Performance of Hierarchical Production Scheduling Policy

RAMAKRISHNA AKELLA, YONG CHOONG, AND STANLEY B. GERSHWIN

Abstract—**The performance of Kimemia and Gershwin's hierarchical scheduling scheme for flexible manufacturing systems, as enhanced by Gershwin, Akella, and Choong, is described. This method calculates times at which to dispatch parts into a system in a way that limits the disruptive effects of such disturbances as machine failures. Simulation results based on a detailed model of an IBM printed circuit card assembly facility are presented. Comparisons are made with other policies and the hierarchical policy is shown to be superior.**

I. INTRODUCTION

IN THIS REPORT we discuss the performance of the hierarchical production scheduling policy of Kimemia [4] and Kimemia and Gershwin [5] as it has been enhanced by Gershwin, Akella, and Choong [2]. We use a detailed simulation of an automated printed circuit card assembly line at the International Business Machines (IBM) Corporation plant at Tucson, Arizona as an experimental test bed.

We compare this with other policies for loading parts into a flexible manufacturing system. We demonstrate that the hierarchical strategy is effective in meeting production requirements (both total volume and balance among part types) while limiting average work-in-process (WIP). The purpose of this policy is to respond to disruptive events that occur as part of the production process, particularly repairs and failures. Simulation experiments described here show that the hierarchical policy allows the system to run relatively smoothly in spite of such events.

Flexible Manufacturing Systems

A flexible manufacturing system (FMS) typically consists of several production machines and associated storage elements, connected by an automated transportation system. The entire system is automatically operated by a network of computers. The purpose of the flexibility and versatility of the configuration is to meet production targets for a variety of part types in the face of disruptions such as demand variations and machine failures.

The IBM Automated Circuit Card Line is an automated assembly system for producing a variety of printed circuit cards. Workholders containing the cards move through the system from machine to machine along transportation elements which are controlled by a hierarchy of computers and microprocessors. At each of these machines electronic components are inserted into the card. Each type of card goes to a specific set of machines. The processing time of each card at any machine depends on the number and type of components that are inserted. If a machine is busy or otherwise unavailable, the workholders are stored in a buffer near the machine.

In an FMS, individual part movements are practical because of the automated transportation system. The time required to change a machine from doing one operation to doing another (the setup or changeover time) is small in comparison with operation times. These features enable the FMS to rapidly redistribute its capacity between different parts. This flexibility enables the FMS to react to potentially disruptive events such as machine failures and changes in demand.

FMS's are useful when 1) a number of related part types require operations at different machines of the same line; 2) different part types go to the same machines, but require different operations; 3) different part types go to some common machines and then to different machines; and 4) the required part-mix varies with time.

All production systems are subject to disruptive events ranging from sudden changes in demand to machine failures. Their times of occurrence cannot be predicted in advance; at best, a historical record can only provide guidelines on when they can be expected. A scheduling policy must provide for these factors. The purpose of the hierarchical policy described in this paper is to efficiently use the available information and system flexibility to anticipate and to react to disruptive events.

Hierarchical Scheduling Policy

Fig. 1 outlines the hierarchical structure of the scheduling policy. The middle level is the heart of the scheduler. It determines the short-term production rates, taking the capacity constraints of the system into account. Based on these rates the lower level determines the actual times at which parts are loaded into the system. The middle level uses machine status information and deviation from demand for its computations. It also needs certain longer term information. This is supplied by the higher level. It is computed from machine data such as failure and repair rate information, and part data such as operation times and demand.

The concept of capacity is crucial to the design of the hierarchical policy. The capacity at any instant depends on the operational states of the machines. It changes as machines fail or are repaired.

The hierarchical structure of the policy reflects the disci-

Manuscript received September 1984. This work was supported by the Manufacturing Research Center of the Thomas J. Watson Research Laboratory of the International Business Machines Corporation and by the U.S. Army Human Engineering Laboratory under Contract DAAK11-82-K-0018.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139.

0148-6411/84/0900-0225$1.00 © 1984 IEEE

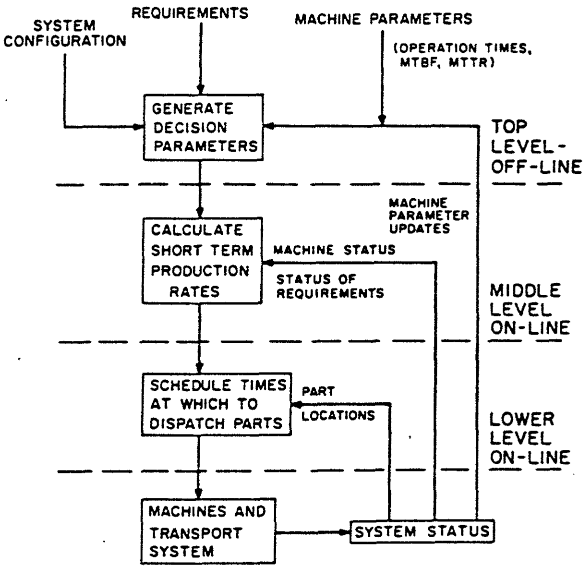
SYSTEM CONFIGURATION
REQUIREMENTS
MACHINE PARAMETERS
(OPERATION TIMES, MTBF, MTTR)
GENERATE DECISION PARAMETERS
TOP LEVEL-OFF-LINE
MACHINE PARAMETER UPDATES
CALCULATE SHORT TERM PRODUCTION RATES
MACHINE STATUS
STATUS OF REQUIREMENTS
MIDDLE LEVEL ON-LINE
SCHEDULE TIMES AT WHICH TO DISPATCH PARTS
PART LOCATIONS
LOWER LEVEL ON-LINE
MACHINES AND TRANSPORT SYSTEM
SYSTEM STATUS

Fig. 1. Hierarchical approach to production planning.

pline that must be imposed in scheduling the FMS. If parts are loaded into the system at a rate that violates the capacity constraints, poor performance results. Material accumulates in buffers or in the transportation system, causing congestion and preventing other material from getting to the machines. Not only does the system perform below expectations, but its effective capacity is reduced.

The hierarchical policy is based on the capacity discipline. Parts are loaded into the system at rates that are within the current capacity, which is determined by the current set of operational machines. This prevents congestion from ever occurring.

In the next section we briefly describe the IBM system. In Section III we describe scheduling objectives and performance measures. The hierarchical policy and some common sense policies are described in Sections IV and V, respectively. In Section VI we compare and discuss the results, and we conclude in Section VII.

II. THE IBM AUTOMATED CARD ASSEMBLY LINE

In this section we describe a system to which the hierarchical scheduler is applicable. Our purpose in using this system is to assess the scheduler in a realistic setting.

Purpose of System

At IBM's General Products Division at Tucson, an automated card assembly line is being built up in stages, through a series of "minilines." The portion of the system of interest to us is the stage consisting of insertion machines. Printed circuit cards from a storage area upstream arrive at the loading area of the insertion stage. Each card is placed in a workholder, which is introduced into the system. It goes to the machines where the electronic components it requires are inserted. It then exits the system and goes to the downstream stages, which consist of testing and soldering machines.

There are several types of insertion machines, each of which inserts one mechanically distinct type of component. The common ones are single in-line package inserters (SIP's), dual in-line package inserters (DIP's), multiform modular inserters (MODI's) and variable center distance inserters (VCD's). By loading different components, the line can be used to assemble a variety of cards.

In order to concentrate on the operational issues of the FMS, we assume that component loading has already been determined. The changeover time is small among the family of parts producible with a given component loading. We also restrict our attention to the Miniline 1300, whose schematic is shown in Fig. 2. This consists of a DIP, a VCD, and two SIP's. Each of the machines also has an associated buffer, which can hold 30 parts.

Transportation System

The workholders are loaded at input station 301 and then move to each of the required machines. Movement is along straight or rotating elements. The straight elements are used to move parts in a single fixed direction and are represented by rectangles. The rotating elements are for 90 deg turns and are represented by circles. Representative movement times are listed in Table I.

Movement of cards in the vicinity of a work station (insertion machine, associated buffer, and transport elements) follows a common pattern. Cards arrive at a rotating element like 603 and either turn towards the insertion machines, or move straight on. The cards going to machines (e.g., 101) either wait at input elements like 605, or go into buffers like 201. After all the required components have been inserted, a similar movement takes the card out of the insertion machines and onto output element 305. After element 606 is rotated toward the work station, the card is placed on it. Element 606 is rotated back to its original position and the card is then loaded onto the next transportation element (306). Finally, after going through the entire system, the cards exit from output element 324.

Machine Parameters and Part Data

The mean time between failures (MTBF) and mean time to repair (MTTR) of the machines are listed in Table II. The average fraction of time a machine is available is the time a machine is available for production divided by the total time. This quantity, called the efficiency or availability of a machine, is also listed in Table II.

There are other random perturbations affecting the system. These include machine tool jams, which occur when a machine jams in trying to insert a component. Rather than regard this as a failure, this small but regular disturbance (approximately once every 100 insertions) can be modeled as part of the processing time.

