INT. J. PROD. RES., 1987, VOL. 25, NO. 6, 907–924

On-line scheduling of a robotic manufacturing cell with stochastic sequence-dependent processing rates

ABRAHAM SEIDMANN†

The problem of on-line production control for a robotic manufacturing cell producing parts of different types is analysed. The approach described in this paper expands the results of an earlier study so as to provide for sequence-dependent (deterministic or exponential) processing times at the cell. In addition, a novel production control strategy permitting temporary suspension of the cell's activities at certain decision epochs is formulated. Several numerical examples are given to illustrate the productive potential of this formulation. They seem to indicate the superiority of the new strategy presented here to earlier optimal control strategies in which the cell had to be active as long as it was unblocked.

1. Introduction

Most manufacturing facilities have very limited buffer capacity. Restricted local buffers between the work stations bound the in-process inventory levels and facilitate faster material handling capabilities. When the number of parts present at any station is limited, parts routed to that station may be blocked. The lack of downstream space forces the upstream station to shut down. Since the effect of blocking is significant in various industrial systems, several models have been proposed to deal with it. Surveys of transfer line models with finite buffers can be found in Buzacott and Hanifin (1978), Gershwin and Berman (1981), Suri and Diehl (1986), Gershwin (1985), and in Pollock *et al.* (1985). In general, analytic solutions of these models are available only for systems having a small number of stations and very few buffer spaces. Numerical solutions with varying degrees of computational complexity are available for the analysis of larger systems.

Other authors were concerned with the assignment of randomly arriving parts to parallel machines, each with its own limited input buffer space. Examples include Cinlar (1967), Lemoine (1975), Davis (1977), Ephremides *et al.* (1980), and Lin and Kumar (1984). Disney (1975), Lemoine (1977) and Stidham (1985) survey the methodological research in this area. Many of these studies stress the significant reduction in production flow times which are possible through the use of dynamic assignment strategies. These strategies are based on a rational function of the queue lengths at various downstream stations.

In this paper, we develop a new on-line production scheduling strategy for a robotic manufacturing cell operating with microprocessor control. Cellular manufacturing, widely used in industry, is a modern concept which is often adopted as a building block of larger facilities (Koren 1985). In such a system the cell processes R types of parts, and forwards them to R emanating assembly lines, each with a finite input buffer space (see, for example, Hutchinson and Wynne (1973)

Revision received November 1986.

† Graduate School of Management, The University of Rochester, Rochester, NY 14627, U.S.A.

or Seidmann *et al.* (1985)). The microprocessor controller at the cell has full knowledge of the instantaneous buffer occupancies at each downstream assembly line. The first model of this system was formulated by Hahne (1981). White's method of successive approximations was then used to develop optimal dynamic routing for a system having two ($R = 2$) unreliable lines. Later, Seidmann and Schweitzer (1984) extended this model to handle any number of parallel-reliable assembly lines.

Another related work is Yao and Shantihikumar (1986), which addresses the optimal input rate to a system of several parallel-reliable manufacturing cells.

In this paper, an expanded representation of the production control strategy for these systems is given.

It differs from Seidmann and Schweitzer (1984) in permitting *sequence-dependent* processing times at the cell, which can be either deterministic or *exponential*; and in formulating an improved optimal production control strategy which permits *temporary suspension* of the cell's processing activities at certain instances even though the buffers are not full. The decision problem considered at the cell is the determination of the optimal production sequence which aims at keeping all R assembly lines busy (i.e. not idle due to lack of input parts). The motivation for the problem considered here lies in its application to various dynamic routing problems in automatic manufacturing (Hodgson *et al.* 1985, Vinod and Solberg 1985), unitary robotic cells (Seidmann and Nof 1985), and flexible manufacturing systems (Yao and Buzacott 1985).

This paper is organized as follows: the manufacturing cell model is formulated and its functional equations are outlined in § 2. Several measures for evaluating the performance of the system under a given control scheme are formulated and computed in § 3. Numerical examples illustrating the structure of the optimal policy are presented in § 4. Section 5 concludes the paper with a proposal for efficient industrial application of this scheduling strategy.

2. Model formulation

2.1. Introduction

This section describes the major components of the system, the state space and the feasible set of decisions at each state; it also outlines the dynamic programming functional equations and the value iteration algorithm for solving them.

2.2. The manufacturing activities

Consider a manufacturing cell processing parts, one at a time, for $R(\geqslant 1)$ assembly or production lines. The mean time for the cell to process a part designated for line k is $1/\mu(k_0, k)$, $1 \leqslant k \leqslant R$, where k_0 denotes the preceding part type processed by the cell. These processing times can be either deterministic or exponentially distributed. Such sequence-dependent service times are commonly found in industry, where a single facility serves different kinds of emanating lines (French 1982). Part type k departing from the cell is admitted into the input buffer of line k. Line k (or, the first workstation on that line) has a production time which is exponentially distributed with a mean of $1/\lambda_k$. The input buffers for line k have space for $B_k \geqslant 1$ parts: one at the first workstation plus $B_k - 1$ spaces

