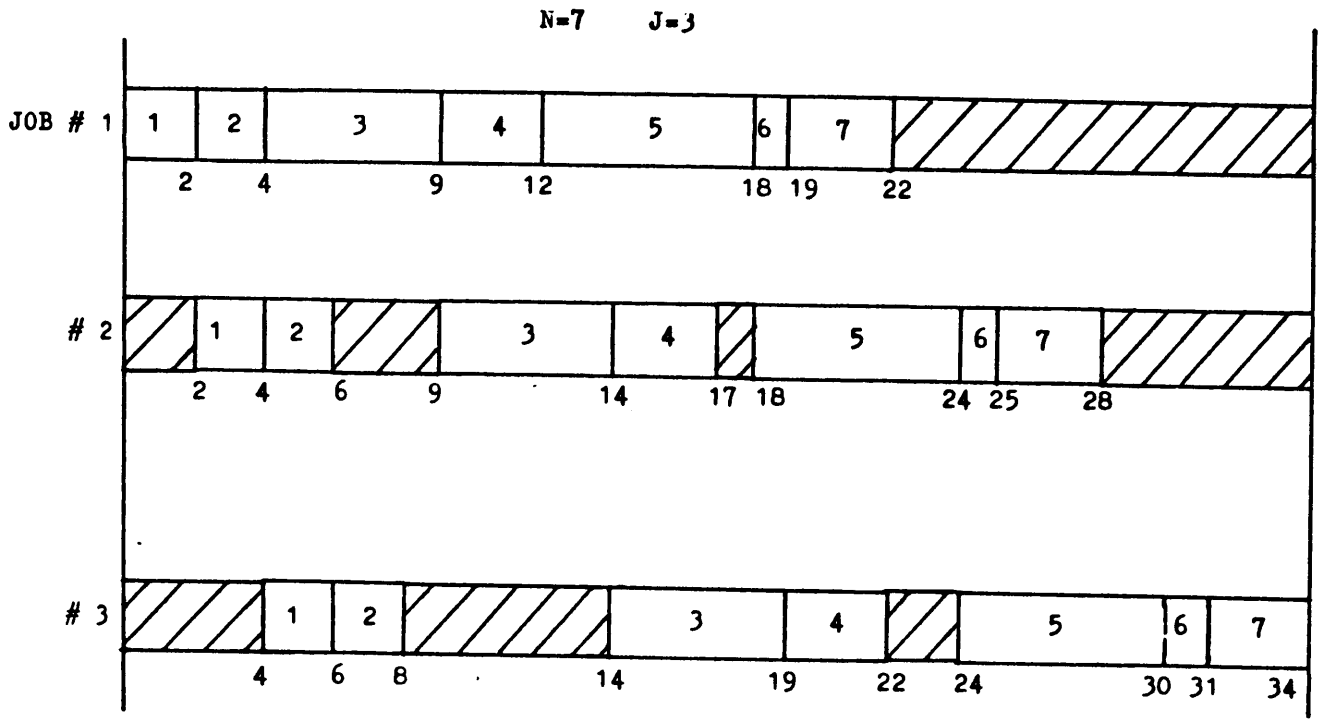
For instance, the Gantt chart for a specific system is depicted in figure 4.2. The completion times for some of the paths in this example are,

$$T_1 = 2+2+3(5)+3+6+1+3 = 32$$
$$T_2 = 2+2+5+3+3(6)+1+3 = 34$$
$$T_3 = 2+2+5+3+6+1+3(1) = 24$$

Path #2 would be the CP in this case. Generally the CP path would pass along the first and last rows, and intermediately pass diagonally across the bottleneck station. Thus, it is observed the CP also includes the critical stages.

---

[2] The actual elapsed time for all paths including slack, would be equal, but we are interested only in the total activity time of the path.



PATH T1:

$1_1 \to 2_1 \to 3_1 \to 3_2 \to 3_3 \to 4_3 \to$
$\to 5_3 \to 6_3 \to 7_3$

PATH T2:

$1_1 \to 2_1 \to 3_1 \to 4_1 \to 5_1 \to 5_2 \to$
$\to 5_3 \to 6_3 \to 7_3$

PATH T3:

$1_1 \to 2_1 \to 3_1 \to 4_1 \to 5_1 \to 6_1 \to$
$\to 7_1 \to 7_2 \to 7_3$

Fig. 4.2 Gantt Chart of Example System.

4.2 DEVELOPMENT OF THE HEURISTIC PROCEDURE

The DP solution enables optimization only of a chain of stages. Hence, if this solution technique is to be applied here, then it is necessary to experience a piecemeal process of optimization. That is, it will be required to select a particular chain from the network, optimize it as an independent element, and then accordingly make corresponding decisions in the system.

It is evident, the length of the CP will primarily determine makespan. Though the makespan is not the prime criterion of interest here, the total system cost can be expected to be positively correlated to it. Furthermore, there is no enforced slack time of inactivity in the CP, and any queueing of subassemblies on the CP will result in an equivalent increase in makespan. Thus the decision policy with regard to the chain of stages on the CP should always be made first.

In situations where the CP is significantly dominant over all other paths, the remaining stages can be optimized by considering their relationship to the CP. But when there are paths which are closely sub-critical to the CP, it is necessary to first schedule the activities on a critical sub-network of these paths.

Based on the above deliberations, a set of heuristic rules can be developed. In developing these rules the following set of assumptions are made.

1. All processing times are deterministic and known.
2. All jobs are identical.
3. The part arrival time is a normally distributed random variable with known variance.
4. The unit queue time cost of a subassembly increases progressively along the assembly line.
5. All jobs together form a batch, and are simultaneously delivered to the customer.
6. Total system cost is composed of the components described later in section 4.2.2.

4.2.1 Heuristic Programming Algorithm

The following steps describe in detail the piecemeal process of optimization adopted. The algorithm was developed by experimentation and, an insight to system behavior. The primary criterion of interest is the total system cost, and the goodness of the solution will be determined by this value. Figure 4.3 depicts what would be an optimal schedule for processing of jobs in a hypothetical case. In conjunction with the following steps it also illustrates the evolution process of the decision policy. The chains of stages referred to in describing the steps, can be identified in figure 4.3.

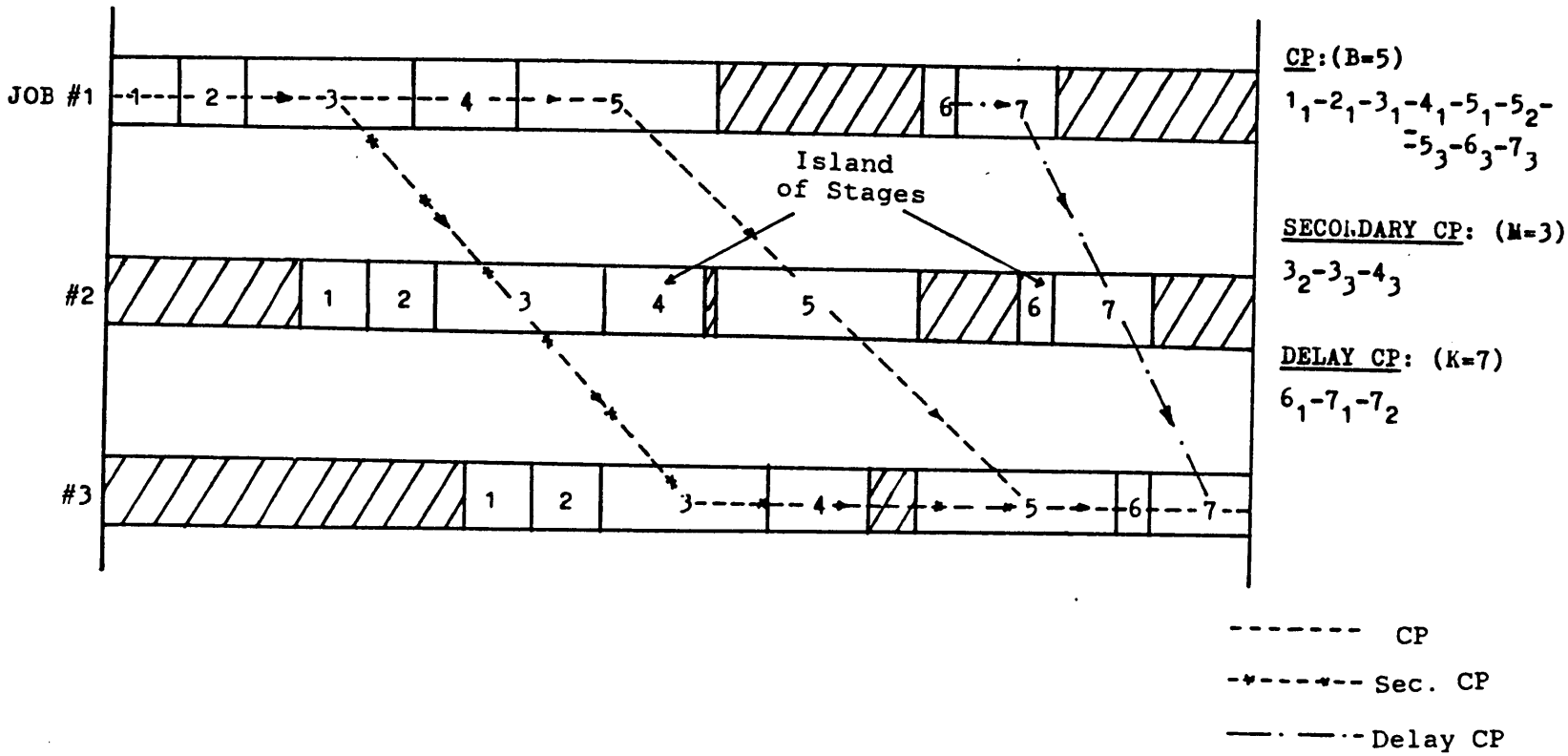


Fig. 4.3 Optimal Schedule Flowchart of Example system.

Step 1:

The CP is identified as described in section 4.1.2. For instance, if station 'B' is the bottleneck station, then the CP would be,

$$1_1 \dashrightarrow 2_1 \dashrightarrow \cdots \dashrightarrow B_1 \dashrightarrow \cdots \dashrightarrow B_J \dashrightarrow \cdots \dashrightarrow N_J$$

Decision variables along this chain of stages are determined by utilizing the DP solution technique. The customer duedates are also simultaneously determined as described in section 3.6.

With regard to the CP it was previously noted that subassembly queueing increases makespan. While analyzing the single-job case the makespan was not introduced in the objective function. But in an attempt to further enrich the model, an additional factor termed makespan cost is now introduced in the objective function. The makespan cost is defined as the cost of dedicating the manufacturing facility, to the processing of the present job for the period of the makespan. Thus, the makespan cost per unit time is added to the subassembly queue cost per unit time, prior to solving. This will ensure that a balance between part queueing and makespan is achieved.

Step 2:

The station with the longest processing time upstream of the bottleneck station is identified. If the ratio between the processing times for this station and the bottleneck

station is greater than 0.9, then it is assumed there is another path which is closely critical. The station in question is labelled M, and is termed the second order metering sation. The path is termed the secondary CP, and the portion of this path for which decisions are to be made is,

$$M_2 \dashrightarrow M_3 - - - \dashrightarrow M_J - - - \dashrightarrow (B-1)_J$$

When scheduling the above chain, an attempt is made to minimize its interference with the CP. Thus, there should be a safety slack between stages $(B-1)_J$ and B_J. This is achieved by using the theory presented in section 3.6. First the moments of the start time of processing at stage B_J is determined. Assuming this start time to be equivalent to the customer due date, decisions for the chain are then made utilizing the DP solution. Should there be no secondary CP, steps 2 and 3 are not executed.

Step 3:

This step determines the decisions for the island of stages, shown in figure 4.3, which are trapped in the critical sub-network. Scheduling is done for each job separately. Suppose job j is presently being considered then the chain would be,

$$(M+1)_j \dashrightarrow (M+2)_j - - - \dashrightarrow (B-1)_j$$

The DP solution technique is applied to the above chain. The start time of stage B_j will be equivalent to the customer due

date for this chain. The process is repeated for all jobs from j=2 to J-1.