Normally there are several part (card) types being processed in the system. We limit our experiment to only six types to better examine the hierarchical policy. Typical demand rates are listed in Table III. Also shown in Table III are the operation times required by each card type at each of the machines. These include the processing time and the time to move in and out of each machine.

Loading

Loading describes how heavily the machines in a system must be utilized to satisfy demand. The expected utilization of

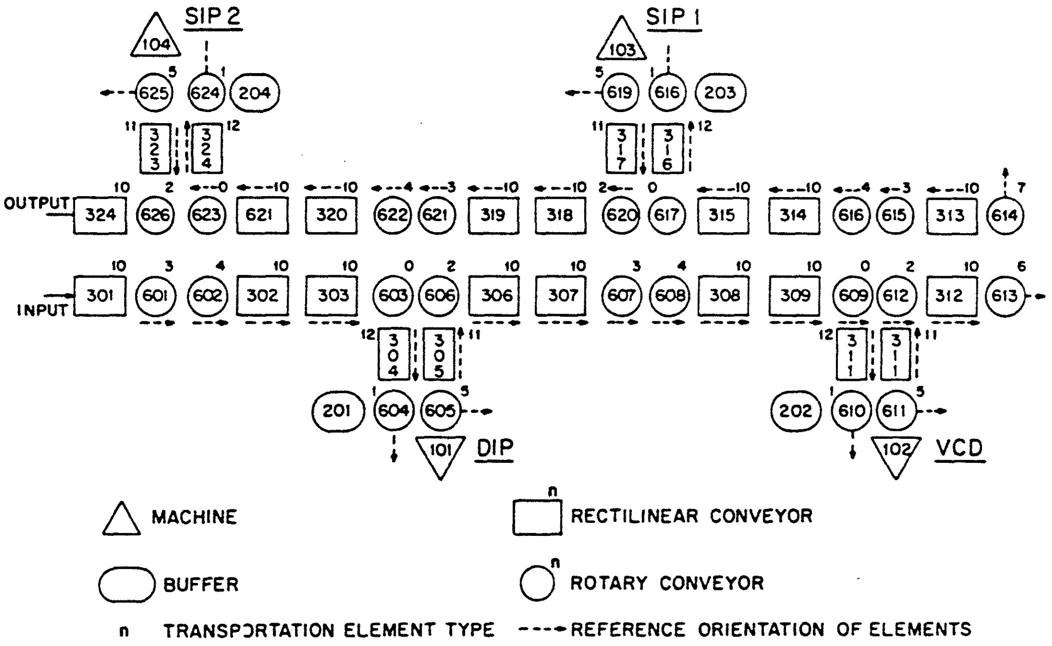

Fig. 2. IBM Miniline 1300.

TABLE I
TRANSPORTATION MECHANISM

Transfer Time of Card from Element to Element (straight or rotation): 1 sec.
Rotation Time: 6 sec.
Number of Movements to Transfer Card via Rotary Mode = 1 Rotation and 1 Transfer

TABLE II
MACHINE PARAMETERS

MACHINE	MTBF (MINUTES)	MTTR (MINUTES)	EFFICIENCY (%)	EXPECTED(%) UTILIZATION
1	600	60	90.91	97.68
2	600	60	90.91	91.10
3	600	60	90.91	96.03
4	600	60	90.91	96.58

TABLE III
OPERATION TIMES AND DEMAND RATES

OPERATION TIMES (sec)

MACHINE \ PART TYPE	1	2	3	4	5	6
1	40	40	0	0	20	60
2	0	0	60	30	40	40
3	0	100	0	0	70	0
4	0	0	0	80	0	80

DEMAND RATES (parts/sec)

PART TYPE	1	2	3	4	5	6
DEMAND RATE	.0080	.0070	.0060	.0070	.0025	.0040

a machine is the ratio of the total machine time required to the expected machine time available. The total machine time required is the product of total demand and processing time. The expected time a machine is available is its availability multiplied by the total time period. Table II displays the average utilizations for the machines in the configuration reported in the runs in Section VI. This is not IBM data; it was created to impose a heavy loading on the simulated production system. The actual utilization in any sample simulation run depends on the time history of machine failures and repairs during that run. This time sequence determines the actual amount of time a machine is available.

III. SCHEDULING OBJECTIVES AND POLICY PERFORMANCE MEASURES

Production Requirements and Scheduling Objectives

An FMS is normally only one stage of a production process, with other stages preceding and following. This necessitates coordinated production scheduling. The schedule must determine the part types and the number of each type to be produced by the FMS over a period of several days. The objective of the short term schedule is to track demand over the course of each day so as to meet the production targets set by the long-term schedule.

The production target is specified for each j as $D_j(T)$ parts of type j having to be made by time T, the production period. The cumulative production $W_j(t)$ is the total amount of material of type j produced by time t. The cumulative production must equal the total demand at time T; that is, one of the objectives is to ensure that $W_j(T)$ is equal to $D_j(T)$.

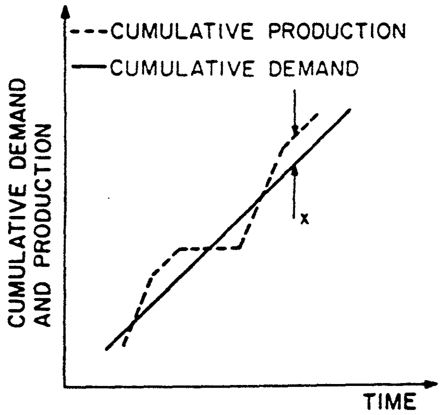

Fig. 3. Production to track demand.

It is convenient to define the demand rate

$$d_j = D_j(T)/T \tag{1}$$

and

$$D_j(t) = td_j. \tag{2}$$

At time t, the production surplus $x_j(t)$ is the difference between the total number of parts of type j produced and the total number of parts required:

$$x_j(t) = W_j(t) - D_j(t). \tag{3}$$

Fig. 3 illustrates the cumulative demand $D_j(t)$ being tracked by the cumulative production $W_j(t)$. Our objective is to meet production targets as closely as possible at the end of time period T, or, equivalently, to keep $x_j(T)$ close to zero.

The hierarchical policy is designed to keep $x_j(t)$ as close as possible to zero for all t. It does this by allowing the production surplus to grow, when enough machines are operational, to a certain level (defined below as the hedging point). When an essential machine fails, the surplus declines and becomes negative. The level is chosen so that the average value of $x_j(t)$ is near zero.

Policy Performance Measures

The production percentage, defined as

$$P_j = W_j(T)/D_j(T) \times 100 \text{ percent}, \qquad \text{for all } j \tag{4}$$

is of primary importance. This is the production of type j parts expressed as a percentage of total demand for type j. The closer this measure is to 100 percent, the better the algorithm is judged to be.

Also of interest is the average work-in-process, i.e., the average number of parts of each type present in the system. The smaller the WIP, the better the algorithm.

Finally, to compare various control policies, it is necessary to aggregate the performance measures by part type, into total performance measures. They are total production percentage

$$P = \Sigma_j W_j / \Sigma_j D_j \times 100 \text{ percent} \tag{5}$$

and total average work-in-process.

To measure the distribution of production between the various part types, we define balance as

$$B = \min_j \ P_j / \max_j \ P_j \times 100 \text{ percent}. \tag{6}$$

This is the ratio of the worst production percentage to the best percentage.

Let T_i(used) be the time that machine i processes parts, during the period of time T_i that it is operational. Machine utilization is given by

$$Z_i = T_i \text{ (used)}/T_i \times 100 \text{ percent}. \tag{7}$$

If this ratio is close to 100 percent, there is an efficient use of system resources, with very little idle time.

IV. THE HIERARCHICAL POLICY

The objective of the hierarchical scheduler is to meet production targets as closely as possible. This is to be achieved in the presence of random disturbances. Here, we treat only machine failures, although other types of uncertainties, such as random demand, will be dealt with in this framework in the future.

For efficient production, congestion in the transportation system and in internal buffers must be minimized. The hierarchical policy ensures this by respecting the system capacity constraints. The loss of production due to machine failures is compensated for by hedging, that is, by building up safety stock. We discuss these important concepts in detail below.

Capacity Constraints

All operations at machines take a finite amount of time. This implies that the rate at which parts are introduced into the system must be limited. Otherwise, parts would be introduced into the system faster than they can be processed. These parts would then be stored in buffers (or worse, in the transportation system) while waiting for the machines to be free, resulting in undesirably large work-in-process. The effect is that throughput (parts actually produced) drops with increasing loading rate, when loading rate is beyond capacity. Thus defining the capacity of the system carefully is a very important first step for on-line scheduling.

Consider a set of I machines processing J part types. Let the time to process the jth part type at machine i be τ_{ij}. Assume that W_j parts of type j must be processed at machine i during a period of T seconds.

The time required by machine i to produce all the parts is

$$\tau_{i1} W_1 + \tau_{i2} W_2 + \cdots + \tau_{iJ} W_J.$$

For the cumulative production to be feasible, this time must be less than or equal to T_i, the time available at machine i during the total time period T. (T_i is less than or equal to T. It is less when failures occur during this period.)