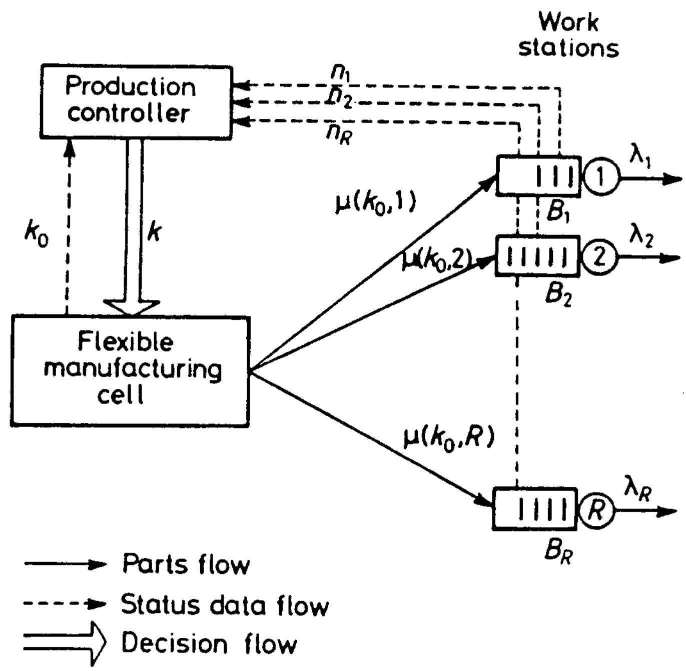

Figure 1. Schematic layout of the manufacturing system control with R finite local buffers. Status feedback information includes the buffer occupancies (n_i) and the cell set up (k_0),

for the queueing parts (Fig. 1). The line takes parts, one at a time, from its input buffer as long as parts are available. A shortage penalty of $C_k \geqslant 0$, $(1 \leqslant k \leqslant R)$ dollars per hour is incurred when line k is idle due to the absence of input parts. This penalty is a tangible surrogate for manufacturing facility idle time and the loss of production opportunities. The production control problem is to determine which part type to process next in order to *minimize* the expected shortage penalty cost per hour. This decision is made by an on-line controller having full knowledge of the buffer status at all lines.

This production control problem is formulated as a finite-state, infinite-horizon undiscounted semi-Markovian decision process (SMDP) (Howard 1971). The successive state occupancies of such a process are determined by markovian transition probabilities. The holding time in any state is described by a random variable that depends on the presently occupied state and on the successive state to which the next transition will be made. The process is markovian only at certain points belonging to the state space where transitions take place. The state space and the decision epochs (or the imbedding points) at which the system is modelled are defined in § 2.4.

2.3. The state-space

Let $n_k (1 \leqslant k \leqslant R)$ denote the number of parts at line k (either at the input buffer or possibly the one part being processed). The last part type produced by the cell is given by k_0. Let $k_0 = 0$ denote the initial (empty) state of the cell. The state space is defined by

$$\Omega = \{(\mathbf{n}, k_0) = (n_1, n_2, \ldots, n_R, k_0) | 1 \leqslant k_0 \leqslant R, \mathbf{o} \leqslant \mathbf{n} \leqslant \mathbf{B} | k_0 = 0, \mathbf{n} = \mathbf{o}\}$$

$$(o \leqslant n_i \leqslant B_i, i = 1, 2, \ldots, R)$$

The total number of states is

$$NS = \prod_{i=1}^{R} (B_i + 1)R + 1$$

These NS states are represented by the set of integers $\{1, 2, \ldots, NS\}$ via the mapping function $\psi: \Omega \to \{1, 2, \ldots, NS\}$, where

$$\psi(\mathbf{n}, k_0) = \begin{cases} 1 & \text{if } k_0 = 0 \text{ and } \mathbf{n} = \mathbf{o} \\ 2 + n_R + \sum_{i=1}^{R-1} n_{R-i} \prod_{j=1}^{i} (B_{R-j+1}) & + \prod_{i=1}^{R} (B_i + 1)(k_0 - 1) \\ \text{if } k_0 = 1, 2, \ldots, R \text{ and } \mathbf{o} \leqslant \mathbf{n} \leqslant \mathbf{B} & \end{cases} \quad (1)$$

2.4. State transitions and decisions epochs

In this time-continuous model, transitions between system states may occur at any time. When the cell is *busy* processing a given part type, its next decision epoch occurs at the completion of processing that part. When the cell remains *idle*, its next decision epoch will occur at the next end of processing event at one of the lines. The control of the system is carried out during such *decision epochs*, which may be of three sorts

(*a*) The initial cases: $k_0 = 0, 1, 2, \ldots, R$, $\mathbf{n} = \mathbf{o}$. In the startup (and transient) case the cell is not set up ($k_0 = 0$) and the buffers are empty ($\mathbf{n} = \mathbf{o}$). In all other initial cases the buffers are empty and the cell is set up to some specific part type ($k_0 = 1, 2, \ldots, R$). The cell must then decide which type $k (k = 1, \ldots, R)$ to process. The mean holding time at that state is $1/\mu(0, k)$.

(*b*) The unblocked case: $o \leqslant \mathbf{n} < \mathbf{B}$, $1 \leqslant k_0 \leqslant R$. The cell must decide whether to *remain idle* until the next change in the system state (denote this decision as $k = 0$) or to *process* part type k ($k = 1, \ldots, R$). This type must be selected only from these lines that have non-full buffers, i.e. from

$$D(\mathbf{n}) = \{k \mid n_k < B_k\} \quad (2)$$

The mean time that the system will hold in this state depends on the prior decision made by the cell:

$$= \begin{cases} 1/\mu(k_0, k) & \text{if part type } k \text{ is produced} \\ 1/\lambda(\mathbf{n}) & \text{if the cell remains idle} \end{cases}$$

where $\lambda(\mathbf{n}) = \sum_{i \in A(\mathbf{n})} \lambda_i$ (the cumulative processing rate of all *active* lines in $\mathbf{n}$), and the set of all active lines is

$$A(\mathbf{n}) = \{\mathbf{n}(n_1, n_2, \ldots, n_R) \mid o < n_i \leqslant B_i, 1 \leqslant i \leqslant R\}$$