Step 4:

The remaining stages upstream of the bottleneck station are now scheduled. Again, each job is considered separately. Thus, for job j the chain of interest would be,

$$1_j \dashrightarrow 2_j \dashrightarrow \cdots \dashrightarrow (M-1)_j$$

If no secondary CP exists, M is replaced by B. Start time of stage M_j will be equivalent to customer due date for this chain. The step is repeated for all jobs from j=2 to J.

Step 5:

The stages downstream of the bottleneck station remain to be scheduled. Since queueing costs increase progressively, an attempt to delay the processing of these stages is made. The procedure involves identifying the station with the longest processing time downstream of the bottleneck station, this is termed K. Then the chain of interest will be,

$$(B+1)_1 \dashrightarrow \cdots \dashrightarrow K_1 \dashrightarrow \cdots \dashrightarrow K_{J-1}$$

This is termed the delay CP. Decisions for this path are made using the DP solution technique. Start time of stage K_J will be equivalent to the customer due date for this chain.

In the second part of this step, the island of stages between the CP and delay CP are scheduled. The procedure is similar to step 3. The chain of interest being,

$$(B+1)_j \dashrightarrow (B+2)_j \dashrightarrow \cdots \dashrightarrow (K-1)_j$$

Step 6:

This step involves a repetitive application of step 4, until K=N. That is, after each execution of step 4, the delay CP is assumed to be the CP. A new station K is then determined.

Execution of the above steps, prescribes the overall decision policy for the system. Several variations of the above algorithm were experimented with, in an effort to improve the solution methodology. A synthesis of the inferences made from these experiments, was summarized in the formulation of the above steps. But to test the goodness of the heuristic program, the expected total system cost must be determined.

4.2.2 Total System Cost

The general objective function was previously defined in section 3.1.1. Based on that definition the objective function is restated here, with respect to the specific type of assembly line modelled. The total system cost consists of the costs which are associated with the operation of the assembly line.

1. Cost of subassemblies waiting at stations for processing. This also includes the cost of finished jobs waiting for batch completion.
2. Cost of assembly components waiting in inventory.
3. Cost of operating the assembly system. This is called makespan cost, it accounts for machine idle time cost.
4. Tardiness cost, incurred if batch is not delivered by customer due date.
5. Cost of holding finished batch in inventory, till customer due date.

4.2.3 Estimating System Cost

Given a set of decisions, and system parameters, the total system cost can be estimatated. As has already been noted, the model tends to become increasingly complex in the multi-job case. Even though good estimates of system returns were derived from the DP solution in the single job case, this will not be true for the multi-job case. This is because a heuristic piecemeal process of optimization is employed, resulting in a loss of some system characteristics.

Simulation is in many situations, the most effective method to replicate system behavior. And since the assembly system studied here is amenable to simulation modelling, it is an appropriate method to employ for cost estimation. Two methods of simulation are possible. First is the methodology used by Wilhelm and Ahmadi-Marandi (1982). This involves a

repetitive application of Clarks equation as previously described in section 3.1. From the computation point of view, this method is very cost economical.

The second method is to develop a digital simulation routine. This is an expensive, but more accurate method of estimation. Since only a single experiment is required for estimating cost, digital simulation was adopted here. The simulation program generates normal random deviates to represent the actual arrival dates of parts. The dates are based on the decision policy prescribed by the DP heuristic algorithm. Using the simulated dates, the assembly line behavior is replicated and system cost computed. The procedure is repeated for a hundred replications. The mean cost over all replications is computed, and this is assumed to be a reliable estimate of the expected system cost.

4.3 TESTING OF THE DP HEURISTIC PROCEDURE

The primary objective in testing a heuristic algorithm is to test its proximity to the optimal solution. But for the problem onhand, an estimate of the optimal system cost is not available. Neither does there exist any comparative method for scheduling the arrival of random parts. To substitute for this lack of information, a lower bound on the system cost is computed. Also a less sophisticated but good method of scheduling was developed.

The lower bound will be the minimal possible cost incurred in a deterministic system. This assumes that parts arrive just-in-time for processing, and subassembly queueing is due to line imbalance only. Of the total system cost, only a portion is due to the stochastic nature of the assembly line. Even when considering a deterministic system, subassembly waiting time costs caused by the unbalanced line, cannot be avoided. Furthermore, there is a lower bound to makespan cost. Hence, given a system it is possible to compute the minimal system cost by ignoring the stochastic element. In reality the expected system cost will always be greater. The minimal system cost can be utilized in estimating the goodness of the heuristic solution.

While testing the DP solution in section 3.5.1, the safety buffer rule (B=1) was found to give good results. In lieu of any other method being unavailable, the safety buffer method was extended to the multi-job case. A description of the procedure is presented in appendix B.

4.3.1 Computer Solution

A computer program written in WATFIV, was developed to replicate the computations of the heuristic procedure. A listing of this program is included in appendix C. The computer program of section 3.5 was adapted and included as subroutine DYNAM in this program. Five other subroutines are called by the main program. The subroutine SIMULA simulates

model behavior and computes expected system cost. The subroutines ROUTE, ROUTE2 and SUBOPT interface with the main program to control variable subscripts, and transalate decision policies into the system. The subroutine CLARK is based on Clark's equations, and computes the moments of the greatest of two normal random variables. A flowchart of the computer program is shown in figure 4.4.

Given a specific problem, system parameters consisting of the following, form the input data set.

1. Number of stations.
2. Number of jobs.
3. Unit queue time cost of assembly parts.
4. Unit queue time cost of subassemblies.
5. The standard deviation of part arrival times.
6. Job processing times.
7. Tardiness cost.
8. Finished product inventory cost.
9. Makespan unit time cost.

The program output is in the form of a table, which gives the prescribed mean part delivery dates; the finish time at each stage; the customer due date; and the expected total system cost. A sample output for a specific problem is given in table 4.1.