The parts can be processed if

$$\tau_{i1} W_1 + \tau_{i2} W_2 + \cdots + \tau_{iJ} W_J \leqslant T_i. \tag{8}$$

The average capacity of machine i is this limit on the number of parts that can be produced in a period of time T.

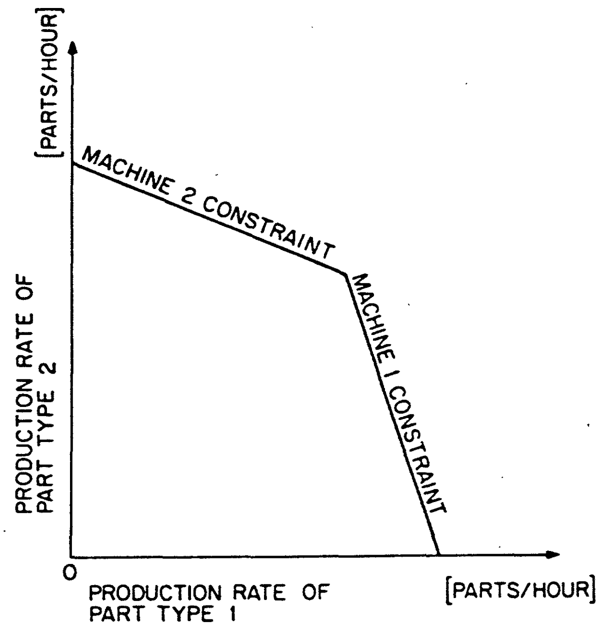

Fig. 4. Production capacity.

Because of the finite processing times, producing parts of one type implies that the time available to produce other types is reduced. The finite resource of machine availability determines, according to (8), the set of production quantities and mixes that can be produced in a given period of time. Fig. 4 describes the feasible production set of parts for a simple case.

Let $e_i = T_i/T$ be the availability of machine i. Let u_j be the average value of the production rate type j. Define the average capacity constraint set

$$\Omega=[u_j, \quad j=1, \cdots, J \;|\; \Sigma_j\tau_{ij}u_j \leqslant e_i, \quad \text{for all } i, \text{ and } u_j \geqslant 0]. \tag{9}$$

The capacity discussed so far is a long-term capacity. However it is necessary to determine at every instant whether a given part can be loaded. We must therefore find the instantaneous capacity. To do this, we first rewrite (8) as

$$\tau_{i1}u_1+\tau_{i2}+\cdots+\tau_{iJ}u_J \leqslant T_i/T \tag{10}$$

where

$$u_j=W_j/T \tag{11}$$

is the production rate of type j parts.

For T sufficiently small, the machine operational state does not change. Depending on whether machine i is up or down, T_i is either T or 0. Denote the operational state of the machine by α_i. That is,

$$\alpha_i=\begin{cases}0, & \text{if machine } i \text{ is down}\\ 1, & \text{if machine } i \text{ is up.}\end{cases} \tag{12}$$

Note that e_i is the average value of α_i over a long period.

The current or instantaneous capacity is then defined by

$$\tau_{i1}u_1+\tau_{i2}u_2+\cdots+\tau_{iJ}u_J \leqslant \alpha_i \tag{13}$$

for each i. As machines fail or are repaired, i.e., as the machine state changes, the set of feasible instantaneous production rates change. The key element of the hierarchical policy is to impose the discipline of satisfying the previous inequality at all times.

If there are several identical class i machines, α_i is a positive integer. This quantity changes as machines fail and are repaired. The machine state is defined by

$$\alpha=(\alpha_1, \alpha_2, \cdots, \alpha_I). \tag{14}$$

An instantaneous production rate is feasible only if it is a member of the capacity constraint set

$$\Omega(\alpha)=[u_j, \quad j=1, \cdots, J \;|\; \Sigma_j\tau_{ij}u_j \leqslant \alpha_i, \quad \text{for all } i, \text{ and } u_j \geqslant 0]. \tag{15}$$

Fig. 4 shows the capacity constraint set for a two part type, two machine system. Figs. 5(a) and 5(b) indicate how production rates drop to zero when one machine fails.

Fig. 5(c) describes a slightly more general situation. Here there are two part types being processed by two machines, two of which are identical ones which have been pooled together. α_1 can take the values 0, 1, 2. When one of the type 1 machines fails, the capacity set reduces to the smaller set indicated by dotted lines.

These examples indicate that when a machine fails, either some part types cannot be produced at all, or can be produced only at a reduced rate. The capacity constraint set describes precisely this notion as a function of the machine state.

To summarize, this notion of instantaneous capacity is crucial in the hierarchical policy. It describes the set of production rates one can choose from, while ensuring that queues do not build up in the system. Any choice of production rates must observe the discipline of staying within the capacity constraint set.

Hedging

Section III concludes that keeping the production surplus x_j small is an effective way of tracking demand. However failures result in a shortfall in production capacity. One compensates by building up safety stocks by overproducing when possible.

Thus rather than maintaining $x_j(t)$ at a value near zero for all t, it is reasonable to maintain it near a level $H_j(\alpha)$ while the machine state is α. We call $H_j(\alpha)$ the hedging point.

Overview of the Hierarchical Policy

The scheduler is divided into three levels, as shown in Fig. 1. The top level generates the decision parameters of the policy. These include the hedging points $H_j(\alpha)$ and other quantities. The repair and failure time data of the machines and the demand rate and processing times for each part type are required for this calculation.

The middle level computes the short-term production rates for each part type for each machine state. The lower level dispatches parts into the manufacturing system with the aim of maintaining the part loading rate equal to the computed production rate.

The top level is intended for off-line computation. It is

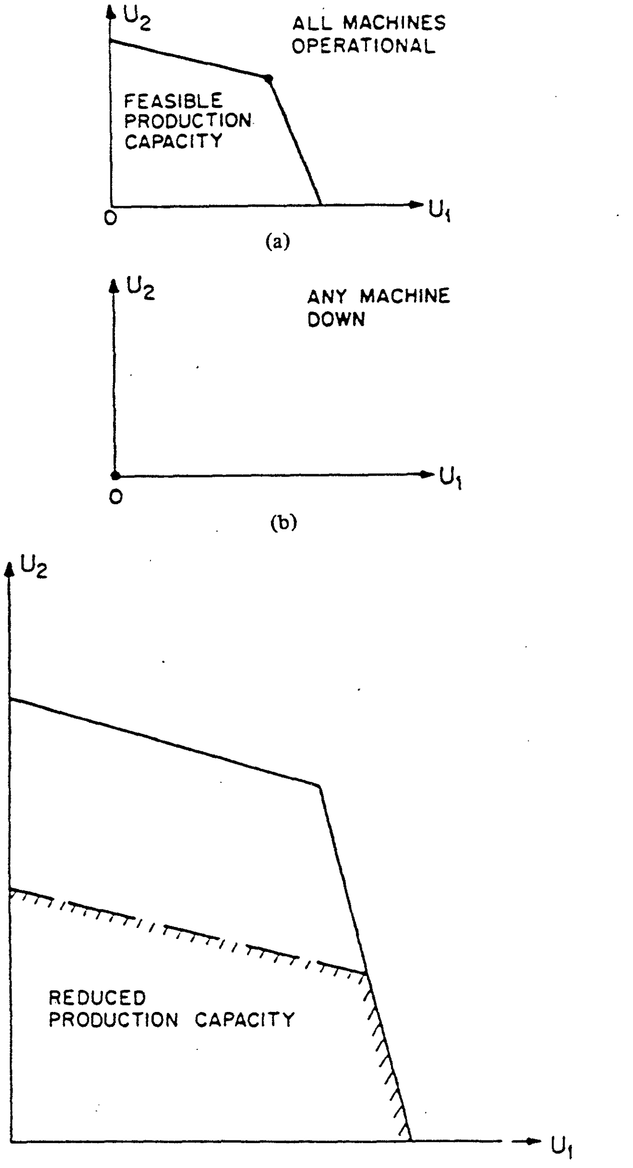

Fig. 5. (a) Capacity with both machines up. (b) Capacity with any machine down. (c) Capacity of two identical machines with one machine down.

designed to be called just once, at the start of a production run. However, if the need arises, it can be called on-line to update the decision tables.

When there is a change in machine state, i.e., when either a machine fails or is repaired, the middle level is called to compute the new values of the production rates. The resulting production surplus or buffer state trajectory is also computed. At the lowest level, parts are loaded into the system so as to follow the buffer state trajectory computed at the middle level as faithfully as possible. A detailed description of each of the levels follows.

Middle Level

This is the most important level in the hierarchy. At this level, the current production rate of each part type is determined for machine state α and buffer level x. The objective is to compute the production rates such that x approaches and then remains equal to $H(\alpha)$. This is possible only if the machine state α is such that the demand rate vector d satisfies

$$d \in \Omega(\alpha),$$

that is, only if the production rate vector u may equal or exceed d. If not, the production surplus must inevitably turn into a backlog (i.e., some components of x eventually become negative).

At the middle level, the scheduler choses production rates so that when enough capacity is present, the production surplus approaches and stays at the hedging point. If too many machines are unavailable for that, the scheduler choses from among the available production rates a set of rates to control the manner in which the production surplus declines and becomes a backlog.