(*c*) The blocked case: $\mathbf{n} = \mathbf{B}$. In the blocked state all buffers are full and the cell must remain idle until the next system state (the first buffer becomes empty). Then the cell can either remain idle until the next end of processing event at one of the other lines, or else start processing a part for one of the lines having a non-full buffer. The mean time that the system will hold in the blocked case is $1/\lambda(\mathbf{B})$, where

$$\lambda(\mathbf{B}) = \sum_{i=1}^{R} \lambda_i \tag{3}$$

2.5. *The functional equations*

Following Jewell (1963), the $(NS + 1)$ functional dynamic programming equations for minimizing the expected shortage penalty per time unit are

$$v(\mathbf{n}, k_0) = \min_{k=1,2,\ldots,R} \{q(\mathbf{n}, \mu(k_0, k)) - g/\mu(k_0, k) + v(\mathbf{e}^k, k)\}$$

$$(\mathbf{n} = \mathbf{o},\ k_0 = 0, 1, 2, \ldots, R) \tag{4}$$

$$v(\mathbf{o}, o) = 0 \tag{4'}$$

$$\begin{aligned} v(\mathbf{n}, k_0) = \min \Bigg[& \min_{k \in D(\mathbf{n})} \Bigg\{ q(\mathbf{n}, \mu(k_0, k)) - g/\mu(k_0, k) \\ & + \sum_{\mathbf{o} \leqslant \mathbf{j} \leqslant \mathbf{n}} P(\mathbf{j} \mid \mathbf{n}, \mu(k_0, k)) v(\mathbf{n} - \mathbf{j} + \mathbf{e}^k, k) \Bigg\}, \\ & \Bigg\{ q(\mathbf{n}, \lambda(\mathbf{n})) - g/\lambda(\mathbf{n}) + \sum_{i \in A(\mathbf{n})} (\lambda_i/\lambda(\mathbf{n})) v(\mathbf{n} - \mathbf{e}^i, k_0) \Bigg\} \Bigg] \end{aligned}$$

$$(\mathbf{n}, k_0) \in \Omega,\ \mathbf{n} \neq \mathbf{o} \text{ or } \mathbf{B},\ 1 \leqslant k_0 \leqslant R \text{ (unblocked case)} \tag{5}$$

$$v(\mathbf{n}, k_0) = -g/\lambda(\mathbf{n}) + \sum_{i=1}^{R} (\lambda_i/\lambda(\mathbf{n})) v(\mathbf{n} - \mathbf{e}^i, k_0)$$

$$\mathbf{n} = \mathbf{B},\ 1 \leqslant k_0 \leqslant R \text{ (blocked case)} \tag{6}$$

where we define:

$\mathbf{e}^k$ = the unit vector in the kth direction $(1 \leqslant k \leqslant R)$,
g = the *long-run* expected shortage penalty cost per unit time following the optimal control policy (the 'gain rate' (Howard 1971)),
$v(\mathbf{n}, k_0)$ = the *relative value* of state $(\mathbf{n}, k_0) \in \Omega$ (Howard 1971),
k = the *control variable* to be determined $(k = 1, 2, \ldots, R)$,
$P(\mathbf{j} \mid \mathbf{n}, \mu(k_0, k))$ = the joint transition probability that line i uses j_i of its n_i $(j_i \leqslant n_i)$ parts during the time that the cell processes one part type k—for all i $(i = 1, \ldots, R)$—given that the cell was initially at state k_0. [See Appendices A and B],
$q(\mathbf{n}, \mu(k_0, k))$ = the mean immediate shortage penalty cost incurred by all R lines, starting with state $\mathbf{n}$ and extending over the time interval needed by the cell to process one part type k. [See Appendices C and D],
$q(\mathbf{n}, \lambda(\mathbf{n})) = \sum_{i \in E(\mathbf{n})} C_i/\lambda(\mathbf{n})$, $\mathbf{n} \neq \mathbf{o}$, the mean immediate one-transition cost incurred by all R lines, starting with state $\mathbf{n}$ $(\mathbf{n} \neq \mathbf{o})$ and extending over the time interval to the next end of processing event in case that the cell decides to remain idle. The set of starving (idle) lines at state $\mathbf{n}$ is $E(\mathbf{n}) = \{\mathbf{n}(n_1, \ldots, n_R) \mid n_k = 0, \text{ for } 1 \leqslant k \leqslant R\}$.

The first eqn. (4) is used to fix the arbitrary additive constant of the relative constant of the relative value vector $v(n, k_0)$. Equation (5) represents two types of possible decisions by the cell: The *first* set of terms computes the relative value of state $(\mathbf{n}, k_0)$ if part type $k \in D(\mathbf{n})$ is processed, while the *second* set of terms computes the relative value of that state if the cell decides to remain idle until the next end of processing event at the lines. In the blocked case no starvation penalty is incurred since all R lines are active (eqn. (6)).

2.6. *Solution procedure*

The $(NS + 1)$ functional equations (eqns. (4)–(6)) are solved by the value iteration scheme developed by Schweitzer (1971). An initial guess of $v(\mathbf{n}, k_0) = 0$ for all states was used along with a step size equal to

$$\tau = \min\,[1/\lambda(\mathbf{B}), \{\min 1/\mu(k_0, k)\}] \qquad 0 \leqslant k_0 \leqslant R,\ 1 \leqslant k \leqslant R \tag{7}$$

The termination criteria was to exit the iterative process when g could be estimated within $\pm 0{\cdot}1\%$. The resulting solution is unique, since under any policy probability matrix of the associated semi-markovian process on Ω has a single ergodic class with few communicating transient states (Schweitzer 1971). In practice, using this solution procedure, the average memory requirements are linear in NS while time complexity is a quadratic function of NS.