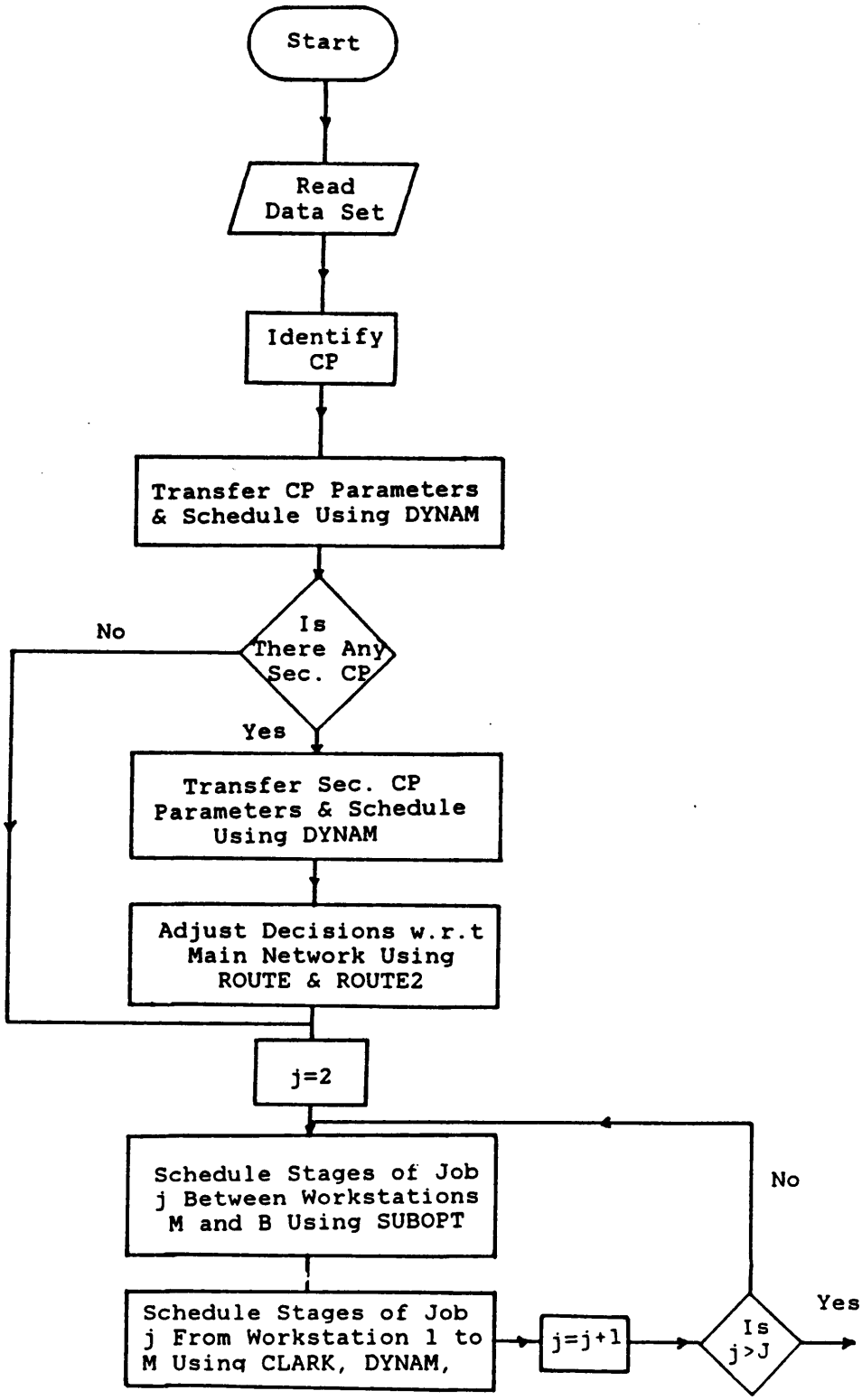


Figure 4.4 Flowchart of Computer Program

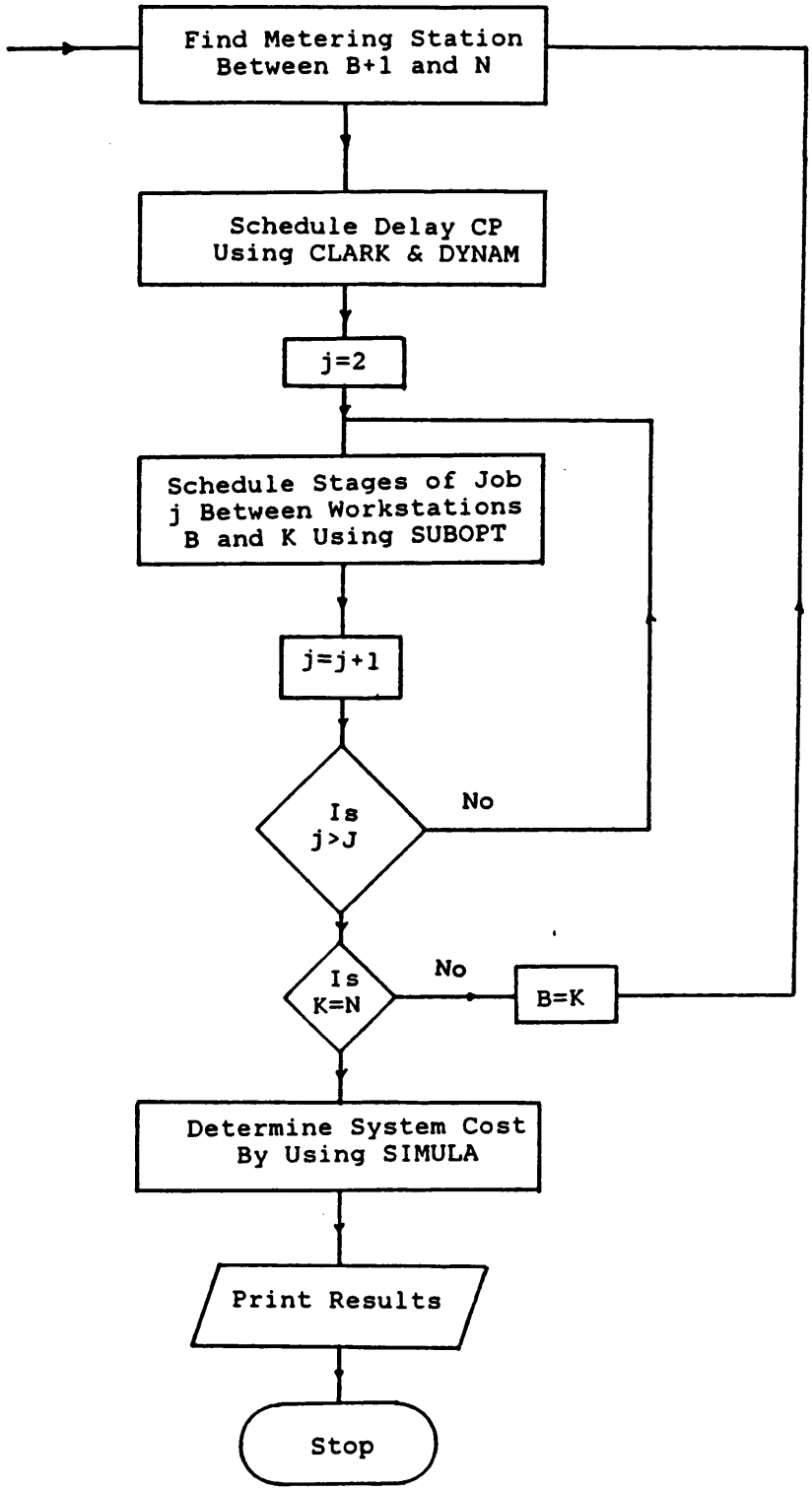


Figure 4.4 (contd.) Flowchart of Computer Program

OPTIMAL SCHEDULE FOR ASSEMBLY LINE OPERATION
SYSTEM COSTS ARE DETERMINED BY SIMULATION

STATION	1	2	3	4	5	6	7	8	9	10	JOB
DELIVERY DATE	8.17	13.07	14.98	25.00	26.51	34.26	54.41	59.81	64.13	97.62	
FINISH TIME	14.02	17.03	26.54	29.54	36.54	46.55	60.33	64.51	73.77	103.49	1
STAGE RETURN	0.75	0.50	1.03	0.32	0.90	0.96	21.01	0.85	1.28	99.61	
DELIVERY DATE	20.33	24.45	28.78	37.45	38.91	44.27	63.31	68.70	73.15	103.63	
FINISH TIME	24.33	27.84	38.41	41.54	48.62	58.62	69.29	73.52	83.15	110.63	2
STAGE RETURN	4.14	0.58	1.46	0.47	0.89	1.71	12.59	1.03	2.80	90.74	
DELIVERY DATE	31.05	35.15	38.39	47.07	48.58	54.27	72.34	77.73	82.18	109.64	
FINISH TIME	35.07	38.52	48.56	51.59	58.61	68.91	78.42	82.58	92.31	117.15	3
STAGE RETURN	8.42	0.54	1.05	0.35	0.92	2.52	9.48	0.79	3.22	79.68	
DELIVERY DATE	40.68	44.79	48.02	56.69	58.21	64.28	81.38	86.78	91.21	115.65	
FINISH TIME	44.50	48.03	58.32	61.33	68.37	79.01	87.26	91.58	101.45	123.47	4
STAGE RETURN	12.20	0.56	1.48	0.37	1.00	3.39	6.10	1.32	3.79	68.06	
DELIVERY DATE	50.33	54.44	57.67	66.34	67.88	74.30	90.41	95.81	100.24	121.66	
FINISH TIME	54.14	57.72	68.07	71.09	78.10	89.04	96.47	100.64	110.55	129.78	5
STAGE RETURN	16.06	0.60	1.59	0.40	0.96	4.00	3.90	0.83	4.07	56.52	
DELIVERY DATE	59.99	64.10	67.32	76.01	77.56	84.31	99.45	104.84	109.26	127.68	
FINISH TIME	64.01	67.41	77.70	80.72	87.73	99.04	105.63	109.80	119.64	136.05	6
STAGE RETURN	20.00	0.47	1.51	0.36	1.09	5.04	1.81	0.84	3.83	44.65	
DELIVERY DATE	69.65	73.76	76.97	85.67	87.24	94.32	108.48	113.88	118.29	133.69	
FINISH TIME	73.68	77.12	87.46	90.47	97.49	109.04	115.22	119.25	128.78	142.33	7
STAGE RETURN	23.87	0.54	1.55	0.39	1.12	5.44	0.99	0.49	2.77	32.64	
DELIVERY DATE	79.30	83.41	86.61	95.34	96.92	104.33	117.51	122.90	127.32	139.70	
FINISH TIME	83.43	86.77	97.16	100.18	107.20	119.04	125.06	129.07	138.20	148.42	8
STAGE RETURN	27.78	0.49	1.72	0.36	1.04	6.15	1.16	0.61	1.46	18.88	
DELIVERY DATE	88.96	93.06	96.24	104.99	106.55	114.34	126.54	131.94	136.34	144.74	
FINISH TIME	92.87	96.34	106.83	109.83	116.86	129.04	135.04	139.04	148.05	154.82	9
STAGE RETURN	31.55	0.54	1.80	0.37	0.95	6.86	1.74	0.91	1.37	4.87	
DELIVERY DATE	98.60	102.71	105.84	114.70	116.38	124.36	135.11	141.11	144.66	151.92	
FINISH TIME	102.58	105.95	116.40	119.41	126.41	139.04	145.04	149.04	158.04	164.04	10
STAGE RETURN	35.43	0.45	1.80	0.35	0.93	7.89	2.86	1.20	2.20	2.42	
PROCESS TIME	4.00	3.00	9.50	3.00	7.00	10.00	6.00	4.00	9.00	6.00	

EXPECTED TOTAL SYSTEM COST = 3333.42

TABLE 4.1 SAMPLE LISTING OF PRESCRIBED DECISION POLICY

4.3.2 Testing With Random Data

To test the goodness of the heuristic, a series of experiments were conducted on randomly generated data sets. For this an additional subroutine DATA was added to the main program. For each data set, the heuristic algorithm system cost, the safety buffer rule system cost, and the minimal system cost were computed. Since we are interested only in the portion of cost due to randomness, the following test statistic is computed.

Let,

HC = DP heuristic algorithm cost

MC = Lower bound on system cost

BC = Safety buffer system cost

Then,

$$\text{Cost Defficiency (\%)} = \frac{HC - MC}{BC - MC}$$

The lower the cost defficiency, the better the heuristic procedure is over the safety buffer method. Clearly this statistic depends on the goodness of the safety buffer method. Thus, a second test statistic is computed.

$$\text{Random Cost Ratio (\%)} = \frac{HC - MC}{MC}$$

This cost ratio measures the affect of randomness on total system cost. The smaller this ratio the better the better the performance of the DP heuristic procedure.

4.4 RESULTS OF TESTING

Extensive testing was done with the heuristic procedure. Table 4.2 lists the results of thirty random experiments. Detailed system parameters and results, for eight of these experiments are included in appendix D.

In all thirty experiments the DP heuristic procedure was superior to the safety buffer method. The cost defficiency ranged from 13.2% to 58.3%, with an average of about 35%. This implies that a two-thirds reduction in cost due to randomness can be expected, by using the D.P heuristic procedure.

The random cost ratio varies from 5.5% to 23.3% with an average of about 13%. Hence, we see that cost due to randomness can form a significant portion of total system cost. From table 4.3 it is apparent that there is no obvious correlation between the number of critical stations and the random cost ratio. The random cost ratio is primarily dependent on the variance of the part arrival dates.

Several factors affect the superiority of the DP heuristic algorithm. Foremost among these, is the presence of stations which have processing times close to that of the bottleneck station. These stations are termed critical stations. Table 4.3 illustrates the relationship between the number of critical stations and the percent cost defficiency. The actual correlation between these two variables, will depend on a host of other factors, such as, proximity of the

critical stations, variance of part arrival times, and the various unit time costs. But as shown in table 4.3 the superiority of the heuristic appears to deteriorate with an increasing number of critical stations.

In conclusion, the DP heuristic procedure does seem to be a good method for scheduling the assembly line. Judging from the low values of random cost ratio, the DP heuristic solution is probably very close to the optimal.

Table 4.2 Results of Random Data Tests

Expt. #	# of Jobs	# of Stns.	Lower Bound Cost	DP Heuristic Cost	Safety Buffer Cost	Cost Def. %	Random Cost %
1	5	16	1859	2198	2592	46.2	18.2
2	8	11	2275	2580	4219	15.7	13.4
3	7	11	1882	2321	2781	48.8	23.3
4	12	13	13348	14513	17684	26.9	8.7
5	6	19	2414	2815	3195	51.3	16.6
6	8	8	2115	2267	2652	28.3	7.2
7	6	13	2439	2744	3138	43.6	12.7
8	9	13	4718	5062	6469	19.6	7.3
9	9	7	2189	2408	3838	13.2	10.0
10	5	7	1013	1095	1341	25.0	8.1
11	9	10	3049	3447	4193	34.8	13.0
12	5	9	768	911	1210	32.2	18.6
13	14	17	22275	23495	29236	17.5	5.5
14	10	11	6045	6774	8788	26.6	12.1
15	6	13	2514	2888	3816	28.7	14.9
16	6	13	2439	2744	3138	43.6	12.5
17	5	18	2196	2689	3775	31.2	22.5
18	13	14	13860	15859	17289	58.3	14.4
19	12	20	12832	14402	16540	42.3	12.2
20	9	17	3978	4437	5754	25.8	11.5

Table 4.2 (contd.) Results of Random Data Tests

Expt. #	# of Jobs	# of Stns.	Minimal Cost	DP Heuristic Cost	Safety Buffer Cost	Cost Def. %	Random Cost %
21	11	8	5825	6147	7884	15.6	5.5
22	14	16	15059	16543	25230	14.6	9.8
23	12	24	21761	24007	27064	42.4	10.3
24	11	19	11914	13580	15317	49.0	14.0
25	7	12	3715	3960	5057	18.2	6.6
26	9	21	7786	8794	11558	26.7	13.0
27	7	5	1292	1433	2142	16.6	10.9
28	10	8	5507	5957	6716	37.2	8.2
29	10	6	5882	6231	7824	18.0	5.9
30	11	21	13010	14328	16579	37.0	10.1

Table 4.3 Percent Cost Defficiency and Critical Stations

Expt. #	# of Stns.	# of Critical Stations	Cost Def. %	Random Cost %
22	16	1	14.6	9.8
2	11	1	15.7	18.2
6	8	2	28.3	7.2
26	21	3	26.7	13.0
17	18	3	31.2	22.4
7	13	3	43.6	12.5
19	20	4	42.3	12.2
3	11	4	48.8	23.3
18	14	5	58.3	14.4
5	19	6	51.4	16.6

Chapter V

SUMMARY

This thesis presents a prescriptive methodology for scheduling the arrival of assembly parts from unreliable vendors. The arrival time of a part was modelled as a normally distributed random variable. First the single job multi-station problem was solved, and later the proposed methodology was extended to the multi-job multi-station case. The descriptive model assumes that the maximum of two normal random variables is also normally distributed. This is the only element of approximation in the descriptive model.

For the single job case, the problem was mathematically formulated and solved using stochastic dynamic programming, with the objective of minimizing total system cost. The process of optimization involved selecting the dates on which vendors are scheduled to deliver their orders. The optimal decisions at a stage were found to be dependent solely on the variance of mainframe arrival times. The state transformation function was derived in its true mathematical form, and found to be quite complex. The inability to apply this function directly resulted in the development of a more convenient form using non-linear regression analysis. The regression equation was incorporated into the model to obtain a working solution. Due to the errors introduced by the approximation of the regression model, the DP solution has to be termi-

nated when a certain condition is violated. This condition is described in section 3.4.3, and may occur for very large problems. Test results showed the DP solution to be superior to those obtained using other techniques.

In analyzing the multi-job case the problem was modelled as a digraph. The problem of dimensionality restricted further application of a pure DP approach. Consequently, a heuristic procedure of optimization, based on a critical network of stages was developed. The DP solution method was integrated with the heuristic procedure to obtain the prescribed decision policy.

The unavailability of a comparable scheduling method, restricted the extent to which the proposed methodology could be tested. However studies were made to estimate the goodness of the solution. The only element of approximation in the DP solution for the single-job case is the regression procedure. Had the true state transformation function been used, the resulting solution would have been optimal. It can thus be expected, that the DP solution is close to the optimal solution. This claim was supported by the test results.