Consider the situation when the machine state α is such that several part types can have production rates exceeding their demand rates. The scheduler tends to allocate manufacturing system resources to those types j whose

$$x_j - H_j(\alpha)$$

is most negative, i.e., whose production surplus is most behind its target value. It sometimes deviates from this behavior; it may allow x_j to decrease even when it is less than the hedging point so as to concentrate resources on some other part type that is farther behind or more vulnerable to future failures.

If machine state α persists for long enough, all part types k whose demands are feasible eventually have their buffer state x_j equal to the hedging point H_j. After that time, the production rate u_j is set equal to the demand rate d_j.

These desirable characteristics are the result of choosing the production rates as the solution to a certain linear programming problem. The cost coefficients are $c_1, \cdots, c_J$. They are functions of x which, along with the hedging points, are determined at the top level. Coefficient c_j tends to be negative when type j is behind or below its hedging point, and its absolute value tends to be larger for more valuable or more vulnerable parts.

The linear program minimizes a weighted sum of the production rates. It is restricted to those production rates that are currently feasible, i.e., that can be achieved by the current set of operational machines.

Linear Program: Minimize

$$c_1u_1 + c_2u_2 + \cdots + c_Ju_J$$

subject to

$$\Sigma_j \, \tau_{ij}u_j \leqslant \alpha_i, \qquad \text{for all } i \tag{16}$$

$$u_j \geqslant 0, \qquad \text{for all } j.$$

Production rates generated according to this program automatically satisfy the instantaneous capacity constraints. This linear program is not hard to solve on-line since the number of constraints and unknowns is not large.

If the coefficients c_j are all positive, the production rates satisfying the linear program are zero. Fig. 6 shows this for a

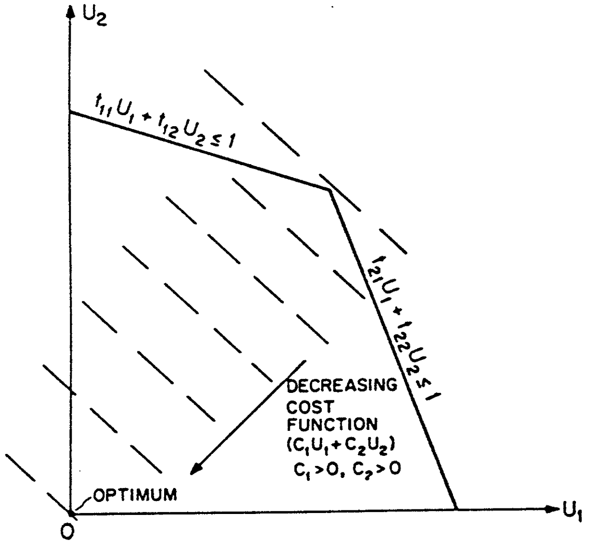
U2
$t_{11}U_1+t_{12}U_2\leq 1$
$t_{21}U_1+t_{22}U_2\leq 1$
DECREASING COST FUNCTION $(C_1U_1+C_2U_2)$ $C_1>0,\ C_2>0$
OPTIMUM
O
U1

Fig. 6. Optimum production rates for all positive cost coefficients.

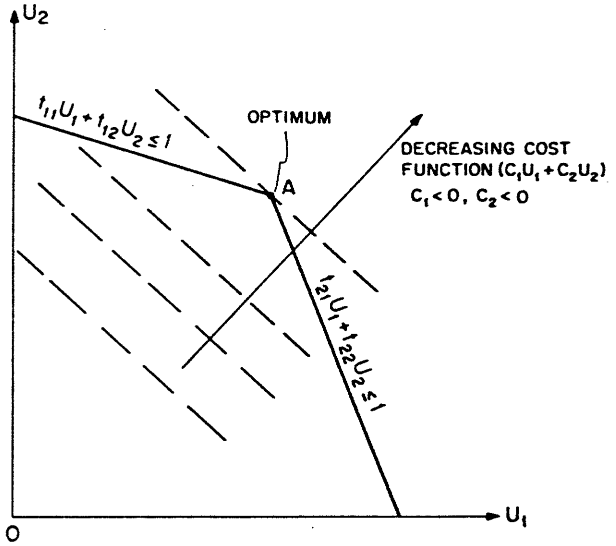
U2
$t_{11}U_1+t_{12}U_2\leq 1$
OPTIMUM
A
DECREASING COST FUNCTION $(C_1U_1+C_2U_2)$ $C_1<0,\ C_2<0$
$t_{21}U_1+t_{22}U_2\leq 1$
O
U1

Fig. 8. Optimum production rates for all negative cost coefficients.

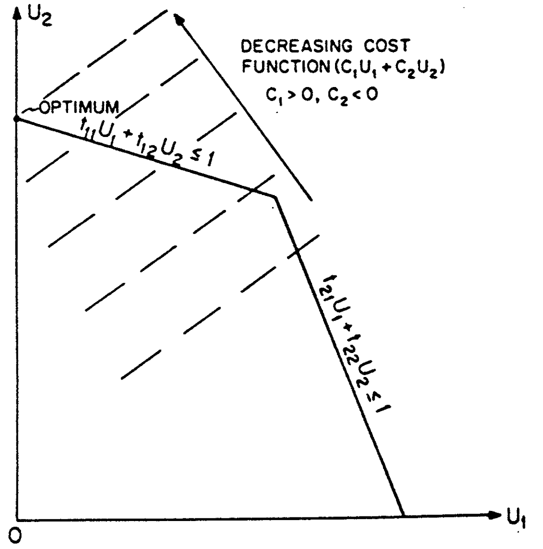
U2
DECREASING COST FUNCTION $(C_1U_1+C_2U_2)$ $C_1>0,\ C_2<0$
OPTIMUM
$t_{11}U_1+t_{12}U_2\leq 1$
$t_{21}U_1+t_{22}U_2\leq 1$
O
U1

Fig. 7. Optimum production rates for positive and negative cost coefficient.

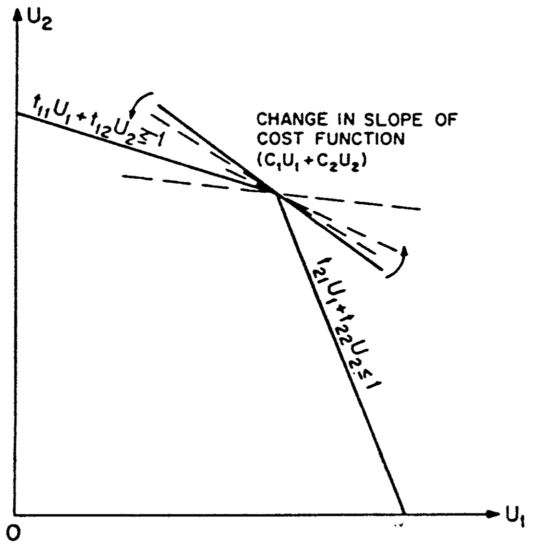
U2
$t_{11}U_1+t_{12}U_2\leq 1$
CHANGE IN SLOPE OF COST FUNCTION $(C_1U_1+C_2U_2)$
$t_{21}U_1+t_{22}U_2\leq 1$
O
U1

Fig. 9. Variation of cost coefficients.

simple two-machine two-part system. Fig. 7 represents the situation when one of the coefficients is negative and the others are all positive. Then the solution is such that the part type associated with the negative coefficient is produced at the maximum permissible rate. All the other production rates are set to zero.

If all the coefficients are negative, Fig. 8 shows the prevailing situation. An optimal production rate mix, corresponding to point A in the figure, is chosen. More general situations follow from these.

The cost coefficients of the linear program are given by

$$c_j(x_j)=A_j(\alpha)(x_j-H_j(\alpha)) \tag{17}$$

where $A_j(a)$ and $H_j(\alpha)$ are determined at the higher level. $A_j(\alpha)$ is a positive quantity that reflects the relative value and vulnerability of each part type.

The production surplus $x(t)$ is given by (3). It is approximately

$$x(t)=\int_0^t [u(t)-d(t)]\ dt \tag{18}$$

since the function of the lower level is to keep the actual production rate close to the value calculated here.

As $x(t)$ changes, the coefficients of linear program change as in Fig. 9. However the production rates of the different part types remain constant, up to a point. When the coefficients change sufficiently, the production rates jump abruptly to new values.

In principle it is necessary to solve the linear program at every time instant because it is constantly changing. This was the approach followed by Kimemia [4] and Kimemia and Gershwin [5]. However this adds a computational burden which would be best to circumvent, and it leads to undersirable behavior when implemented. Gershwin, Akella, and Choong [2] discuss this behavior and a technique for eliminating it. This technique reduces much of the computational burden associated with the linear program.

To describe the behavior of the scheduling system, there are two cases to consider. The first is that the machine state is such that the demand rate is feasible; that is, that $u = d$ is a possible choice for the production rates. In this case, $x(t)$ eventually reaches the hedging point, and the cost coefficients are all zero and the linear program does not determine the value of u. Gershwin, Akella, and Choong [2] demonstrate, however, that when that happens, the solution is $u = d$ and, according to (18), $x(t)$ remains constant, at the hedging point.