3. The performance measures

Once the desired production control policy is determined, several performance measures can be computed. Let $k^*(\mathbf{n}, k_0)$ denote the optimal (or an arbitrarily prescribed) policy.

3.1. *The production rates of the lines*

Define a new semi-markovian process producing a unit 'reward' whenever the cell produces one part of type t: i.e.

$$q(\mathbf{n}, k_0) = \begin{cases} 1 & \text{if } k^*(\mathbf{n}, k_0) = t \quad (t = 1, \ldots, R),\ (\mathbf{n}, k_0) \in \Omega \\ & \quad k^*(\mathbf{n}, k_0) \in \{1, 2, \ldots, R\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Let $r_t (t = 1, \ldots, R)$ denote the expected production rate of line t following $k^*(\mathbf{n}, k_0)$. The analogous value equations for computing r_t are

$$\begin{aligned} v(\mathbf{n}, k_0) = {} & q(\mathbf{n}, k_0) - r_t/\mu(k_0, k^*(\mathbf{n}, k_0)) \\ & + \sum_{0 \leqslant \mathbf{j} \leqslant \mathbf{n}} P[\mathbf{j} \mid \mathbf{n}, \mu(k_0, k^*(\mathbf{n}, k_0))]\, v(\mathbf{n} - \mathbf{j} + \mathbf{e}^{(k^*(\mathbf{n}, k_0))}, k^*(\mathbf{n}, k_0)) \\ & \qquad k^*(\mathbf{n}, k_0) = 1, \ldots, R,\ (\mathbf{n}, k_0) \in \Omega \\ & \qquad \text{(cell processes part type } k^*(\mathbf{n}, k_0)) \end{aligned} \tag{9}$$

$$v(\mathbf{n}, k_0) = q(\mathbf{n}, k_0) - r_t/\lambda(\mathbf{n}) + \sum_{i \in A(\mathbf{n})} \frac{\lambda_i}{\lambda(\mathbf{n})} v(\mathbf{n} - \mathbf{e}^i, k_0)$$

$$k^*(\mathbf{n}, k_0) = 0,\ (\mathbf{n}, k_0) \in \Omega \text{ (cell remains idle)} \tag{10}$$

$$v(\mathbf{n}, k_0) = 0 \qquad \mathbf{n} = \mathbf{B} \tag{11}$$

Again, these equations are solved by the value iterative scheme of Schweitzer (1971). The computed long run expected reward per unit time (the 'gain rate') is r_t—which is also the processing rate of line t.

3.2. *The utilization of the lines*

The utilization of line t is given by

$$U_t = r_t/\lambda_t \qquad t = 1, \ldots, R \tag{12}$$

3.3. *The utilization of the cell*

Since the processing times at the cell are sequence-dependent, one cannot directly compute its utilization. Let CU denote the cell's utilization. To compute CU define another semi-markovian process similar to eqns. (9)–(11) with a one transition reward equal to

$$q(\mathbf{n}, k_0) = \begin{cases} 1/\mu(k_0, k^*(\mathbf{n}, k_0)) & \text{if } k^*(\mathbf{n}, k_0) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The gain rate used here will be CU (instead of r_t above). Clearly, $(1 - CU)$ is the fraction of time that the cell is idle (i.e. the fraction of time in which the cell is either blocked or is deliberately kept idle).

4. Numerical illustrations

This section illustrates the structures of the optimal policies for various operational modes. Consider a cell processing three part types for three distinct production lines ($R = 3$). Each line has two buffer spaces ($B_1 = B_2 = B_3 = 2$) and a production rate of 6 units per hour ($\lambda_1 = \lambda_2 = \lambda_3 = 6$). The shortage penalties are \$120, \$370 and \$210 per idle hour for lines 1, 2 and 3, respectively ($C_1 = 120$, $C_2 = 370$, $C_3 = 210$). The processing rate matrix of the cell ($=\mu(k_0, k)$) is given by Table 1. The state space of this problem has $(2 + 1)^3 3 + 1 = 82$ states.

Models having either *deterministic* or *exponentially distributed* processing times at the cell are examined below. Two optimal strategies were computed for each model in this study.

1. Initiated suspension strategy (ISS): in which the processing activities at the cell can be suspended temporarily at certain decision epochs having $\mathbf{n} < \mathbf{B}$. The rational of this policy is to save the capacity of the cell for some

k_0 \ k	1	2	3
0	15	15	15
1	21	10	5
2	10	21	10
3	5	10	21

Table 1. The cell's processing rates.

of the production lines which are blocked at that instant. The functional dynamic programming equations describing this strategy are given by eqns. (4)–(6).

2. Unsuspendable strategy (USS): in which the cell cannot remain idle unless all the buffers are full. The functional dynamic programming equations describing this strategy are similar to eqns. (4)–(6), except for (5) which is *replaced* by the following equation for the unblocked case

$$v(\mathbf{n}, k_0) = \min_{k \in D(\mathbf{n})} \left\{ q(\mathbf{n}, \mu(k_0, k)) - g/\mu(k_0, k) + \sum_{\mathbf{o} \leqslant \mathbf{j} \leqslant \mathbf{n}} P(\mathbf{j} \mid \mathbf{n}, \mu(k_0, k)) v(\mathbf{n} - \mathbf{j} + \mathbf{e}^k, k) \right\}$$

$$\mathbf{o} < \mathbf{n} < \mathbf{B}, \ 1 \leqslant k_0 \leqslant R \qquad (14)$$

Let ISS/DET and ISS/EXP denote initiated suspension strategy with deterministic and exponentially distributed processing times, respectively. Further, denote by USS/DET and USS/EXP the unsuspendable strategy with deterministic and exponentially distributed processing times, respectively. Table 2 presents several performance measures for these four models. Performance measures presented are: production lines throughputs (r_i), lines utilizations (U_i), the cell's utilization (CU), and the expected starvation penalty cost per hour (g).