As is common with most heuristics, there was a lack of rigorous mathematical sophistication in developing the multi-job solution. Hence, it is difficult to hypothesize on the quality of the solution. Therefore, a battery of random experiments were conducted with the proposed method. The results indicate that the solution is superior to that ob-

tained by the safety buffer method, and are in proximity of the lower bound on expected system cost. The proposed methodology is also cost economical with regard to computation time. For an average problem (N=10, J=10) the execution time is about 2.5 secs, of which a major portion is taken by the simulation routine to estimate system cost.

5.1 CONCLUSION

Assembly manufacture involving slow moving assembly lines is a common feature in industry. Assembly lines of this kind typically involve discrete parts, large cycle times, and expensive products. In the majority of cases, these setups have high levels of inventory stock, resulting in considerable non-productive capital costs. The application of the proposed DP heuristic method can significantly reduce the in-process inventory costs. Furthermore it results in a better estimate of system cost and production flow times.

A better relationship with vendors and customers is developed as a consequence. Also the system can be rescheduled periodically enabling the development of an on-line scheduling system.

5.2 AREAS OF FUTURE RESEARCH

Several assumptions were made during the development of the proposed methodology. Relaxation of these assumptions

will open up new areas of research. Some of these are as follows.

1. All jobs were assumed to be identical. The methodology should be extended to the case when the processing times are not identical. This additional feature will not effect the working of the DP solution methodology. But in executing the heuristic procedure, a new method for identifying the CP will have to be found.
2. Random processing times should be considered. Wilhelm and Johnson (1983) developed a descriptive model for assembly line operations with random processing times and parallel machines. Their model should be incorporated into the DP heuristic procedure.

   The random processing time will make the present state transformation function invalid. The variance of the processing time at a stage, will have to be included in equation (3.19). Consequently, a new regression model has to be developed.

   With regard to the heuristic procedure, a method of stochastic orderings will have to be employed to determine bottleneck stations
3. In many situations the part is supplied from an in-house manufacturing facility, which also has to be scheduled. It is necessary to determine an approach to consider this possibility.

The above suggestions will increase the extent of the applicability of the DP heuristic procedure. But there also exist opportunities for improving the goodness of the methodology itself. The quality of the regression fit will primarily determine the goodness of the prescribed methodology. Attention should be focused on developing a better regression model, which not only improves the solution but also eliminates the error condition introduced by the present regression model. Also modifications to the heuristic algorithm should be explored.

APPENDIX A

DP Solution Computer Program Listing

```
//WATFIV       DAS,T=10
C
C*************************************************************************
C
C PROGRAM TO DETERMINE THE OPTIMAL MEAN PART DUE DATES IN A MULTI STAGE
C SINGLE JOB ASSEMBLY LINE. OPTIMIZATION IS BASED ON CLARK'S EQUATIONS,
C WILHELM'S METHODOLGY, AND THE DEVELOPED REGRESSION MODEL.
C
C THE METHOD OF OPTIMIZATION IS BACKWARD RECURRSION STOCHASTIC DYNAMIC
C PROGRAMMING USING VARIANCES AND EXPECTATIONS. ALL INHERENT STOCHASTIC
C PROCESSES ARE ASSUMED TO BE NORMAL IN NATURE.
C
C STAGES ARE NUMBERED BACKWARDS.(I.E THE LAST STAGE NUMBER ONE)
C
C*************************************************************************
C
C VARIABLES
      REAL Q(20,2,20),K(20,9),CR(20,20,2),Z(20,2),CS(20),CP(20),SD(20)
      REAL UD(20),UF(20),SF(20),CPS,OPTK(20),P(20),RET(20),SUMR,GRET,G
      REAL PI,UDI(20)
      INTEGER NS,I,J,M,N,Y
      RHO(Z)=1/(EXP((Z**2)/2)*SQRT(2*3.1416))
C
C-----------------------------------------------------------------
C READ INPUT DATA FOR PRESENT CASE AND DEFINE REGRESSION CONSTANTS.
C-----------------------------------------------------------------
C
      READ, NS
      READ, (CP(I), I=1,NS)
      READ, (CS(I), I=1,NS)
      READ, (SD(I), I=1,NS)
      READ, (P(I), I=1,NS)
      READ, UF(NS), SF(NS)
      C1=0.04577093
      C2=-0.47805193
      C3=-2.39312763
      C4=-0.03059004
      C5=2.355
      C6=0.02354503
      C7=-0.58349074
      PI=3.1415926
C
C-----------------------------------
C COMMENCE FIRST STAGE OPTIMIZATION.
C-----------------------------------
C
      CPS=CS(1)+CP(1)
      K(1,1)=CS(1)/CPS
      CALL MDNRIS(K(1,1),K(1,2),Y)
      OPTK(1)=K(1,2)
      K(1,3)=CPS*((1/(EXP((K(1,2)**2)/2)*SQRT(2*PI)))+(K(1,2)*K(1,1)))
```

```
      K(1,3)=K(1,3)-(CS(1)*K(1,2))
      K(1,4)=SD(1)
      CR(1,1,1)=K(1,3)
      CR(1,1,2)=K(1,4)
      N=1
      IF (N .EQ. NS) GO TO 1000
C
C-------------------------------------------------------------
C DETERMINE RETURN EQUATION COEFFICIENTS FOR REMAINING STAGES.
C-------------------------------------------------------------
C
      DO 30 I=2,NS
        CPS=CS(I)+CP(I)
        K(I,1)=CS(I)/CPS
        CALL MDNRIS(K(I,1),K(I,2),Y)
        K(I,3)=CPS*((1/(EXP((K(I,2)**2)/2)*SQRT(2*PI)))+(K(I,2)*K(I,1)))
        K(I,3)=K(I,3)-(CS(I)*K(I,2))
        K(I,4)=SD(I)
        SUMQ1=SUMQ2=0
        M=I-1
        DO 31 J=1,M
          Q(I,1,J)=C1*CR(I-1,J,1)*(SD(I)**C2)
          Q(I,2,J)=(C3*(SD(I)**C4))+C5+(C6*(CR(I-1,J,2)**C7))
          Q(I,2,J)=CR(I-1,J,1)*Q(I,2,J)
          SUMQ1=SUMQ1+Q(I,1,J)
          SUMQ2=SUMQ2+Q(I,2,J)
   31   CONTINUE
        K(I,5)=CPS-SUMQ1
        K(I,6)=CS(I)+SUMQ2
        K(I,7)=K(I,6)/K(I,5)
        IF (K(I,7) .GE. 1.00) K(I,7)=0.97
        CALL MDNRIS(K(I,7),K(I,8),Y)
        OPTK(I)=K(I,8)
        K(I,9)=CPS*((1/(EXP((K(I,8)**2)/2)*SQRT(2*PI)))+(K(I,8)*K(I,7)))
        K(I,9)=K(I,9)-(CS(I)*K(I,8))
        CR(I,1,1)=K(I,9)
        CR(I,1,2)=K(I,4)
        R=K(I,7)
        U=RHO(K(I,8))
        Z(I,1)=K(I,8)*((K(I,8)*(R-(R**2)))+(U*(1-(2*R))))
        Z(I,1)=Z(I,1)+R-(U**2)
        Z(I,2)=(Z(I,1)+(1-(2*R)))*(SD(I)**2)
        DO 32 J=2,I
          CR(I,J,1)=CR(I-1,J-1,1)*SQRT(Z(I,1))
          CR(I,J,2)=SQRT((Z(I,2)+(CR(I-1,J-1,2)**2))/Z(I,1))
   32   CONTINUE
   30 CONTINUE
C
C-------------------------------------------
C COMPUTE OPTIMAL SET OF MATERIAL DUE DATES.
C-------------------------------------------
C
      NS1=NS-1
```

```
      DO 40 M=1,NS1
        I=NS+1-M
        VAR=SQRT((SD(I)**2)+(SF(I)**2))
        UD(I)=UF(I)-(OPTK(I)*VAR)
        ALPHA=(UF(I)-UD(I))/VAR
        CALL MDNOR(ALPHA,PHI)
        PSI=RHO(ALPHA)
        UF(I-1)=(UF(I)*PHI)+(UD(I)*(1-PHI))+(VAR*PSI)+P(I)
        US12=(UF(I)**2)+(SF(I)**2)
        US22=(UD(I)**2)+(SD(I)**2)
        SECM=(US12*PHI)+(US22*(1-PHI))+((UF(I)+UD(I))*VAR*PSI)
        SF(I-1)=SQRT(SECM-((UF(I-1)-P(I))**2))
        UDI(I)=UF(I)-(K(I,2)*VAR)
   40 CONTINUE
 1000 VAR=SQRT((SD(1)**2)+(SF(1)**2))
      UD(1)=UF(1)-(OPTK(1)*VAR)
      UDI(1)=UD(1)
      ALPHA=(UF(1)-UD(1))/VAR
      CALL MDNOR(ALPHA,PHI)
      PSI=RHO(ALPHA)
      FTIME=(UF(1)*PHI)+(UD(1)*(1-PHI))+(VAR*PSI)+P(1)
      US12=(UF(1)**2)+(SF(1)**2)
      US22=(UD(1)**2)+(SD(1)**2)
      SECM=(US12*PHI)+(US22*(1-PHI))+((UF(1)+UD(1))*VAR*PSI)
      STIME=SQRT(SECM-((FTIME-P(1))**2))
C
C-------------------------------------------
C COMPUTE TOTAL AND INDIVIDUAL STAGE RETURNS.
C-------------------------------------------
C
      SUMR=GRET=0
      DO 50 I=1,NS
         RET(I)=CR(I,1,1)*SQRT((SF(I)**2)+(CR(I,1,2)**2))
         SUMR=SUMR+RET(I)
         GR=CR(NS,I,1)*SQRT((SF(NS)**2)+(CR(NS,I,2)**2))
         GRET=GRET+GR
   50 CONTINUE
C
C----------------------------------------
C PRINT RESULTS OF OPTIMIZATION PROCESSES.
C----------------------------------------
C
      PRINT 60
   60 FORMAT ('1',7X,'OPTIMAL SET OF MATERIAL DELIVERY DATES BY D.P. MET
     CHOD'/6X,55('_'))
      DO 61 I=1,NS
        PRINT 62, I,UD(I)
   62   FORMAT ('0'/,11X,'OPTIMAL DELVERY DATE STAGE (',I2,') =',F7.2)
        PRINT 63, UF(I)
   63   FORMAT (13X,'MEAN SUBASSEMBLY ARRIVAL DATE =',F7.2)
        PRINT 64, SF(I)
   64   FORMAT (9X,'VARIANCE SUBASSEMBLY ARRIVAL DATE =',F7.2)
```

```
109          PRINT 65, RET(I)
110     65   FORMAT (19X,'INDIVIDUAL STAGE RETURN =',F7.2)
111          PRINT 66, UDI(I)
112     66   FORMAT (18X,'INDEPENDENT OPTIMAL DATE =',F7.2)
113     61 CONTINUE
114        PRINT 67, SUMR
115     67 FORMAT ('0'/,3X,'CUMMULATIVE STAGE RETURN =',F8.3)
116        PRINT 68, GRET
117     68 FORMAT ('0',2X,' RECURRSIVE STAGE RETURN =',F8.3)
      C
      C-----------------------------
      C PROGRAM TERMINATION TO OCCUR.
      C-----------------------------
      C
118        STOP
119        END
      C
      C
      C

      //DATA
```