When the demand is not feasible, some of the production rates must be less than the corresponding demand rates. The production surplus for these part types fall below the corresponding hedging points. The c_j coefficients then become negative and decrease. Only those part types which are feasible and at or below their hedging points are produced. The rate at which they are produced depends on the coefficients, which describe the relative deviations of the production surplus from desired values.

The system operates on a random cycle: when the machine state α is feasible, the production surplus x approaches H and then stays there. When a machine fails so that the machine state is not feasible, x moves away from H and eventually may become negative.

To complete the picture, the top level is required to determine A and H. These are functions of the relative values of the parts and of the reliabilities of the machines that they visit. The bottom level is required to choose time instants to load parts to guarantee that the production rates and production surplus calculated at the middle level are actually realized.

Lower Level

The lower level has the function of dispatching parts into the system in a way that agrees with flow rates calculated at the middle level. As described in detail in Gershwin, Akella and Choong [2], the middle level of the scheduler calculates the projected trajectory, $x^P(t)$, the best possible future behavior of $x(t)$ if no repairs or failures would occur for a long time.

The lower level treats the projected trajectory $x^P(t)$ as the value that the actual production surplus $x^A(t)$ (given by (3)) should be close to. A part of type j is loaded into the system whenever the actual production surplus $x_j{}^A(t)$ is less than its projected value $x_j{}^P(t)$. When there is a machine state change, a new projected trajectory is calculated starting at the time of the change, and the same loading process continues with the new trajectory.

A fuller description of the implementation of the loading process appears in Akella, Bevans, and Choong [1]. A qualitative description of its behavior is in Gershwin, Akella, and Choong [2].

Higher Level

The purpose of the top level of the algorithm is to provide the A_i and H_i parameters to the middle level. These quantities are used in (17) to evaluate the cost coefficients c_i of linear program (16).

Gershwin, Akella, and Choong [2] provided the following formula for the hedging point of part i, where the machine state is feasible:

$$H_i = \frac{T_r d_i(bU_i - ad_i) - T_f ad_i(U_i - d_i)}{(a+b)U_i} \tag{19}$$

where T_r is the average mean time to repair (MTTR) of all the machines part i visits, T_f is the average mean time between failures (MTBF). U_i is the average production rate of part i before x_i reaches the hedging point, and a and b are weighting parameters. The last two quantities reflect the relative penalty incurred for temporary surplus and backlog.

To further simplify the analysis, we assumed that a, b, T_r and T_f and U_i were such that

$$H_i = d_i T_r/2. \tag{20}$$

The coefficients $A_j(\alpha)$ can be computed from the number of machines that type j parts visit. The more machines each part type visits, the more vulnerable that part type is to failures. Also the smaller the mean time between failures, the more the vulnerability. To simplify our analysis, we assumed that the mean times between failures of all the machines are the same. Thus,

$$A_j(\alpha) = \text{number of machines that type } j \text{ parts visit.} \tag{21}$$

These formulas are highly simplified, but, as the simulations show, they work very well. Further research is required to ascertain under what general conditions they can be expected to provide good results.

The reference values for the H and A parameters for the simulated system, computed according to (20) and (21), are tabulated in Table IV.

V. ALTERNATIVE POLICIES

In this section we discuss a number of simpler policies. All of them limit the number of parts in the system. The differences lie in the amount of information they use about system status and how they use this information.

There are important differences between the hierarchical policy and those described in this section. The most important is that these policies are not explicitly based on satisfying the capacity constraints. As a result, there are periods during which they load more parts than the system can process. Material accumulates in the system during those periods, leading to congestion and diminished effective capacity.

The second is that they require a fair amount of tuning to perform well. "Tuning" is the process of repeating a simulation several times in order to obtain the best values for a set of parameters. Tuning is undesirable because it is expensive. It is impractical because actual production may differ radically from tuning runs, so that good performance cannot be guaranteed.

The third difference is that the policies are not hierarchical. They do not separate the scheduling problem into a set of problems with different characteristic time scales. As a consequence they are difficult to analyze and their performance—and more importantly, the performance of any manu-

TABLE IV
REFERENCE VALUES OF CONTROL PARAMETERS

MACHINE STATE:	(1,1,1,1)
A:	(1,2,1,2,3,3)
HEDGING PT:	(15,12,15,10,6,8)
MACHINE STATE:	(0,1,1,1)
A:	(1,2,1,2,3,3)
HEDGING PT:	(0,0,35,31,0,0)
MACHINE STATE:	(1,0,1,1)
A:	(1,2,1,2,3,3)
HEDGING PT:	(34,16,0,0,0,0)
MACHINE STATE:	(1,1,0,1)
A:	(1,2,1,2,3,3)
HEDGING PT:	(19,0,19,16,0,13)
MACHINE STATE:	(1,1,1,0)
A:	(1,2,1,2,3,3)
HEDGING PT:	(19,16,19,0,12,0)
MACHINE STATE:	(any 2 machines operational)
A:	(1,2,1,2,3,3)
HEDGING PT:	(40,35,40,35,0,0)

facturing system they control—is difficult to predict other than by simulation.

These policies are based on the amount of material already loaded into the system. Cumulative production for each part type is considered to be the total number of parts loaded. It is equal to the number of parts completed (PDONE_j) plus the number of parts currently in the system (PINSYS_j). That is,

$$W_j(t) = \text{PDONE}_j(t) + \text{PINSYS}_j(t). \tag{22}$$

Also,

$$D_j(t) = d_j t \tag{23}$$

$$\begin{aligned} x_j(t) &= W_j(t) - D_j(t) \\ &= \text{PDONE}_j(t) + \text{PINSYS}_j(t) - d_j t. \end{aligned} \tag{24}$$

Simplest Policy: Policy X

This policy loads a part whose type is furthest behind or least ahead of cumulative demand. That is, it loads a type j part, where x_j is minimal.

Some limit has to be set on the total number of parts in the system in order to avoid filling up the buffers and transportation system. We define N to be the maximum permissible total number of parts in the system.

Also, buffers upstream and downstream of the FMS maybe have limited capacities, or the cost of extra inventory may be high. Thus even if production is ahead of demand, a limit E_j on excess production is useful. That is, we require that

$$x_j \leqslant E_j. \tag{25}$$

Our experience suggests that this is necessary. Production system performance is considerably degraded in the absence of this constraint.

The policy can now be described more precisely.

Policy X: At each time step,

1) do not load any part type if $\Sigma_j\,\text{PINSYS}_j > N$,
2) do not load a type j part if $x_j(t) > E_j$,
3) do not load a type j part if $W_j(t) > D_j(T)$, that is, if the cumulative production at time t exceeds the cumulative demand for the entire period T,
4) of the remaining part types, pick type j that minimizes $x_j(t)$, i.e., load the part type which is least ahead or furthest behind the production target.

Little's Law

Little's law [6] is useful in estimating number of parts in the system. It provides an expression for the sizes of queues of parts (N_j) in terms of the rate at which they arrive (the demand rate d_j) and the average time required for each part to be processed by the system (w_j). The expression is

$$N_j = d_j w_j. \tag{26}$$

That is,

$$\text{part in system} = \begin{array}{l}\text{(demand rate)} \times \text{(average time} \\ \text{required to process each part).}\end{array}$$

Note that for N_j to represent the total number of parts of type j throughout the system, w_j must include all sources of delay, including operation time, travel time, and queuing time. Queuing delay, i.e., time spent waiting in buffers or in the transportation system, is neglected for the first guess because it is difficult to calculate and because we intend to keep the number of parts in the system sufficiently small so that such delays are small.

Using this result, a first guess for N can be obtained. The expected number of parts in the system is the sum of the expected number of parts of each type, or

$$N = \Sigma_j N_j. \tag{27}$$

As the threshold limit N is increased, the following system performance is expected and is indeed confirmed by simulation runs.

1) The production rate increases—up to a limit. This limit is less than the system's capacity as calculated in Section IV.
2) The WIP increases.

In addition the balance improves. This was not expected since there is no direct connection between balance and N.

Note that an increase in the work-in-process (WIP) is particularly likely when a machine fails. The parts going to that machine can not be processed. One of these part types will soon fall furthest behind. Consequently more parts of the same type will be loaded. If N is large, the corresponding buffer eventually fill up, and the whole system becomes congested. If N is small, this problem is avoided, but the production performance will be poor, due to under-utilization of machines.

In the rest of this section, we describe other policies, which use more information than policy X to obviate some of its limitations.

More Sophisticated Policy: Policy Y

Two changes are likely to improve the performance of policy X. First, treating each part type separately should result in better balance. This is incorporated in policy Y. Considering machine operational status when loading parts is part of policy Z.

Policy Y is the same as policy X except that there is a separate threshold N_j for each part type. It can be stated as follows.

Policy Y:

1) Do not load a type j part if $\mathrm{PINSYS}_j > N_j$.
2) Steps 2–4 are as in policy X.

The initial guesses for the N_j parameters are simple (26). While performance should improve as a result of using a policy that uses more information about the current status of the system, it comes at a price. There are more parameters to tune now, which in principle requires more computer simulation runs. We circumvented that (possibly at the price of not getting the best possible performance) by using a common scaling factor for all N_j.