Comparing the performance measures of the deterministic and the exponential models in Table 2 reveals that lower shortage penalties are associated with the models assuming deterministic processing times at the cell. This is an expected result that is attributable to the apparent reduction in the processing time variability.

Performance measures	i	ISS/DET	USS/DET	ISS/EXP	USS/EXP
r_i	1	2·155	2·783	2·278	3·132
	2	4·996	4·665	4·624	4·149
	3	4·482	4·194	4·134	3·693
U_i	1	0·359	0·464	0·380	0·522
	2	0·827	0·777	0·771	0·691
	3	0·747	0·699	0·689	0·615
CU		0·841	0·978	0·799	0·934
g		191·92	209·85	224·53	252·22

r_i throughput of production line i
U_i utilization of production line i
CU the utilization of the cell
g the expected starvation penalty cost per hour
ISS initiated suspension strategy
USS unsuspendable strategy
DET deterministic cell processing times
EXP exponentially distributed cell processing times

Table 2. Performance measures for various models. Identical buffer capacities are assumed in all 3 lines ($= B_1 = B_2 = B_3 = 2$).

Table 2 clearly indicates the superiority of the ISS mode over the USS mode: switching from ISS to USS *increased* the expected shortage penalty by 9·3% in the deterministic case and by 12·3% in the exponential case. Moreover, while USS resulted in *higher* cell utilization than ISS, it also resulted in *higher* shortage penalties. These results contra-indicate the common scheduling practice of continuously loading and operating the bottleneck stations in manufacturing networks. This special phenomenon was investigated by Meilijson and Yechiali (1977) in the case of isolated non-preemptive G/GI/1 queues without blocking.

Several representative system states are depicted along with their relative values and the associated optimal decisions in Table 3. For example, when the system is empty and the cell is not set up [$\mathbf{n} = (0, 0, 0)$, $k_0 = 0$, $\psi(\mathbf{n}, k_0) = 1$], the optimal decision in all four models is to process part type 2 ($k^*(0, 0, 0, 1) = 2$). The relative values function $v(\mathbf{n}, k_0)$ shows the marginal one time set up price that one may be willing to pay in order to start the system in state $(\mathbf{n}, k_0)$ rather than in state $(\mathbf{o}, o)$. Starting the system at a state having $v(\boldsymbol{n}, k_0) > 0$ is less favourable than starting it at state $(\mathbf{o}, o)$.

Table 3 shows that except for rare occasions, the optimal policy ($k^*(\mathbf{n}, k_0)$) is independent of the cell's processing time *distribution* if one follows the same strategy. Such similarity in the optimal discrete response reveals the robustness of these optimal policies to changes in the relative variability of the distribution function; it is clearly evident here since the deterministic and the exponential assumptions represent two extreme points in terms of the processing time variability. Similar pattern was detected in several other problems that we studied. An intuitive explanation of this feature, when found, could give useful insights into both this problem and other related issues.

Investigating the impact of the local buffer spaces on the cell's performance and on the structure of the optimal policy, we varied $\mathbf{B}$ from (2, 2, 2) to (4, 4, 4). Table 4 presents the performance measures computed for the four models assuming *increased* storage capacity ($\mathbf{B} = 4$). In this case, switching from ISS to USS increased the expected shortage penalty only by 1·05% at the deterministic case and by 7·2% at the exponential case.

These marginal differences in the expected shortage penalty between the two operating strategies are significantly smaller than those shown in Table 2. It seems, therefore that the relative benefit of the new ISS proposed here, becomes less pronounced as the probability of blocking decreases.

Finally, Figs. 2–5 depict the optimal policy for several decision epochs. The optimal decision regions for each state space rectangle are simply connected sets in all cases. The boundary curves between the decision regions are monotonically non-decreasing as a function of the buffer occupancies along the state space axes. In general, the decision regions are not convex and the boundary curves have no inflection points, i.e. they are not S-shaped.

Figure 2 shows that, in several cases using ISS, the cell is kept idle although it is not blocked (e.g. $k_0 = 1$ and $\mathbf{n} = (2, 2, 0)$ or ($\mathbf{n} = (2, 2, 1)$)). The same decision boundaries under USS are shown for comparison in Fig. 3. The decision boundaries in these figures have the same structure, except for the suspended production states.

The intercept in Fig. 5 indicates the number of parts one would be willing to see in station 3 before choosing to produce a part for station 2 which is kept starving. Note that only parts type 2 or 3 are selected, despite the fact that

State					$k^0(\mathbf{n}, k_0)$				$v(\mathbf{n}, k_0)$			
k_0	n_1	n_2	n_3	$\psi(\mathbf{n}, k_0)$	ISS/DET	USS/DET	ISS/EXP	USS/EXP	ISS/DET	USS/DET	ISS/EXP	USS/EXP
0	0	0	0	1	2	2	2	2	0	0	0	0
1	0	0	0	2	2	2	2	2	16·92	16·32	15·85	14·91
1	0	0	1	3	2	2	2	1	−12·38	−12·09	−10·85	−11·04
1	0	0	2	4	2	2	2	2	−35·99	−37·76	−34·55	−37·39
1	0	1	0	5	1	1	1	1	−21·84	−23·94	−21·78	−24·59
⋮				⋮								
1	2	1	1	24	2	2	2	2	−84·48	−83·87	−77·77	−76·56
1	2	1	2	25	2	2	2	2	−104·39	−105·60	−97·88	−98·42
1	2	2	0	26	0	3	0	3	−50·51	−52·05	−51·26	−48·17
1	2	2	1	27	0	3	0	3	−82·12	−79·27	−81·29	−69·98
1	2	2	2	28	0	0	0	0	−108·11	−109·02	−106·62	−103·73
2	0	0	0	29	2	2	2	2	−9·67	−9·33	−9·05	−8·52
2	0	0	1	30	2	2	2	2	−36·88	−35·32	−33·75	−32·15
⋮				⋮								
3	2	1	2	79	2	2	2	2	−104·39	−105·61	−97·88	−98·42
3	2	2	0	80	3	3	3	3	−99·98	−96·41	−97·24	−92·26
3	2	2	1	81	3	3	3	3	−112·41	−107·04	−109·08	−101·75
3	2	2	2	82	0	0	0	0	−115·04	−108·46	−112·95	−103·49

Table 3. Optimal policies and relative values for various models ($B_1 = B_2 = B_3 = 2$).