APPENDIX B

DP Heuristic Procedure Computer Program Listing

```
//WATFIV        DAS,T=60
C
C***********************************************************************
C
C PROGRAM TO DETERMINE THE OPTIMAL MEAN PART DUE DATES IN A MULTI STAGE
C MULTI JOB ASSEMBLY LINE. THIS IS THE MAIN PROGRAM DESIGNED TO HANDLE
C THE INHERENT DIMENSIONALITY OF THE PROBLEM.
C
C THE PROGRAM INTERFACES WITH THE SUBROUTINES:
C                          1. DYNAMIC
C                          2. CLARK
C                          3. SUBOPT
C                          4. ROUTE
C                          5. ROUTE2
C                          6. SIMULA
C                          7. DATA
C
C STAGES ARE NUMBERED BACKWARDS.(I.E THE LAST STAGE IS NUMBER ONE)
C
C DECISIONS HAVE A STARTING TIME BASE OF 10.
C
C MAXIMUM NUMBER OF STATIONS = 25
C
C     MAXIMUM NUMBER OF JOBS = 15
C
C***********************************************************************
C
C VARIABLES
      REAL QCS(25),QCP(25),CS(40),CP(40),SD,SFIN,UFIN,UD
      REAL UF,SF,P(40),TIME(25),TOTAL,TCOST,WCOST
      REAL PI,UDI,DCOST,TARD,WARH,SDEV(25),RET,EFS
      INTEGER NS,I,J,M,N,Y,NC,NJ,NSUB,MAX,BUF
      DOUBLE PRECISION DSEED
      COMMON /A1/ SFIN(25,15),UFIN(25,15),UD(25,15),MAX,NJ
      COMMON /A2/ UDI(40),SD(40),UF(40),SF(40)
      COMMON /A3/ RET(25,15),EFS(26,15),TOTAL,TDELV
      COMMON /A4/ TARD,WARH,TCOST,WCOST,QF,QCOST,COST
      COMMON /A5/ DSEED,SRT(25,15)
      RHO(Z)=1/(EXP((Z**2)/2)*SQRT(2*3.1416))
C
C-------------------------------------------
C GENERATE INPUT DATA FOR ASSEMBLY LINE MODEL.
C-------------------------------------------
C
      READ, DSEED
      NX=1
 1001 CALL DATA(TIME,QCP,QCS,SDEV,DCOST,TARD,WARH,QF,NS,NJ)
      PI=3.1415926
C
C--------------------------------------
C IDENTIFY CRITICAL PATH IN FLOW NETWORK.
C--------------------------------------
```

```
C
      X=XTIME=0
      DO 10 I=1,NS
        XTIME=XTIME+TIME(I)
        IF (TIME(I) .GT. X) THEN
          MAX=I
          X=TIME(I)
        ENDIF
   10 CONTINUE
      XTIME=XTIME+((NJ-1)*TIME(MAX))+10
      SRT(1,NJ)=XTIME-TIME(1)
      NC=NS+NJ
      DO 11 I=2,MAX
        CP(I)=QCP(I-1)
        CS(I)=QCS(I-1)+DCOST
        SD(I)=SDEV(I-1)
        P(I)=TIME(I-1)
        SRT(I,NJ)=SRT(I-1,NJ)-TIME(I)
   11 CONTINUE
      DO 12 I=1,NJ
        CP(MAX+I)=QCP(MAX)
        CS(MAX+I)=QCS(MAX)+DCOST
        SD(MAX+I)=SDEV(MAX)
        P(MAX+I)=TIME(MAX)
   12 CONTINUE
      IF (MAX .EQ. NS) GO TO 18
      L=MAX+NJ+1
      DO 13 I=L,NC
        K=I-NJ
        CP(I)=QCP(I-NJ)
        CS(I)=QCS(I-NJ)+DCOST
        SD(I)=SDEV(I-NJ)
        P(I)=TIME(I-NJ)
        SRT(K,NJ)=SRT(K-1,NJ)-TIME(K)
   13 CONTINUE
   18 CP(1)=TARD
      CS(1)=WARH
      P(1)=0
      SD(1)=0.001
      K=NJ-1
      DO 14 J=1,K
        L=NJ-J
        SRT(MAX,L)=SRT(MAX,L+1)-TIME(MAX)
        IF (MAX .EQ. NS) GO TO 14
        M=MAX+1
        DO 15 I=M,NS
          SRT(I,L)=SRT(I-1,L)-TIME(I)
   15   CONTINUE
   14 CONTINUE
C
C-------------------------------------
C OPTIMIZE CRITICAL PATH USING DYNAMIC.
```

```
C-------------------------------------
C
      CALL DYNAM(NC,CP,CS,P,0.1)
      DO 20 I=2,MAX
        UD(I-1,NJ)=UDI(I)
        UFIN(I-1,NJ)=UF(I-1)
        SFIN(I-1,NJ)=SF(I-1)
   20 CONTINUE
      DO 21 I=1,NJ
        UD(MAX,NJ+1-I)=UDI(MAX+I)
        UFIN(MAX,NJ+1-I)=UF(MAX+I-1)
        SFIN(MAX,NJ+1-I)=SF(MAX+I-1)
   21 CONTINUE
      IF (MAX .EQ. NS) GO TO 24
      L=MAX+NJ+1
      DO 22 I=L,NC
        UD(I-NJ,1)=UDI(I)
        UFIN(I-NJ,1)=UF(I-1)
        SFIN(I-NJ,1)=SF(I-1)
   22 CONTINUE
   24 TDELV=UDI(1)+(0.2*TIME(MAX))
      IF (MAX .EQ. NS) GO TO 60
C
C-----------------------------------------------------------------------
C IDENTIFY IF PATH THROUGH 2ND DEGREE METERING STATION IS SIGNIFICANTLY
C CRITICAL. IF SO OPTIMIZE USING DYNAMIC.
C-----------------------------------------------------------------------
C
      CRIT=0.9*TIME(MAX)
      L=MAX+1
      MAX2=MAX
      DO 30 I=L,NS
        IF (TIME(I) .GE. CRIT) THEN
          MAX2=I
          QUIT 2
        ENDIF
   30 CONTINUE
      IF (MAX2 .EQ. MAX) GO TO 50
      NC2=MAX2-MAX+NJ-1
      L=NC2-NJ+1
      DO 31 I=2,L
        CP(I)=QCP(MAX-1+I)
        CS(I)=QCS(MAX-1+I)
        SD(I)=SDEV(MAX-1+I)
        P(I)=TIME(MAX-1+I)
   31 CONTINUE
      L=L+1
      DO 32 I=L,NC2
        CP(I)=QCP(MAX2)
        CS(I)=QCS(MAX2)
        SD(I)=SDEV(MAX2)
        P(I)=TIME(MAX2)
```

```
   32 CONTINUE
      CP(1)=DCOST
      CS(1)=QCS(MAX+1)
      SV=SFIN(MAX2,1)
      UCOM2=UFIN(MAX2,1)
      CALL CLARK(SFIN(MAX,NJ-1),UFIN(MAX,NJ-1),SDEV(MAX),UD(MAX,NJ),
     1SCOM,UCOM)
      SD(1)=SCOM
      CALL DYNAM(NC2,CP,CS,P,SV)
      TRANS1=UCOM2-10
      TRANS2=UDI(1)+TRANS1-UCOM
      IF (TRANS2 .LE. 0) THEN
        TRANS=UCOM-UDI(1)
      ELSE
        TRANS=TRANS1
      ENDIF
      L=MAX2-MAX
      CALL ROUTE(NJ,L,TRANS,MAX)
      NL=NJ-1
      CALL ROUTE2(MAX2,NJ,L,NL,TRANS)
C
C-----------------------------------------------------------------
C OPTIMIZE STAGES BETWEEN PRIMARY AND SECONDARY CRITICAL PATHS.
C-----------------------------------------------------------------
C
   40 J=2
      E=DCOST
      F=QCS(MAX)
   41 CALL SUBOPT(MAX2,MAX,J,E,F,CP,CS,P,QCS,QCP,TIME,SDEV)
      J=J+1
      IF (J .LE. NJ-1) GO TO 41
C
C-------------------------------------------------------
C OPTIMIZE SUB-BRANCHES TO THE LEFT OF CRITICAL NETWORK.
C-------------------------------------------------------
C
      IF (MAX2 .EQ. NS) GO TO 60
   50 J=2
      NSUB=NS-MAX2+1
      DO 51 I=2,NSUB
        CP(I)=QCP(MAX2+I-1)
        CS(I)=QCS(MAX2+I-1)
        P(I)=TIME(MAX2+I-1)
        SD(I)=SDEV(MAX2+I-1)
   51 CONTINUE
      CP(1)=DCOST
      CS(1)=QCS(MAX2)
      P(1)=0
   52 L=J-1
      CALL CLARK(SFIN(MAX2,L),UFIN(MAX2,L),SD(MAX2),UD(MAX2,J),
     1SCOM,UCOM)
      SD(1)=SCOM
```

```
148          CALL DYNAM(NSUB,CP,CS,P,0.1)
149          TRANS=UCOM-UDI(1)
150          CALL ROUTE(J,NSUB,TRANS,MAX2)
151          IF (J .LT. NJ) THEN
152            J=J+1
153            GO TO 52
154          ENDIF
       C
       C-----------------------------------------------------
       C OPTIMIZE SUB-BRANCHES TO THE RIGHT OF CRITICAL NETWORK.
       C-----------------------------------------------------
       C
155       60 MAX4=MAX
156          L=MAX-1
157       66 IF (L .EQ. 0) GO TO 75
158          X=0
159          DO 61 I=1,L
160            IF (TIME(I) .GT. X) THEN
161              MAX3=I
162              X=TIME(I)
163            ENDIF
164       61 CONTINUE
165          NSUB=NJ+MAX4-MAX3-1
166          NQ=NJ-1
167          DO 62 I=2,NJ
168            CP(I)=QCP(MAX3)
169            CS(I)=QCS(MAX3)
170            SD(I)=SDEV(MAX3)
171            P(I)=TIME(MAX3)
172            SRT(MAX3,NJ+1-I)=SRT(MAX3,NJ+2-I)-TIME(MAX3)
173       62 CONTINUE
174          IF (MAX3 .EQ. MAX4-1) GO TO 55
175          IZ=NJ+1
176          DO 63 I=IZ,NSUB
177            L2=MAX3+I-NJ
178            CP(I)=QCP(L2)
179            CS(I)=QCS(L2)
180            SD(I)=SDEV(L2)
181            P(I)=TIME(L2)
182            DO 69 K=1,NQ
183              J=NJ-K
184              SRT(L2,J)=SRT(L2-1,J)-TIME(L2)
185       69   CONTINUE
186       63 CONTINUE
187       55 CP(1)=DCOST
188          CS(1)=QCS(MAX3)
189          CALL CLARK(SFIN(MAX3+1,NJ),UFIN(MAX3+1,NJ),SDEV(MAX3),
            1UD(MAX3,NJ),SCOM,UCOM)
190          SV=SFIN(MAX,1)
191          SD(1)=SCOM
192          CALL DYNAM(NSUB,CP,CS,P,SV)
193          TRANS1=UFIN(MAX,1)-10
```

```
      TRANS2=UDI(1)+TRANS1-UCOM
      IF (TRANS2 .LE. 0) THEN
        TRANS=UCOM-UDI(1)
      ELSE
        TRANS=TRANS1
      ENDIF
      DO 64 I=2,NQ
        L2=NJ+1-I