Production percentage as well as balance should improve relative to policy X. This is a consequence of loading individual part types according to demand. WIP also decreases for the same reason.

Most Sophisticated Policy: Policy Z

While policy Y uses demand information for individual part types, it does not use machine failure information. When a machine fails, the flow rate of parts going to it should be set to zero. Equivalently the limit N_j should be set to zero. This ensures that the WIP does not increase due to the introduction of parts which cannot be processed. The production percentage is likely to increase as delays due to loading the wrong part types are reduced.

Policy Z:

1) Do not load a type j part if $\mathrm{PINSYS}_j > q_j N_j$. The parameter q_j is given by

$$q_j = \begin{cases} 0, & \text{if any machine that type } j \text{ parts visit has failed} \\ 1, & \text{otherwise.} \end{cases} \tag{28}$$

2) Steps 2–4 are as in policies X and Y.

The same considerations about tuning both the N_j and the E_j parameters apply here as in policies X and Y. Note that N_j should be greater than (26) since parts should be loaded at a rate greater than d_j when their machines are operational. This means that we are making more parts of each type when we can, hedging against future machine failures.

Policy Z shares these features with the hierarchical policy. However the hierarchical policy guarantees that capacity constraints are always satisfied. Policy Z does not, so WIP can be expected to be greater. Note that the E_j parameters here are similar in their effect to the hedging points H_j in the hierarchical policy.

It is reasonable to expect that this policy behaves better than X or Y, but not quite as well as the hierarchical. Simulations confirm this.

VI. SIMULATION RESULTS

In this section we describe simulation results to evaluate the performance of the hierarchical policy. We also compare the hierarchical policy and policies X, Y, Z. We use the part and machine data of Section II. To understand the policies and not get lost in a welter of detail, a relatively small number of part types are treated.

The system is heavily loaded. That is, machines have to be used for a large percentage of the time they are operational to satisfy demand. This is the only situation in which it is meaningful to compare policies. Under lighter loading conditions, any strategy may be effective. However light loading is not generally realistic; the cost of capital equipment is such that managers will need to get the most they can from an FMS.

In these simulations, the objective is to produce a given quantity of material by the end of one shift. There is no incentive to produce more than the required amount. Consequently the maximum production of any part is 100 percent of requirements, and, because we are loading the system heavily, less than 100 percent is produced in most cases. We expect that over a longer period, such as a week, the hierarchical policy would most often fully meet the requirements imposed here.

Hierarchical Versus Policy X

Our runs correspond to an eight-hour production shift. We first examine the performance of the hierarchical policy during a given run, with different values of the hedging and A parameters. This is compared with the performance of policy X for different values of the threshold limit N on parts in the system. The highlights of the performance are summarized in Figs. 10 and 11. Tables VII–XVI contain detailed production summaries.

Fig. 10 is a plot of total production percentage versus in-process-inventory, for different parameter values of the two strategies. The reference values of the A_j and hedging points H_j are chosen as described in Section IV and tabulated in Table IV. They are varied as shown in Tables V and VI. The parameter N is chosen as described in Section V and tuned. The actual values are tabulated in Tables XII–XVI.

All the points corresponding to the hierarchical controller lie in the upper left region of the graph in Fig. 10. This indicates a high total production percentage, and a low WIP. Both high production percentage and low WIP are highly desirable, as we indicated in Section III. Simultaneously achieving these objectives demonstrates the effectiveness of the hierarchical structure.

The points corresponding to different hedging parameters are clustered close together. This shows robustness to parameter perturbations. The parameters are computed from demand, machine and part type data, which are not always known accurately. Any strategy not unduly sensitive to these is preferred. This is a very important characteristic. Not only does it imply that a great deal of data-gathering and processing

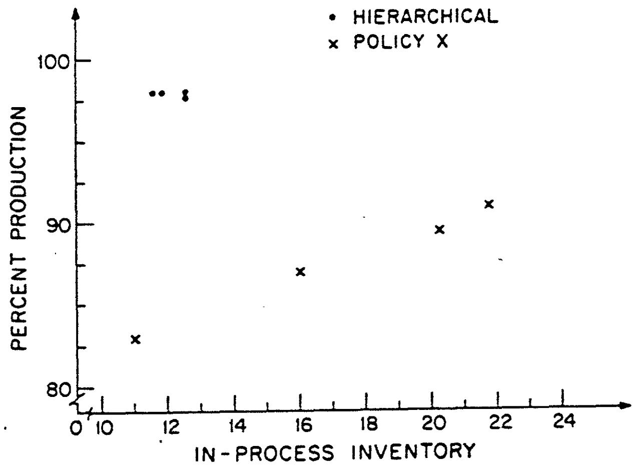

Fig. 10. Production versus in-process inventory.

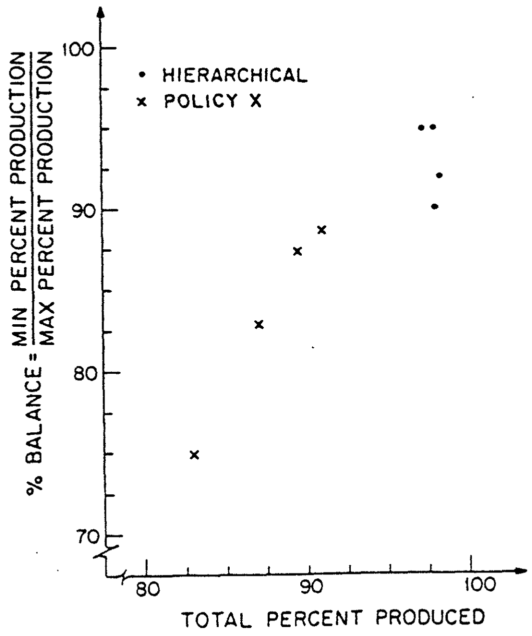

Fig. 11. Balance versus total production percentage.

TABLE V
VARIATION OF A

ORIGINAL A:	(1,2,1,1,3,3)
NEW A:	(1,2,1,5,3,8)

is not required, but it also means that the system's behavior can be expected to be stable even as its reliability drifts over time.

In contrast, the simpler policy's results are more scattered and corresponded to a combination of higher WIP and lower production percentage. The hierarchical policy far out-performs policy X.

Consider the effect of tuning policy X by increasing the threshold limit N of parts in the system. The average WIP in the system is increased in an attempt to increase the production percentage. More parts are loaded into the system and are available at the buffers so that idle time is reduced. Consequently the machines are better utilized and production percentage increases. This approach is relatively crude and

TABLE VI
VARIATION OF HEDGING POINTS

MACHINE STATE:	(1,1,1,1)
ORIGINAL HEDGING PT:	(15,12,15,10,6,8)
NEW HEDGING PT:	(15,12,15,13,6,10)
MACHINE STATE:	(0,1,1,1)
ORIGINAL HEDGING PT:	(0,0,35,31,0,0)
NEW HEDGING PT:	(0,0,35,40,0,0)
MACHINE STATE:	(1,0,1,1)
ORIGINAL HEDGING PT:	(34,16,0,0,0,0)
NEW HEDGING PT:	(34,16,0,0,0,0)
MACHINE STATE:	(1,1,0,1)
ORIGINAL HEDGING PT:	(19,9,19,16,0,13)
NEW HEDGING PT:	(19,0,19,20,0,16)
MACHINE STATE:	(1,1,1,0)
ORIGINAL HEDGING PT:	(19,16,19,0,12,0)
NEW HEDGING PT:	(19,16,19,0,12,0)

TABLE VII
HIERARCHICAL POLICY RUNS WITH VARYING CONTROL PARAMETERS

DESCRIPTION OF RUN	TOTAL PERCENT PRODUCED	BALANCE	WIP
REFERENCE	98.1	90.4	11.81
INCREASED A	98.0	95.0	12.64
INCREASED HEDGING POINTS	98.3	92.2	11.58
INCREASED A AND HEDGING POINTS	97.8	95.0	12.62

SEED = 123457.