Performance measures		Policy			
	i	ISS/DET	USS/DET	ISS/EXP	USS/EXP
r_i	1	3·995	4·007	3·537	3·849
	2	5·791	5·782	5·615	5·505
	3	5·241	5·225	5·004	4·798
U_i	1	0·666	0·668	0·589	0·641
	2	0·965	0·963	0·936	0·917
	3	0·837	0·871	0·834	0·799
CU		0·978	0·994	0·899	0·970
g		79·52	80·36	107·79	115·56

Table 4. Performance measures for various models. Identical buffer capacities are assumed in all 3 lines ($= B_1 = B_2 = B_3 = 4$). Note the parameters and variables used here are defined in Table 2.

$n_1 = 0$. This figure exemplifies the impact of using a relatively small value of $\mu(3, 1)$ (i.e. high set-up times), on the general pattern of these optimal decision regions.

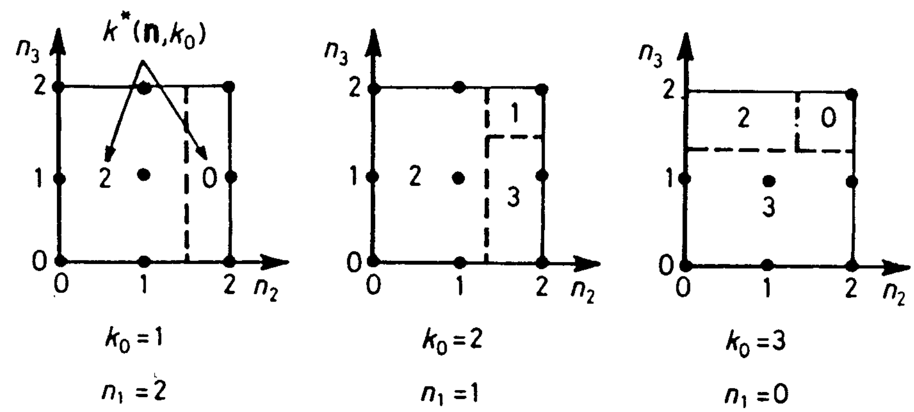

Figure 2. Optimal decision boundaries for ISS/EXP with buffer capacities $B_1 = B_2 = B_3 = 2$: $k^*(\mathbf{n}, k_0)$ denotes the optimal responses when the cell is set up to k_0 with n_i parts in buffer i.

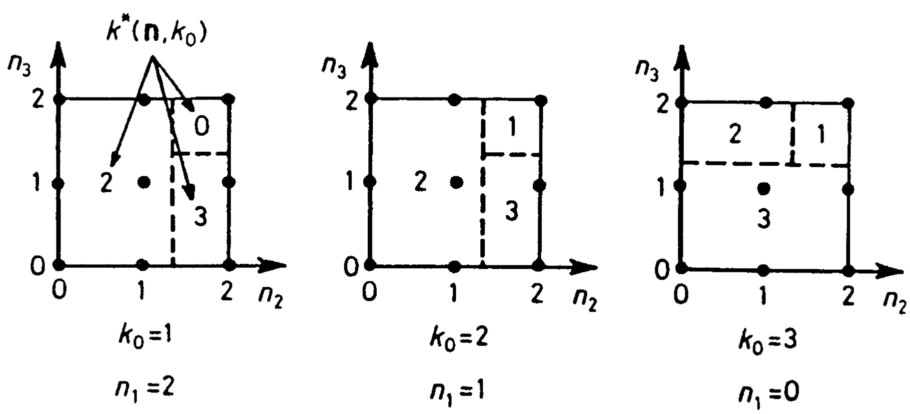

Figure 3. Optimal decision boundaries for USS/EXP with buffer capacities $B_1 = B_2 = B_3 = 2$: $k^*(\mathbf{n}, k_0)$ denotes the optimal responses when the cell is set up to k_0 with n_i parts in buffer i.

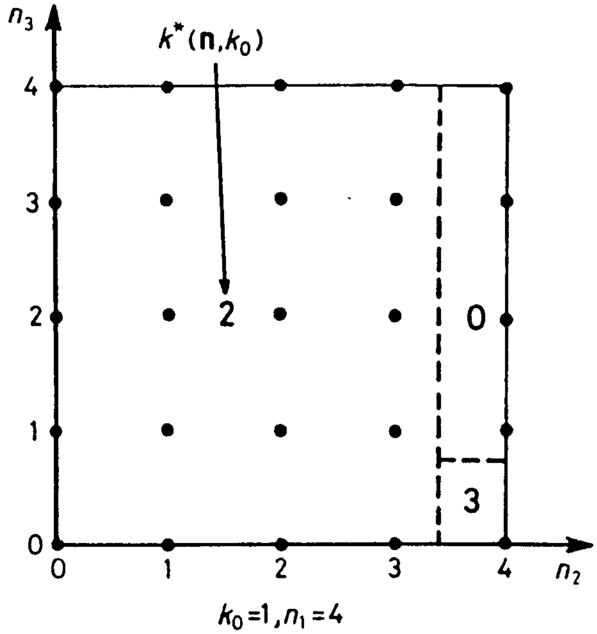

Figure 4. Optimal decision boundaries for initiated suspension strategies assuming exponentially distributed cell processing times (ISS/EXP) with buffer capacities $B_1 = B_2 = B_3 = 4$. $k^*(\mathbf{n}, k_0)$ denotes the optimal responses when the cell is set up to part type $k_0 = 1$. The buffer status of lines 1, 2 and 3 are given by $n_1 = 4$, n_2 and n_3, respectively.