        UD(MAX3,L2)=UDI(I)+TRANS
        UFIN(MAX3,L2)=UF(I-1)+TRANS
        SFIN(MAX3,L2)=SF(I-1)
   64 CONTINUE
      DO 65 I=NJ,NSUB
        L2=MAX3+I-NJ
        UD(L2,1)=UDI(I)+TRANS
        UFIN(L2,1)=UF(I)+TRANS
        SFIN(L2,1)=SF(I)
   65 CONTINUE
      IF (MAX3 .EQ. MAX4-1) GO TO 68
      J=2
      E=DCOST
      F=QCS(MAX3)
   67 CALL SUBOPT(MAX4,MAX3,J,E,F,CP,CS,P,QCS,QCP,TIME,SDEV)
      J=J+1
      IF (J .LE. NJ-1) GO TO 67
   68 MAX4=MAX3
      L=MAX4-1
      GO TO 66
C
C-----------------------------------------------------------
C DETERMINE TOTAL SYSTEM COST BY SIMULATION AND PRINT RESULTS.
C-----------------------------------------------------------
C
   75 BUF=0
      CALL SIMULA(UD,SDEV,QCP,QCS,TIME,NS,NJ,DCOST,BUF)
      E1=TOTAL
      PRINT 108
  108 FORMAT('1'////////)
      PRINT 80,NX
   80 FORMAT('0',21X,'EXPERIMENT NUMBER',I4/22X,21('='))
      PRINT 81,NJ
   81 FORMAT('0',22X,'NUMBER OF JOBS =',I3)
      PRINT 82,NS
   82 FORMAT('0',18X,'NUMBER OF STATIONS =',I3)
      PRINT 85
   85 FORMAT('0','D.P. HEURISTIC METHOD:'/1X,21('-'))
 1002 PRINT 83,EFS(1,NJ)
   83 FORMAT('0',28X,'MAKESPAN =',F9.2)
      PRINT 84,TOTAL
   84 FORMAT('0',19X,'TOTAL SYSTEM COST =',F9.2)
      IF (BUF .EQ. 1) GO TO 100
C
```

```
C-----------------------------------------------------------
C DETERMINE DECISONS AND SYSTEM COSTS USING BUFFER RULE.
C ALSO COMPUTE MINIMAL SYSTEM COST.
C-----------------------------------------------------------
C
      BUF=1
      CALL SIMULA(UD,SDEV,QCP,QCS,TIME,NS,NJ,DCOST,BUF)
      PRINT 92
   92 FORMAT('0'//,2X,'SAFETY BUFFER METHOD:')
      PRINT 96
   96 FORMAT(2X,20('-'))
      GO TO 1002
  100 PRINT 109
  109 FORMAT('0'//,5X,'SYSTEM PARAMETERS:')
      PRINT 93,(QCP(NS+1-I),I=1,NS)
   93 FORMAT ('0',5X,'QCP(I):',2X,10(F5.2,3X)/15X,10(F5.2,3X))
      PRINT 94,(QCS(NS+1-I),I=1,NS)
   94 FORMAT ('0',5X,'QCS(I):',2X,10(F5.2,3X)/15X,10(F5.2,3X))
      PRINT 95,(TIME(NS+1-I),I=1,NS)
   95 FORMAT ('0',4X,'TIME(I):',2X,10(F5.2,3X)/15X,10(F5.2,3X))
      PRINT 97,(SDEV(NS+1-I),I=1,NS)
   97 FORMAT ('0',4X,'SDEV(I):',2X,10(F5.2,3X)/15X,10(F5.2,3X))
      CHEP1=CHEP2=CHEP3=CHEAP=0
      DO 101 J=1,NJ
        CHEP1=CHEP1+(SRT(NS,J)-10)
        IF (MAX .EQ. 1) GO TO 101
        CHEP2=CHEP2+(QCS(MAX-1)*(SRT(MAX-1,J)-SRT(MAX,J)-TIME(MAX)))
        K=MAX-2
        IF (MAX .EQ. 2) GO TO 101
        DO 102 I=1,K
          CHEAP=CHEAP+(QCS(I)*(SRT(I,J)-SRT(I+1,J)-TIME(I+1)))
  102   CONTINUE
  101 CONTINUE
      CHEP3=(NJ*(NJ-1)*TIME(1)/2)
      CHEP1=CHEP1*QCS(NS)
      CHEP3=CHEP3*QF
      TCHEP=CHEP1+CHEP2+CHEP3+CHEAP+(DCOST*(SRT(1,NJ)+TIME(1)-10))
      PRINT 105,TCHEP
  105 FORMAT('0'//,18X,'MINIMAL SYSTEM COST =',F9.2)
      ECON=100*((E1-TCHEP)/(TOTAL-TCHEP))
      RPER=100*((E1-TCHEP)/TCHEP)
      PRINT 106,ECON
  106 FORMAT('0',16X,'COST DEFFICIENCY (%) =',F9.2)
      PRINT 107,RPER
  107 FORMAT('0',15X,'RANDOM COST RATIO (%) =',F9.2)
C
C-----------------------------
C PROGRAM TERMINATION TO OCCUR.
C-----------------------------
C
      NX=NX+1
      IF (NX .LE. 15) GO TO 1001
```

```
      STOP
      END
C
C
C
      SUBROUTINE ROUTE(J,NSUB,TRANS,MID)
C
C-----------------------------------------------------------------
C ADJUST AND ASSIGN DECISION DERIVED FROM OPTIMIZATION TO A LEFT
C SUB-BRANCH STARTING FROM STATION 'MID'.
C-----------------------------------------------------------------
C
      COMMON /A1/ SFIN(25,15),UFIN(25,15),UD(25,15),MAX,NJ
      COMMON /A2/ UDI(40),SD(40),UF(40),SF(40)
C
      DO 400 I=2,NSUB
        L=MID+I-1
        UD(L,J)=UDI(I)+TRANS
        UFIN(L,J)=UF(I-1)+TRANS
        SFIN(L,J)=SF(I-1)
  400 CONTINUE
      RETURN
      END
C
C
C
C
      SUBROUTINE DYNAM(NQ,CP,CS,P,SV)
C
C-----------------------------------------------------------------------
C PROGRAM TO DETERMINE THE OPTIMAL MEAN PART DUE DATES IN A MULTI STAGE
C SINGLE PATH ASSEMBLY LINE. OPTIMIZATION IS BASED ON CLARK'S EQUATIONS,
C WILHELM'S METHODOLGY, AND THE DEVELOPED REGRESSION MODEL.
C
C THE METHOD OF OPTIMIZATION IS BACKWARD RECURRSION STOCHASTIC DYNAMIC
C PROGRAMMING USING VARIANCES AND EXPECTATIONS. ALL INHERENT STOCHASTIC
C PROCESSES ARE ASSUMED TO BE NORMAL IN NATURE.
C
C STAGES ARE NUMBERED BACKWARDS.(I.E THE LAST STAGE IS NUMBER ONE)
C-----------------------------------------------------------------------
C
C VARIABLES
      REAL Q(40,2,40),K(40,9),CR(40,40,2),Z(40,2),CS(40),CP(40),SD
      REAL UD(40),UF,SF,CPS,OPTK(40),P(40)
      REAL PI
      INTEGER NQ,I,J,M,N,Y
      COMMON /A2/ UDI(40),SD(40),UF(40),SF(40)
      RHO(Z)=1/(EXP((Z**2)/2)*SQRT(2*3.1416))
C
C
```

```
      UF(NQ)=10
      SF(NQ)=SV
      C1=0.04577093
      C2=-0.47805193
      C3=-2.39312763
      C4=-0.03059004
      C5=2.35
      C6=0.02354503
      C7=-0.58349074
      PI=3.1415926
C
C-----------------------------------
C COMMENCE FIRST STAGE OPTIMIZATION.
C-----------------------------------
C
      CPS=CS(1)+CP(1)
      K(1,1)=CS(1)/CPS
      CALL MDNRIS(K(1,1),K(1,2),Y)
      OPTK(1)=K(1,2)
      K(1,3)=CPS*((1/(EXP((K(1,2)**2)/2)*SQRT(2*PI)))+(K(1,2)*K(1,1)))
      K(1,3)=K(1,3)-(CS(1)*K(1,2))
      K(1,4)=SD(1)
      CR(1,1,1)=K(1,3)
      CR(1,1,2)=K(1,4)
      N=1
      IF (N .EQ. NQ) GO TO 1000
C
C-------------------------------------------------------------
C DETERMINE RETURN EQUATION COEFFICIENTS FOR REMAINING STAGES.
C-------------------------------------------------------------
C
      DO 130 I=2,NQ
        CPS=CS(I)+CP(I)
        K(I,1)=CS(I)/CPS
        CALL MDNRIS(K(I,1),K(I,2),Y)
        K(I,3)=CPS*((1/(EXP((K(I,2)**2)/2)*SQRT(2*PI)))+(K(I,2)*K(I,1)))
        K(I,3)=K(I,3)-(CS(I)*K(I,2))
        K(I,4)=SD(I)
        SUMQ1=SUMQ2=0
        M=I-1
        DO 131 J=1,M
          Q(I,1,J)=C1*CR(I-1,J,1)*(SD(I)**C2)
          Q(I,2,J)=(C3*(SD(I)**C4))+C5+(C6*(CR(I-1,J,2)**C7))
          Q(I,2,J)=CR(I-1,J,1)*Q(I,2,J)
          SUMQ1=SUMQ1+Q(I,1,J)
          SUMQ2=SUMQ2+Q(I,2,J)
  131   CONTINUE
        K(I,5)=CPS-SUMQ1
        K(I,6)=CS(I)+SUMQ2
        K(I,7)=K(I,6)/K(I,5)
        IF (K(I,7) .GE. 0.98) K(I,7)=0.96
        IF (K(I,7) .LE. 0.02) K(I,7)=0.04
```

```
344          CALL MDNRIS(K(I,7),K(I,8),Y)
345          OPTK(I)=K(I,8)
346          K(I,9)=CPS*((1/(EXP((K(I,8)**2)/2)*SQRT(2*PI)))+(K(I,8)*K(I,7)))
347          K(I,9)=K(I,9)-(CS(I)*K(I,8))
348          CR(I,1,1)=K(I,9)
349          CR(I,1,2)=K(I,4)
350          R=K(I,7)
351          U=RHO(K(I,8))
352          Z(I,1)=K(I,8)*((K(I,8)*(R-(R**2)))+(U*(1-(2*R))))
353          Z(I,1)=Z(I,1)+R-(U**2)
354          Z(I,2)=(Z(I,1)+(1-(2*R)))*(SD(I)**2)
355          DO 132 J=2,I
356            CR(I,J,1)=CR(I-1,J-1,1)*SQRT(Z(I,1))
357            CR(I,J,2)=SQRT((Z(I,2)+(CR(I-1,J-1,2)**2))/Z(I,1))
358    132   CONTINUE
359    130 CONTINUE
      C
      C-----------------------------------------
      C COMPUTE OPTIMAL SET OF MATERIAL DUE DATES.
      C-----------------------------------------
      C
360        NQ1=NQ-1
361        DO 140 M=1,NQ1
362          I=NQ+1-M
363          VAR=SQRT((SD(I)**2)+(SF(I)**2))
364          UDI(I)=UF(I)-(OPTK(I)*VAR)
365          ALPHA=(UF(I)-UDI(I))/VAR
366          CALL MDNOR(ALPHA,PHI)
367          PSI=RHO(ALPHA)
368          UF(I-1)=(UF(I)*PHI)+(UDI(I)*(1-PHI))+(VAR*PSI)+P(I)
369          US12=(UF(I)**2)+(SF(I)**2)
370          US22=(UDI(I)**2)+(SD(I)**2)
371          SECM=(US12*PHI)+(US22*(1-PHI))+((UF(I)+UDI(I))*VAR*PSI)
372          ARG=SECM-((UF(I-1)-P(I))**2)
373          IF (ARG .LT. 0) ARG=(-1)*ARG
374          SF(I-1)=SQRT(ARG)
375          UD(I)=UF(I)-(K(I,2)*VAR)
376    140 CONTINUE
377   1000 VAR=SQRT((SD(1)**2)+(SF(1)**2))
378        UDI(1)=UF(1)-(OPTK(1)*VAR)
379        UD(1)=UDI(1)