TABLE VIII
PRODUCTION SUMMARY—HIERARCHICAL POLICY WITH REFERENCE A AND HEDGING POINTS

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	230	100.0	193	1.54
2	201	201	100.0	435	3.04
3	172	172	100.0	179	1.07
4	201	193	96.0	410	2.77
5	71	71	100.0	523	2.29
6	115	104	90.4	573	2.08
TOTAL:	990	971	98.1	348.6	11.81

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.9	93.1
2	100.0	80.6
3	94.7	91.9
4	85.3	96.8

TABLE IX
PRODUCTION SUMMARY—HIERARCHICAL POLICY WITH INCREASED A

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	226	98.3	189	1.49
2	201	198	98.5	465	3.22
3	172	172	100.0	189	1.13
4	201	191	95.0	468	3.13
5	71	71	100.0	538	1.33
6	115	112	97.4	600	2.35
TOTAL:	990	970	98.0	373.3	12.64

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
101	92.9	93.8
102	100	81.5
103	94.7	90.8
104	85.3	98.9

TABLE X
PRODUCTION SUMMARY—HIERARCHICAL POLICY WITH INCREASED HEDGING POINTS

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	230	100.0	186	1.49
2	201	201	100.0	424	2.96
3	172	172	100.0	179	1.07
4	201	193	96.0	413	2.84
5	71	71	100.0	505	1.25
6	115	108	92.2	533	1.96
TOTAL:	990	973	98.3	341.1	11.58

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.9	93.5
2	100.0	81.1
3	94.7	91.9
4	85.3	97.6

TABLE XI
PRODUCTION SUMMARY—HIERARCHICAL POLICY WITH INCREASED A AND HEDGING POINTS

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	226	98.3	184	1.45
2	201	198	98.5	445	3.08
3	172	172	100.0	183	1.10
4	201	191	95.0	489	3.27
5	71	71	100.0	507	1.25
6	115	111	96.5	633	2.48
TOTAL:	990	969	97.8	372.4	12.62

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.9	93.9
2	100.0	81.5
3	94.7	90.8
4	85.3	98.6

TABLE XII
POLICY X RUN WITH VARYING N

N	TOTAL PERCENTAGE PRODUCTION	BALANCE (%)	WIP
11	82.2	74.8	11.00
16	87.1	82.9	15.98
20	89.5	87.3	19.96
22	91.1	88.4	21.95

SEED = 123457.

disregards system capacity constraints. This is the reason that the price of increasing WIP must be paid in order to increase production percentage. In fact, if the threshold N is increased inordinately, the system gets congested.

On the other hand the hierarchical policy always satisfies the capacity constraints and is thus able to achieve low WIP. The instantaneous feedback feature, which combines system status information with hedging for future machine failures, ensures a high production percentage.

The hierarchical policy and policy X are compared with respect to balance and production percentage in Fig. 11. The hierarchical policy is superior. The total production percentage is uniformly high and robust with respect to hedging point variations. This again checks with our expectation that the exact value of the hedging point is not as important as long as it is in the right range. What matters is that the hedging should ensure that the average production surplus is close to zero.

The policy is also robust with respect to changes in A_j, though less so. While the approximation based on vulnerabil-

TABLE XIII
PRODUCTION SUMMARY—POLICY X WITH $N = 11$

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	199	86.5	199	1.48
2	201	172	85.6	472	2.86
3	172	146	84.9	173	0.88
4	201	174	86.6	506	3.06
5	71	46	64.8	541	0.86
6	115	87	75.7	603	1.85
TOTAL:	990	824	83.2	388.0	10.99

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.9	78.5
2	100.0	67.1
3	94.7	74.8
4	85.3	85.0

TABLE XIV
PRODUCTION SUMMARY—POLICY X WITH $N = 16$

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	205	89.1	227	1.79
2	201	177	88.1	720	4.57
3	172	153	89.0	176	0.94
4	201	181	90.0	669	4.21
5	71	53	74.6	932	1.71
6	115	93	80.9	824	2.75
TOTAL:	990	862	87.1	519.7	15.98

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.0	82.1
2	100.0	71.2
3	94.7	78.8
4	85.3	89.2

TABLE XV
PRODUCTION SUMMARY—POLICY X WITH $N = 20$

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	208	90.4	277	2.26
2	201	181	90.0	828	5.44
3	172	158	91.9	176	0.97
4	201	185	92.0	921	5.93
5	71	57	80.3	1122	2.22
6	115	97	84.3	900	3.13
TOTAL:	990	886	89.5	628.6	19.96

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.9	84.2
2	100.0	73.8
3	94.7	81.0
4	85.3	92.2

TABLE XVI
PRODUCTION SUMMARY—POLICY X WITH $N = 22$

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	211	91.7	248	2.13
2	181	90.0	90.0	984	6.55
3	172	161	93.6	184	1.03
4	201	189	94.0	1024	6.73
5	71	59	83.1	967	2.06
6	115	101	87.8	956	3.45
TOTAL:	990	902	91.1	673.2	21.95

SEED = 123457.

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	92.9	85.7
2	100.0	75.6
3	94.7	81.5
4	85.3	94.4

ity to machine failures is adequate, even better balance may be possible with a more careful choice of these parameters. In any case, by redistributing available machine capacity effectively between the various part types and hedging, the hierarchical policy achieves good balance.

Policy X has lower balance, lower production percentage, and greater sensitivity to scheduling parameters (N) than the hierarchical policy. The production summaries in Tables XII–XVI show that considerably lower percentages of part types 5 and 6 are produced than are required. This is because these part types must visit more machines than the others. As a result, the likelihood of their waiting for disabled machines to be repaired is higher.

To compensate, more parts can be introduced into the system, at the expense of increased WIP. While the production of part types 5 and 6 improves, that of the other parts does not. Hence balance is better, but neither balance nor overall production percentage are as high as those achieved by the hierarchical policy.

Observe that the hierarchical policy is able to take into account these failures by hedging and building up buffer stocks (see Tables VIII–XI). The benefit of respecting capacity constraints is amply demonstrated by the much lower WIP of the hierarchical policy.

Another insight into the functioning of the hierarchical policy is provided by the machine utilization data. Under heavy loading, all the machines are scheduled to be as highly utilized as possible. Tables VIII–XI indicate that the machines that are down for the greatest periods are the ones that have the highest utilization when up. This implies that the policy is using these machines effectively. Policy X utilizes every machine much less (Tables XIII-XVI).

Comparison with Different Seeds

The same type of comparison is conducted between the hierarchical policy and policy X but for a set of different seeds. Each seed corresponds to a sequence of machine failures and repairs. That is, each seed represents a unique day. The same value of N (16) is used with each seed. The hierarchical policy is run with the same set of seeds. The results, shown in Figs. 12 and 13 and Tables XVII–XXII, are essentially similar to those seen in the previous subsection. The hierarchical policy achieves higher production percentages with lower WIP and better balance.

There is a particularly great difference between the performances of the hierarchical and policy X on certain days. The performance of the simpler policy is more variable, i.e., less predictable, from day to day. Tables XVII and XX indicate that the production percentages of the hierarchical policy stay within the range of 88.7 to 98 percent while those of policy X varies from 69 to 87.1 percent. Moreover the production balance of the hierarchical policy is in the range of 80.4 to 90.4 percent while that of policy X varies from a very low 38 to 83 percent. Table XXI shows the low percentage of type 5 parts produced for one of the runs. This variability is a serious consideration. A policy which is more predictable is more desirable to those who must make long range plans and predictions.

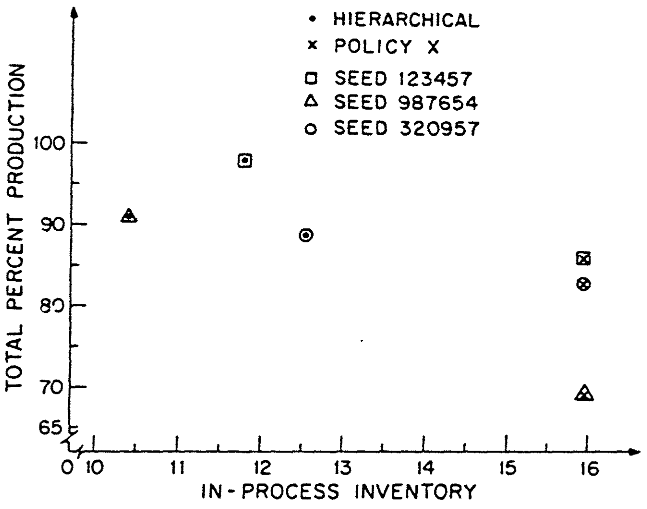

Fig. 12. Total production percentage versus in-process inventory for different seeds.

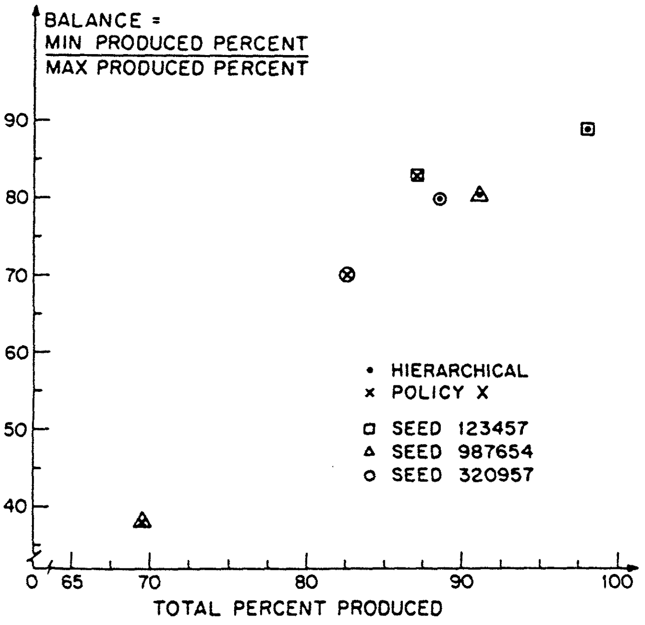

Fig. 13. Balance versus total production percentage for different random seeds.