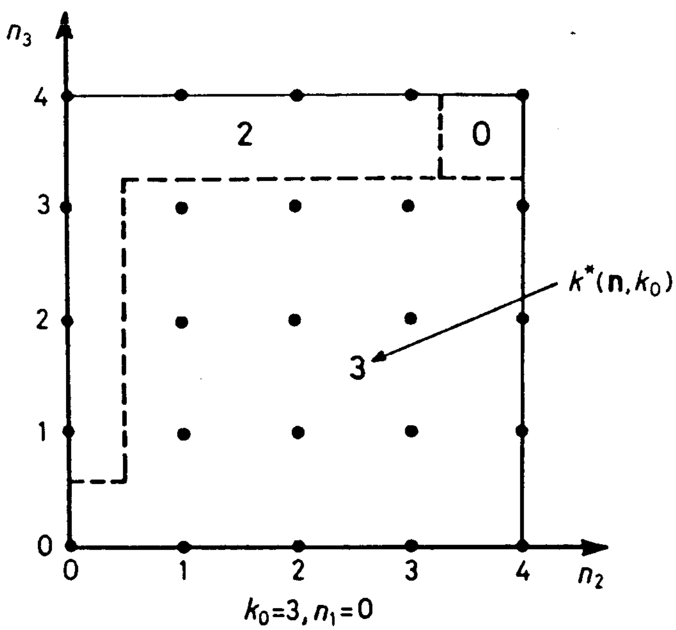

Figure 5. Optimal decision boundaries for initiated suspension strategy assuming exponentially distributed cell processing times (ISS/EXP) with buffer capacities $B_1 = B_2 = B_3 = 4$. $k^*(\mathbf{n}, k_0)$ denotes the optimal responses when the cell is set up to part type $k_0 = 1$. The buffer status of lines 1, 2 and 3 are given by $n_1 = 0$, n_2 and n_3, respectively.

5. Concluding remarks

A new manufacturing control strategy is presented in this paper. This strategy permits *temporary suspension* of the cell's processing activities at certain instances even though the buffers are not full. Analysing the structure of this optimal control strategy, and comparing it with the common scheduling practice of continuously loading and operating the cell, lead to the following observations:

(*a*) Employing the temporary suspension option leads to improved system performance, despite possible reduction in the expected utilization of the cell.

(*b*) The impact of the initiated temporary suspension of the cell becomes more influential with the increasing portion of the set ups in the processing times.

(*c*) Temporary suspension is significantly valuable for systems characterized by highly variable sequence-dependant processing times.

(*d*) The relative benefit of employing temporary suspension strategy, become less pronounced as the probability of blocking the cell decreases.

(*e*) The discrete responses of the optimal policy are (almost) independent of the cell's processing time distribution.

Explicit models are provided here for generating the optimal policy assuming *deterministic* or *exponential* cell processing times. Since processing activities at the cell are modelled as a semi-markovian decision process, various other distributions of the processing times at the cell can be accommodated—provided they are known and possess a rational Laplace transform. The computer code for computing the optimal policy and a variety of performance measures is available to readers on request from the author.

The current shortcomings of these models are that reliability and production-mix issues cannot be addressed at their present form. Ongoing research, however, indicates that this modelling approach also lends itself to extensions for optimally controlling robotic manufacturing systems with a more general topology.

Industrial implementation of the control scheme outlined above involves two phases. The first phase is the *off-line* policy computation which includes the data collection, the application of the semi-markovian decision model and the tabulation of $k^*(\mathbf{n}, k_0)$ as a function of $\psi(\mathbf{n}, k_0)$. This is a one-time step, best carried out on a mainframe or mini-computer. Memory requirements are *linear* in NS and time complexity is a quadratic function of NS. The second phase is the actual *on-line* industrial control. The instantaneous system state $(\mathbf{n}, k_0)$ is constantly monitored by the on-line controller. At each decision epoch it computes the mapping transformation $\psi(\mathbf{n}, k_0)$ and then immediately retrieves the tabulated optimal decision $k^*(\mathbf{n}, k_0)$. (The interested reader is referred to Seidmann and Schweitzer (1984) for further details).

Appendix A. The transition probabilities (exponential cell processing times)

The joint transition probability that line i, $1 \leqslant i \leqslant R$, uses up j_i of its n_i parts during the time the cell processes one part to line k, $1 \leqslant k \leqslant R$, is denoted by

$$P(\mathbf{j} \mid \mathbf{n}, \mu(k_0, k)) \quad \text{with} \quad \mathbf{n} = (n_1, n_2, \ldots, n_R) \quad \text{and} \quad \mathbf{j} = (j_1, j_2, \ldots, j_R)$$

If z denotes the actual time for the cell to process one part for line k ('holding time'), then its p.d.f. is given by

$$h_{k_0,k}(z) = \mu(k_0, k) \exp[-\mu(k_0, k) z], \quad z \geqslant 0 \qquad \text{(A 1)}$$