      C
380        RETURN
381        END
      C
      C
      C
      C
      C
382        SUBROUTINE CLARK(S1,E1,S2,E2,STD,FIN)
      C
      C-----------------------------------------------------------------
```

```
C THIS SUBROUTINE USES CLARKS METHOD TO DETERMINE THE EXPECTATION AND
C VARIANCE OF THE OPERATION START FINISH TIMES AT A STATION ASSUMING
C THERE IS NO CORRELATION.
C-----------------------------------------------------------------------
C
C VARIABLES
      REAL E4,S4,V2,Z,A,B,C
C
C STATEMENT FUNCTIONS USED
C
      RHO(Z)=1/(EXP((Z**2)/2)*SQRT(2*3.1416))
C
      A=SQRT ((S1**2)+(S2**2))
      B=(E1-E2)/A
      CALL MDNOR(B,C)
      Q=(B**2)/2
      S=(((E1**2)+(S1**2))*C)+(((E2**2)+(S2**2))*(1-C))
      IF (Q .GT. 174.0) GO TO 270
      E4=(E1*C)+(E2*(1-C))+(A*RHO(B))
      V2=S+((E1+E2)*A*RHO(B))
      GO TO 271
  270 E4=(E1*C)+(E2*(1-C))
      V2=S
  271 S4=SQRT(V2-(E4**2))
C
      FIN=E4
      STD=S4
      RETURN
      END
C
C

      SUBROUTINE SUBOPT(M1,M2,J,E,F,CP,CS,P,QCS,QCP,TIME,SDEV)
C
C-----------------------------------------------------------------------
C THIS SUBROUTINE UTILIZES THE SUBROUTINE DYNAM TO OPTIMIZE THE NETWORK
C ELEMENT BETWEEN STAGES M1 ANS M2. ROUTE IS ALSO CALLED.
C-----------------------------------------------------------------------
C
      REAL QCS(25),QCP(25),TIME(25),CP(40),CS(40),SD,SDEV(25)
      REAL P(40)
      COMMON /A1/ SFIN(25,15),UFIN(25,15),UD(25,15),MAX,NJ
      COMMON /A2/ UDI(40),SD(40),UF(40),SF(40)
C
      NMID=M1-M2
      DO 300 I=2,NMID
        CP(I)=QCP(M2+I-1)
        CS(I)=QCS(M2+I-1)
        SD(I)=SDEV(M2+I-1)
        P(I)=TIME(M2+I-1)
  300 CONTINUE
      CP(1)=E
      CS(1)=F
```

```
      CALL CLARK(SFIN(M2,J-1),UFIN(M2,J-1),SDEV(M2),UD(M2,J),
     1SCOM,UCOM)
      SD(1)=SCOM
      SB=SFIN(M1,J)
      CALL DYNAM(NMID,CP,CS,P,SB)
      TRANS1=UFIN(M1,J)-10
      TRANS2=UDI(1)+TRANS1-UCOM
      IF (TRANS2 .LE. 0) THEN
        TRANS=UCOM-UDI(1)
      ELSE
        TRANS=TRANS1
      ENDIF
      CALL ROUTE(J,NMID,TRANS,M2)
C
  310 RETURN
      END
C
C
C
C

      SUBROUTINE ROUTE2(M2,J,L,N1,TRANS)
C
C-----------------------------------------------------------
C ADJUST AND ASSIGN DECISION DERIVED FROM OPTIMIZATION TO A
C VERTICAL SUB-BRANCH LOCATED AT STATION M2.
C-----------------------------------------------------------
C
      COMMON /A1/ SFIN(25,15),UFIN(25,15),UD(25,15),MAX,NJ
      COMMON /A2/ UDI(40),SD(40),UF(40),SF(40)
C
      DO 500 I=1,N1
        UD(M2,J+1-I)=UDI(L+I)+TRANS
        UFIN(M2,J+1-I)=UF(L+I-1)+TRANS
        SFIN(M2,J+1-I)=SF(L+I-1)
  500 CONTINUE
C
      RETURN
      END
C
C
C

      SUBROUTINE SIMULA(UD,SDEV,QCP,QCS,TIME,NS,NJ,DCOST,BUF)
C
C-------------------------------------------------------------------
C PROGRAM TO SIMULATE THE FLOW OF PRODUCTS THROUGH A MULTI-STAGE
C MULTI-PRODUCT ASSEMBLY LINE WITH RANDOM DELIVERY OF ASSEMBLY COMPONENTS
C-------------------------------------------------------------------
C
C VARIABLES
      REAL TIME(25),SDEV(25),UD(25,15),EFS,SSF
      REAL SRET(25,15),QCP(25),QCS(25),DCOST,RN(375),QCT(25)
      REAL FIN,A(25,15),F(26,15),SUMF(25,15),RET,START
```

```
      INTEGER NS,NJ,K,I,J,N,BUF,NGEN
      DOUBLE PRECISION DSEED
      COMMON /A3/ RET(25,15),EFS(26,15),TOTAL,TDELV
      COMMON /A4/ TARD,WARH,TCOST,WCOST,QF,QCOST,COST
      COMMON /A5/ DSEED,SRT(25,15)
C
C-----------------------------------------------------------------
C CALCULATE EXPECTATION AND VARIANCE OF ASSEMBLY PART DELIVERY
C DATE FOR EACH STAGE.
C-----------------------------------------------------------------
C
      DO 701 I=1,NS
        DO 702 J=1,NJ
          SRET(I,J)=0.0
          SUMF(I,J)=0.0
  702   CONTINUE
  701 CONTINUE
      EFS(26,1)=10.0
      SSF=0.10
      DO 711 J=1,NJ
        EFS(26,J)=EFS(26,1)
        QCT(J)=0.0
  711 CONTINUE
      PI=3.1416
      N=0
      TOTAL=0
      COST=QCOST=TCOST=WCOST=0
      IF (BUF .EQ. 0) GO TO 7001
      EFS(NS,1)=EFS(26,1)+TIME(NS)
      UD(NS,1)=EFS(NS,1)-(BUF*SDEV(NS))
      DO 720 J=2,NJ
        EFS(NS,J)=EFS(NS,J-1)+TIME(NS)
        UD(NS,J)=EFS(NS,J)-(BUF*SDEV(NS))
  720 CONTINUE
      DO 721 K=2,NS
        I=NS+1-K
        EFS(I,1)=EFS(I+1,1)+TIME(I)
        UD(I,1)=EFS(I,1)-(BUF*SDEV(I))
        DO 722 J=2,NJ
          START=AMAX1(EFS(I,J-1),EFS(I+1,J))
          EFS(I,J)=START+TIME(I)
          UD(I,J)=EFS(I,J)-(BUF*SDEV(I))
  722   CONTINUE
  721 CONTINUE
      TDELV=EFS(1,NJ)+(BUF*SDEV(1))
C
C-------------------------------------------------
C GENERATE NORMAL RANDOM DEVIATES FOR PART DELIVERY.
C-------------------------------------------------
C
 7001 NGEN=(NS*NJ)+NJ
 7000 CALL GGNML(DSEED,NGEN,RN)
```

```
      DO 725 J=1,NJ
        DO 726 I=1,NS
          A(I,J)=UD(I,J)+(SDEV(I)*RN(J+(NJ*I)))
  726   CONTINUE
        F(26,J)=EFS(26,J)+(SSF*RN(J))
  725 CONTINUE
C
C-----------------------------------------------
C SIMULATION OF JOB FLOW THROUGH ASSEMBLY STAGES
C-----------------------------------------------
C
      F(NS,1)=AMAX1(A(NS,1),F(26,1))+TIME(NS)
      COST1=QCS(NS)*(F(NS,1)-TIME(NS)-F(26,1))
      COST2=QCP(NS)*(F(NS,1)-TIME(NS)-A(NS,1))
      SRET(NS,1)=SRET(NS,1)+COST1+COST2
      SUMF(NS,1)=SUMF(NS,1)+F(NS,1)
      DO 730 K=2,NS
        I=NS+1-K
        F(I,1)=AMAX1(A(I,1),F(I+1,1))+TIME(I)
        COST1=QCS(I)*(F(I,1)-TIME(I)-F(I+1,1))
        COST2=QCP(I)*(F(I,1)-TIME(I)-A(I,1))
        SRET(I,1)=SRET(I,1)+COST1+COST2
        SUMF(I,1)=SUMF(I,1)+F(I,1)
  730 CONTINUE
      DO 731 J=2,NJ
        F(NS,J)=AMAX1(A(NS,J),F(26,J),F(NS,J-1))+TIME(NS)
        COST1=QCS(NS)*(F(NS,J)-TIME(NS)-F(26,J))
        COST2=QCP(NS)*(F(NS,J)-TIME(NS)-A(NS,J))
        SRET(NS,J)=SRET(NS,J)+COST1+COST2
        SUMF(NS,J)=SUMF(NS,J)+F(NS,J)
        DO 732 K=2,NS
          I=NS+1-K
          F(I,J)=AMAX1(A(I,J),F(I+1,J),F(I,J-1))+TIME(I)
          COST1=QCS(I)*(F(I,J)-TIME(I)-F(I+1,J))
          COST2=QCP(I)*(F(I,J)-TIME(I)-A(I,J))
          SRET(I,J)=SRET(I,J)+COST1+COST2
          SUMF(I,J)=SUMF(I,J)+F(I,J)
  732   CONTINUE
  731 CONTINUE
      FIN=F(1,NJ)
      L=NJ-1
      DO 733 J=1,L
        QCT(J)=QCT(J)+(QF*(FIN-F(1,J)))
  733 CONTINUE
      IF (FIN .GT. TDELV) TCOST=TCOST+(TARD*(FIN-TDELV))
      IF (TDELV .GT. FIN) WCOST=WCOST+(WARH*(TDELV-FIN))
      COST=COST+(DCOST*(FIN-10))
      N=N+1
      IF (N .LT. 80) GO TO 7000
C
C-----------------------------------------
C COMPUTE EXPECTED VALUES OF FINISH TIMES
```

```
C-----------------------------------------------
C
      DO 750 J=1,NJ
        DO 751 I=1,NS
          EFS(I,J)=SUMF(I,J)/N
          RET(I,J)=SRET(I,J)/N
          TOTAL=TOTAL+RET(I,J)
  751   CONTINUE
        QCOST=QCOST+QCT(J)
  750 CONTINUE
      TCOST=TCOST/N
      WCOST=WCOST/N
      QCOST=QCOST/N
      COST=COST/N
      TOTAL=COST+TOTAL+TCOST+WCOST+QCOST
C
C-------------------------
C PROGRAM CONTROL TO MAIN.
C-------------------------
C
      RETURN
      END
C
C
C

      SUBROUTINE DATA(TIME,QCP,QCS,SDEV,DCOST,TARD,WARH,QF,NS,NJ)
C
C---------------------------------------------
C PROGRAM TO RANDOMLY GENERATE A TEST PROBLEM.
C---------------------------------------------
C
      REAL QCP(25),QCS(25),TIME(25),SDEV(25),R(100)
      DOUBLE PRECISION DSEED
      COMMON /A5/ DSEED,SRT(25,15)
C
      CALL GGUBS(DSEED,100,R)
      NS=IFIX(R(91)*20)+5
      NJ=IFIX(5+(R(92)*10))
      CUM=4*R(93)
      DO 900 I=1,NS
        K=NS+1-I
        TIME(I)=10*R(I)
        QCP(K)=3*R(30+I)
        QCS(K)=CUM
        CUM=CUM+(0.7*QCP(K))
        SDEV(I)=0.1+(2.0*R(60+I))
  900 CONTINUE
      TARD=12*R(94)
      WARH=9*R(95)
      DCOST=6*R(96)
      QF=CUM
C
```

```
      RETURN
      END
C
C
C
//DATA
```