TABLE XVII
HIERARCHICAL POLICY RESULTS WITH DIFFERENT SEQUENCES OF MACHINE REPAIRS AND FAILURES

	REFERENCE CONTROL PARAMETERS		
SEED	TOTAL PRODUCTION PERCENTAGE	BALANCE	WIP
123457	98.0	90.4	11.81
987654	91.0	80.3	10.38
320957	88.7	80.0	12.56

TABLE XVIII
PRODUCTION SUMMARY—HIERARCHICAL POLICY WITH SEED = 987654

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	230	100	165	1.33
2	201	182	90.5	480	3.06
3	172	156	90.7	189	1.03
4	201	177	88.1	381	2.38
5	71	57	80.3	485	0.97
6	115	99	86.1	467	1.60
TOTAL:	990	901	91.0	328.6	10.38

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	100.0	82.7
2	79.0	92.7
3	78.3	98.9
4	100.0	77.0

TABLE XIX
PRODUCTION SUMMARY—HIERARCHICAL POLICY WITH SEED = 320957

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	192	83.5	287	2.02
2	201	169	84.1	511	3.02
3	172	166	96.5	173	0.99
4	201	201	100.0	352	2.46
5	71	58	81.7	699	2.10
6	115	92	80.0	612	1.97
TOTAL:	990	878	88.7	384.7	12.56

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	74.5	99.4
2	100.0	76.6
3	100.0	72.9
4	89.7	90.7

TABLE XX
POLICY *X* RESULTS WITH DIFFERENT SEQUENCES OF MACHINE REPAIRS AND FAILURES WITH $N = 16$

SEED	TOTAL PRODUCTION PERCENTAGE	BALANCE	WIP
123457	87.1	83	15.98
987654	69.0	38	15.98
320957	82.6	71	15.98

TABLE XXI
PRODUCTION SUMMARY—POLICY *X* WITH SEED = 987654 ($N = 16$)

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	179	77.8	185	1.16
2	201	150	74.6	1101	5.75
3	172	123	71.5	411	1.76
4	201	149	74.1	601	3.39
5	71	21	29.6	2498	1.84
6	115	61	53.0	875	2.07
TOTAL:	990	683	69.0	650.4	15.98

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	100.0	61.3
2	79.0	67.7
3	78.3	73.2
4	100.0	58.5

TABLE XXII
PRODUCTION SUMMARY—POLICY *X* WITH SEED = 320957 ($N = 16$)

	REQUIREMENTS	PRODUCED	%	AVERAGE TIME SECS	AVERAGE WIP
TYPE					
1	230	200	87.0	429	3.46
2	201	171	85.1	633	4.03
3	172	141	84.3	179	0.91
4	201	172	85.6	615	3.69
5	71	44	62.0	669	1.03
6	115	86	74.8	951	2.87
TOTAL:	990	818	82.6	533.4	15.98

MACHINE	UP TIME PERCENT	UTILIZATION TIME PERCENT WHEN UP
1	74.5	98.4
2	100.0	66.7
3	100.0	70.2
4	89.7	80.2

Even if a policy is tuned carefully for a given run, its performance is not guaranteed to be good in runs with other seeds.·This shows the impracticality of parameter tuning. Not only is tuning expensive, since it may require many simulation runs, but the parameter values determined this way may be good only for one set of repairs and failures. In contrast the hierarchical policy is not tuned for a specific failure and repair pattern.

Comparison of Hierarchical Policy and Policies Y and Z

The performance of the hierarchical policy with the reference values of the hedging parameters is also compared with that of policies Y and Z. The parameters of these policies are chosen as described in Section V. We discuss the results only for one run with a single seed.

Figs. 14 and 15 show the comparative performances of all four policies. The hierarchical strategy has the best performance. It is better than policy Z, which is better than Y, which, in turn, is better than X.

This order is a direct result of the more effective use of information. Policy X does not differentiate between part types and does not make use of machine repair state information. It performs poorly in terms of all measures. Policy Y does much better in terms of average WIP and total production percentage by differentiating among part types. Policy Z also makes use of machine state and so has lower WIP and higher balance. The implication is that effective feedback based on more information results in better performance. The series of policies culminates in the hierarchical policy, whose sophisticated information usage helps it achieve superior performance.

VII. CONCLUSION

From the simulation results, we conclude that a hierarchically structured policy designed on the basis described here and elsewhere [4], [5], [2] is very effective in scheduling a FMS. It can achieve high output with low WIP and can cope with changes and disturbances. Future research will be directed toward incorporating other kinds of uncertainties and disturbances in the hierarchical structure.

The success of the policy is a result of using feedback and adhering to the discipline of respecting system capacity constraints. Capacity limits are not just observed in the long run; they are considered as each part is considered for loading into the system. All relevant machine and system status information is fully utilized.

This approach is robust so that for a wide range of policy parameters it works very well. This obviates the need for precise machine and part data which may not always be available. It also eliminates the need to use time consuming (and thus infeasible) trial runs. Further research is needed in choosing hedging and A_j parameters for larger systems. The grouping of parts into families when there are a large number of part types is another research issue.

A variety of new problems arise when we explicitly consider the scheduling of an FMS in the context of a factory. The FMS is then one of the stages of the automated production system. It is supplied with raw material by an upstream stage. It must supply the stage which is downstream from it. Coordinated production between the different stages becomes a necessity. This influences and complicates the short term scheduling problem.

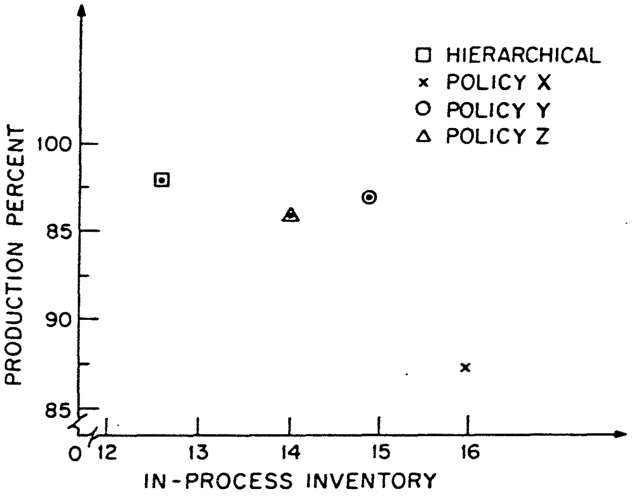

Fig. 14. Total production versus in-process inventory for various policies.

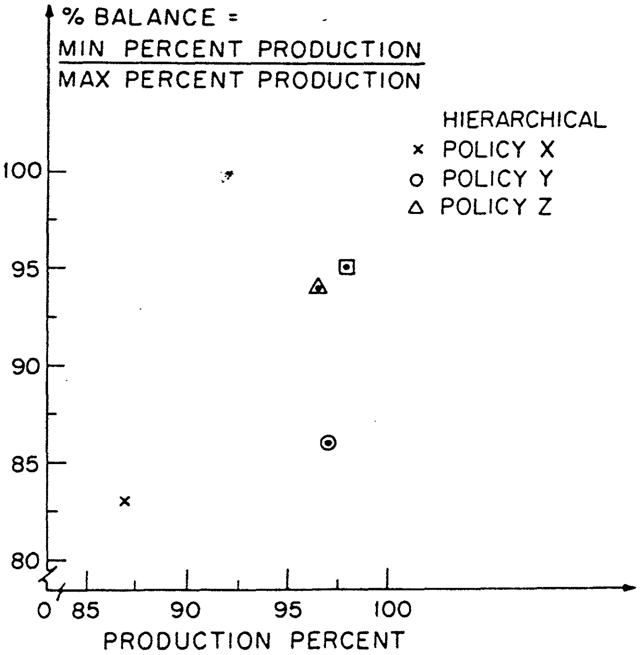

Fig. 15. Balance versus total production percentage for various policies.

ACKNOWLEDGMENT

We are grateful for the guidance of Mr. Mike Kutcher and Dr. Chacko Abraham of IBM. Ms. Susan Moller and Mr. David White of IBM provided important advice. Mr. J. Patrick Bevans of the C. S. Draper Laboratory was primarily responsible for writing the simulation, and we were assisted by M.I.T. students George Nikolau and Jean-Jacques Slotine.

REFERENCES

[1] R. Akella, J. P. Bevans, and Y. Choong, "Simulation of a flexible manufacturing system," Massachusetts Institute of Technology Laboratory for Information and Decision Systems Rep., to appear.

[2] S. B. Gershwin, R. Akella, and Y. Choong, "Short term scheduling of an automated manufacturing facility," Massachusetts Institute of Technology Laboratory for Information and Decision Systems Rep. LIDS-FR-1356, 1984.

[3] Special Issue on Data Driven Automation, IEEE Spectrum, May 1983.

[4] J. G. Kimemia, "Hierarchical control of production in flexible manufacturing systems," Massachusetts Institute of Technology Laboratory for Information and Decision Systems Rep. LIDS-TH-1215, 1982.

[5] J. G. Kimemia and S. B. Gershwin, "An algorithm for the computer control of production in flexible manufacturing systems, *IIE Trans.*, vol. 15, no. 4, pp. 353-362, Dec. 1983.

[6] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Operations Res.*, vol. 9, no. 3, pp. 383-387, 1961.