APPENDIX C

Safety Buffer Scheduling Method

The safety buffer scheduling method prescribes a decision policy for the ordering of parts from unreliable vendors. The strategy involves, coinciding the historical mean delivery date of the part with the date of its requirement. A safety lead time is then subtracted from this date to obtain the prescribed delivery date.

Wilhelm and Ahmadi-Marandi (1982) applied this method to their descriptive model of the assembly line. They made the safety lead time a function of the standard deviation of part arrival time. Thus,

$$\mu D_{ij} = V_{ij} - B\sigma D_{ij}$$

Where V_{ij} is a deterministic estimate of the earliest possible start time of processing at stage i_j. The value of B defines the risk of schedule deviations. The probabilities that a part will be delivered on or before its due date are 0.5, 0.8413, and 0.9722, respectively, for values of K equal to 1,1, and 2. Clearly, the safety buffer will reduce the queueing of subassemblies, but component queue time will increase.

Selecting a suitable value of K is difficult. Walker and Wysk (1983) experimented with different values of K, to de-

termine purchased part lead time in an MRP system. From the results of their simulation they concluded a strategy of K=1 was best. A major drawback in this method is that it considers all stages to be equally important, which is usually not the case.

APPENDIX D

Details of Random Test Experiments

EXPERIMENT NUMBER 4

NUMBER OF JOBS = 12

NUMBER OF STATIONS = 13

D.P. HEURISTIC METHOD:

MAKESPAN = 184.33

TOTAL SYSTEM COST = 14513.50

SAFETY BUFFER METHOD:

MAKESPAN = 190.24

TOTAL SYSTEM COST = 17684.45

SYSTEM PARAMETERS:

QCP(I):	0.02	1.75	0.97	2.63	0.75	0.85	2.42	2.41	2.54	1.08
	0.93	2.86	2.63							
QCS(I):	3.66	3.67	4.89	5.57	7.41	7.94	8.53	10.23	11.91	13.69
	14.45	15.09	17.09							
TIME(I):	1.22	0.49	7.51	9.31	7.56	1.95	9.60	2.79	2.68	1.64
	8.94	9.48	0.37							
SDEV(I):	0.46	0.87	1.14	1.27	1.89	0.13	2.08	0.68	1.39	2.08
	1.27	0.84	1.99							

MINIMAL SYSTEM COST = 13348.08

COST DEFFICIENCY (%) = 26.88

RANDOM COST RATIO (%) = 8.73

EXPERIMENT NUMBER 7

NUMBER OF JOBS = 7

NUMBER OF STATIONS = 10

D.P. HEURISTIC METHOD:

MAKESPAN = 94.34

TOTAL SYSTEM COST = 3056.03

SAFETY BUFFER METHOD:

MAKESPAN = 99.87

TOTAL SYSTEM COST = 4195.48

SYSTEM PARAMETERS:

QCP(I):	2.27	1.76	2.67	1.86	0.13	1.16	0.33	1.25	2.11	0.65
QCS(I):	2.41	4.00	5.24	7.11	8.41	8.50	9.31	9.54	10.42	11.90
TIME(I):	3.05	1.61	1.00	7.94	5.46	1.09	2.13	1.76	5.55	4.40
SDEV(I):	1.41	1.24	1.84	0.73	1.97	0.62	0.68	1.91	0.91	1.24

MINIMAL SYSTEM COST = 2712.69

COST DEFFICIENCY (%) = 23.16

RANDOM COST RATIO (%) = 12.66

EXPERIMENT NUMBER 13

NUMBER OF JOBS = 14

NUMBER OF STATIONS = 17

D.P. HEURISTIC METHOD:

MAKESPAN = 235.89

TOTAL SYSTEM COST = 23494.89

SAFETY BUFFER METHOD:

MAKESPAN = 243.00

TOTAL SYSTEM COST = 29236.35

SYSTEM PARAMETERS:

QCP(I):	2.90	2.70	0.96	1.39	1.20	0.01	2.13	1.77	0.61	1.12
	0.89	2.57	2.61	1.61	1.10	0.19	2.62			
QCS(I):	3.14	5.17	7.06	7.72	8.70	9.54	9.55	11.04	12.28	
	13.49	14.11	15.91	17.74	18.87	19.64	19.77			
TIME(I):	4.84	5.39	6.52	3.28	7.59	5.16	0.95	5.89	9.97	2.30
	8.73	1.34	9.49	8.47	0.90	6.83	5.59			
SDEV(I):	1.43	0.79	0.17	1.37	0.76	1.16	1.75	1.61	0.21	0.71
	1.33	0.80	1.22	1.36	0.90	0.17	0.62			

MINIMAL SYSTEM COST = 22275.82

COST DEFFICIENCY (%) = 17.51

RANDOM COST RATIO (%) = 5.47

EXPERIMENT NUMBER 17

NUMBER OF JOBS = 5

NUMBER OF STATIONS = 18

D.P. HEURISTIC METHOD:

MAKESPAN = 138.66

TOTAL SYSTEM COST = 2689.50

SAFETY BUFFER METHOD:

MAKESPAN = 144.21

TOTAL SYSTEM COST = 3774.97

SYSTEM PARAMETERS:

QCP(I):	2.74	2.49	1.79	1.84	0.27	0.32	1.42	0.44	1.08	1.61
	1.88	1.61	0.67	0.34	0.84	2.93	2.21	1.84		
QCS(I):	2.40	4.32	6.06	7.31	8.60	8.78	9.01	10.01	10.31	11.07
	12.20	13.51	14.64	15.11	15.35	15.94	17.99	19.54		
TIME(I):	1.83	2.99	1.53	5.75	2.47	8.46	4.90	5.88	9.39	0.57
	6.10	9.20	3.66	6.74	6.81	4.34	0.11	6.04		
SDEV(I):	0.67	0.37	0.50	0.73	0.94	1.26	1.44	0.89	1.58	1.95
	0.43	0.98	1.19	1.42	1.92	2.04	1.31	1.71		

MINIMAL SYSTEM COST = 2196.17

COST DEFFICIENCY (%) = 31.25

RANDOM COST RATIO (%) = 22.46

EXPERIMENT NUMBER 19

NUMBER OF JOBS = 12

NUMBER OF STATIONS = 20

D.P. HEURISTIC METHOD:

MAKESPAN = 207.78

TOTAL SYSTEM COST = 14402.56

SAFETY BUFFER METHOD:

MAKESPAN = 211.51

TOTAL SYSTEM COST = 16540.14

SYSTEM PARAMETERS:

QCP(I):	0.60	2.96	1.34	2.44	2.04	1.75	0.86	2.67	0.82	0.21
	0.80	0.36	0.85	2.86	0.17	2.53	1.43	0.19	0.96	0.24
QCS(I):	1.01	1.43	3.50	4.44	6.15	7.58	8.80	9.40	11.27	11.84
	11.99	12.55	12.80	13.40	15.40	15.52	17.29	18.29	18.42	19.09
TIME(I):	8.69	2.13	0.27	2.96	3.33	1.67	8.95	1.36	5.74	0.01
	9.47	3.90	3.28	9.36	6.34	0.09	8.25	7.92	1.90	1.53
SDEV(I):	0.84	1.43	0.31	1.69	1.23	0.46	1.59	0.15	1.88	1.74
	0.75	1.63	1.64	1.53	0.60	0.38	1.60	1.96	1.20	0.17

MINIMAL SYSTEM COST = 12832.12

COST DEFFICIENCY (%) = 42.35

RANDOM COST RATIO (%) = 12.24

EXPERIMENT NUMBER 22

NUMBER OF JOBS = 14

NUMBER OF STATIONS = 16

D.P. HEURISTIC METHOD:

MAKESPAN = 179.60

TOTAL SYSTEM COST = 16543.55

SAFETY BUFFER METHOD:

MAKESPAN = 184.43

TOTAL SYSTEM COST = 25230.13

SYSTEM PARAMETERS:

QCP(I):	1.80	2.98	0.07	2.09	1.35	1.70	0.33	1.53	2.46	1.69
	2.36	2.56	0.27	2.60	2.15	2.96				
QCS(I):	1.79	3.05	5.13	5.18	6.64	7.59	8.77	9.01	10.08	11.80
	12.98	14.63	16.42	16.61	18.43	19.94				
TIME(I):	0.91	4.05	2.45	9.06	2.10	0.95	0.00	2.92	4.60	3.42
	1.14	6.95	1.72	3.08	3.82	0.23				
SDEV(I):	1.30	1.75	1.31	0.28	2.04	1.91	0.55	1.81	0.13	1.80
	1.86	1.10	1.87	0.82	0.21	1.16				

MINIMAL SYSTEM COST = 15059.55

COST DEFFICIENCY (%) = 14.59

RANDOM COST RATIO (%) = 9.85

EXPERIMENT NUMBER 24

NUMBER OF JOBS = 11

NUMBER OF STATIONS = 19

D.P. HEURISTIC METHOD:

MAKESPAN = 190.46

TOTAL SYSTEM COST = 13580.80

SAFETY BUFFER METHOD:

MAKESPAN = 190.85

TOTAL SYSTEM COST = 15317.23

SYSTEM PARAMETERS:

QCP(I):	1.78	2.30	2.00	0.70	1.47	0.47	1.72	1.97	2.44	0.95
	2.13	2.96	2.04	0.16	0.85	2.15	2.48	2.51	2.72	
QCS(I):	0.91	2.16	3.77	5.17	5.66	6.69	7.02	8.23	9.60	11.31
	11.98	13.47	15.54	16.97	17.08	17.68	19.18	20.92	22.68	
TIME(I):	2.08	5.48	0.21	7.35	6.30	6.91	7.21	6.55	3.62	1.24
	5.86	4.31	7.85	3.17	1.77	6.77	4.35	0.11	8.28	
SDEV(I):	1.78	1.15	0.13	1.13	0.80	0.82	0.34	0.70	1.66	1.36
	1.57	1.57	1.59	0.54	1.96	0.49	0.33	2.07	0.30	

MINIMAL SYSTEM COST = 11914.00

COST DEFFICIENCY (%) = 48.98

RANDOM COST RATIO (%) = 13.99

EXPERIMENT NUMBER 27

NUMBER OF JOBS = 7

NUMBER OF STATIONS = 5

D.P. HEURISTIC METHOD:

MAKESPAN = 90.86

TOTAL SYSTEM COST = 1433.15

SAFETY BUFFER METHOD:

MAKESPAN = 99.25

TOTAL SYSTEM COST = 2141.07

SYSTEM PARAMETERS:

QCP(I):	1.23	1.38	1.73	0.69	1.90
QCS(I):	0.71	1.57	2.53	3.74	4.23
TIME(I):	9.04	4.66	3.10	5.51	3.00
SDEV(I):	1.60	1.51	1.78	0.44	0.87

MINIMAL SYSTEM COST = 1292.54

COST DEFFICIENCY (%) = 16.57

RANDOM COST RATIO (%) = 10.88

REFERENCES

1. Altiok T., and S.Stidham (1983), "The Allocation of Interstage Buffer Capacities in Production Lines." IIE Trans., 15, 4, pp. 292-299.

2. Bigham P.E., and J.M.Mogg (1981), "Integer Multiple Schedules for a Multi-Stage System with Stochastic Lead Times." Int. J. Prod. Res., Vol.19, 1, pp.97-107.

3. Buzacott J.A. (1967), "Automatic Transfer Production Lines With Buffer stocks." Int. J. Prod. Res., Vol.5, 3, pp. 183-200.

4. Clark E.C. (1961), "The Greatest of a Finite Set of Random Variables." Opns. Res. (March-April).

5. Cooper L. and M.W.Cooper (1981), Introduction to Dynamic Programming, Pergamon International Library.

6. Davis A.D. (1984), Material Management Decision Support Model. Unpublished Master's Thesis, Virginia Polytechnic Institute, Blacksburg.

7. Elmaghraby S.E. (1977), Activity Networks, John Wiley, New York.

8. Emshoff J.R., and R.L.Sisson (1970), Design and Use of Computer Simulation Models, Macmillian Publishing Co., New York.

9. Fox K.A. (1984), "MRP II: A Natural Hub For a CIM System." Industrial Eng., Vol. 16, Oct.,

10. Greer W.R., and G.J.LaCava (1979), "Normal Approximation of the Gretest of Two Normal Random Variables." Omega, 7, 4, pp. 361-363.

11. Jensen P.A., and M.A.Khan (1972), "Scheduling of a Multi-Stage Production System With Setup and Inventory Costs." IIE Trans., 4, 3, pp. 126-133.

12. Johnson N.L., and S.Kotz (1970), Continous Univariate Distributions - 1, John Wiley, New York.

13. Kraemer S.A., and R.F.Love (1970), "A Model For Optimizing the Buffer Inventory Storage Size in a Sequential Production System." IIE Trans., 2, 1, pp. 64-69.

14. Harrison J.M. (1973), "Assembly Like Queues." Journal of Applied Prob.,Vol.10, pp. 345-367.

15. Mize J.H. (1979), "Simulation Modelling." Procs. IIE Fall Conf.

16. Muth E.J. (1973), "The Production Rate of a Series of Workstations With Variable Sevice Times." Int. Journal of Prod. Res., Vol.14, 4, pp. 437-443.

17. Pearson C.E. (1983), Handbook of Applied Mathematics, Van Nostrand Reinhold Company.

18. Rao S.S. (1979), Optimization Theory, Tata McGraw-Hill, New Delhi, pp. 609-620.

19. SAS User's Guide: Basics (1982), SAS Institute Inc., Cary, Nort Carolina.

20. SAS User's Guide: Statistics (1982), SAS Institute Inc., Cary, North Carolina.

21. Suresh S. (1984), Stochastic Flowshop Scheduling, Unpublished Master's Thesis, Virginia Polytechnic Institute, Blacksburg.

22. Tippet L.H.C. (1925), "On the Extreme Individuals and the Range of Samples Taken From a Normal Population." Biometrika, Vol.17, pp. 364-387.

23. Walker D.M., and R.A.Wysk (1983), "A Procedure to Determine the Least Cost Purchased Part Planned Lead-Time for Items Controlled by an MRP System." Comput. & Indus. Engng., 7, 4, pp. 295-308.

24. Wight O.W. (1984), MRP II: Unlocking America's Productivity Potential. CBI Publishing Co., Boston.

25. Wilhelm W.E., and S.Ahmadi-Marandi (1982), "A Methodology to Describe Operating Characteristics of Assembly Systems." IIE Trans., 14, 3, pp. 204-213.

26. Wihelm W.E., and R.A.Johnson (1983), "Extensions to An Approach For Modelling The Transient Performance of Stochastic Assembly Systems." Working Paper , Ohio State University, Columbus.

The vita has been removed from the scanned document