Next, let $f_i(j_i | n_i, z)$ define the *conditional probability* that line i having n_i parts consumes j_i parts during the time interval z. Hence, we conclude that in general

$$P(\mathbf{j} | \mathbf{n}, \mu(k_0, k)) = \int_0^{\infty} \prod_{i=1}^{R} f_i(j_i | n_i, z) h_{k_0, k}(z)\, dz \tag{A 2}$$

Since each station i consumes parts in a Poisson process during the time interval z

$$f_i(j_i | n_i, z) = \begin{cases} \delta_{j_i 0} & n_i = 0 \\ \dfrac{(\lambda_i z)^{j_i} \exp(-\lambda_i z)}{j_i!} & n_i \geq 1,\ 0 \leq j_i \leq n_i - 1 \\ 1 - \sum_{m=0}^{n_i - 1} f_i(m | n_i, z) & n_i \geq 1,\ j_i = n_i \\ 0 & \text{otherwise} \end{cases} \tag{A 3}$$

Combining the four cases of eqn. (A 3) and then inserting eqn. (A 1) into (A 2) leads to

$$P(\mathbf{j} | \mathbf{n}, \mu(k_0, k)) = \mu(k_0, k) \int_0^{\infty} dz \exp[-\mu(k_0, k) z]$$

$$\sum_{m1=1}^{S1} \sum_{m2=1}^{S2} \cdots \sum_{mR=1}^{SR} X(1, m1) X(2, m2) \ldots X(m, mR) z^{\tilde{l}} \exp(-\tilde{\omega} z) \tag{A 4}$$

where

$$\left.\begin{aligned} \tilde{l} &= \sum_{i=1}^{R} l(i, mi) \\ \tilde{\omega} &= \sum_{i=1}^{R} \omega(i, mi) \end{aligned}\right\} \tag{A 5}$$

If $n_i \geq 1$ and $j_i = n_i$

$$\left.\begin{aligned} & Si = n_i + 1 \\ & l(i, m) = \begin{cases} m - 1 & 1 \leq m \leq n_i \\ 0 & m = n_i + 1 \end{cases} \\ & \omega(i, m) = \begin{cases} \lambda_i & 1 \leq m \leq n_i \\ 0 & m = n_i + 1 \end{cases} \\ & X(i, m) = \begin{cases} -(\lambda_i)^{m-1}/(m-1)! & 1 \leq m \leq n_i \\ 1 & m = n_i + 1 \end{cases} \end{aligned}\right\} \tag{A 6}$$

If $n_i = 0$ then let

$$\left.\begin{aligned} Si &= 1 \\ l(i, 1) &= 0 \\ \omega(i, 1) &= 0 \\ X(i, 1) &= \delta_{j0} \end{aligned}\right\} \tag{A 7}$$

If $n_i > 1$ and $0 \leqslant j_i \leqslant n_i - 1$

$$\left.\begin{aligned} Si &= 1 \\ l(i, 1) &= j_i \\ \omega(i, 1) &= \lambda_i \\ X(i, 1) &= (\lambda_i)^{j_i}/j_i! \end{aligned}\right\} \qquad \text{(A 8)}$$

Evaluating the integral over z yields

$$P(\mathbf{j} \mid \mathbf{n}, \mu(k_0, k)) = \mu(k_0, k) \sum_{m1=1}^{S1} \dots \sum_{mR}^{SR} (\tilde{l})! X(1, m1) \dots X(R, mR))/(\mu(k_0, k) + \tilde{\omega})^{1+l} \qquad \text{(A 9)}$$

Appendix B. The transition probabilities (deterministic cell processing times)

In the case of deterministic cells' processing times the p.d.f. of z, the actual cell's processing time ('holding time'), is given by

$$h_{k0,k}(z) = \begin{cases} 1 & z = 1/\mu(k_0, k) \\ 0 & \text{otherwise} \end{cases} \qquad \text{(B 1)}$$

Inserting $h_{k0,k}(z)$ into eqn. (A 2) leads after several computations to the final result

$$P(\mathbf{j} \mid \mathbf{n}, \mu(k_0, k)) = \sum_{m1=1}^{S1} \sum_{m2=1}^{S2} \cdots \sum_{mR=1}^{SR} X(1, m1) X(2, m2) \dots X(R, mR)[1/\mu(k_0, k)]^l \exp[-(\tilde{\omega}/\mu(k_0, k))] \qquad \text{(B 2)}$$

The values of $(S1, \dots, SR)$, $(m1, \dots, mR)$, $(X(1, m1), \dots X(R, mR))$, $\tilde{l}$ and $\tilde{\omega}$ are given in Appendix A.

Appendix C. The one-transition expected cost (exponentially distributed cell processing times)

The one-transition expected cost is given by

$$q(\mathbf{n}, \mu(k_0, k)) = \sum_{i=1}^{R} C_i U(n_i, \mu(k_0, k)) \qquad \text{(C 1)}$$

where $U(n_i, \mu(k_0, k))$ is the mean duration, within the time interval when the cell is processing one part of type k, that line i is starved, given that this time interval began with n_i parts at line i. Extending the analytical framework of Seidmann and Schweitzer (1984) leads to the final result

$$q(\mathbf{n}, \mu(k_0, k)) = \sum_{i=1}^{R} \frac{C_i}{\mu(k_0, k)} \left(\frac{\lambda_i}{\lambda_i + \mu(k_0, k)} \right)^{n_i} \qquad 1 \leqslant i \leqslant R,\ 0 \leqslant n_i \leqslant B_i \qquad \text{(C 2)}$$

Appendix D. The one-transition expected cost (deterministic cell processing times)

Following the definitions in Appendix C, the one-transition expected cost is given by

$$q(\mathbf{n}, \mu(k_0, k)) = \sum_{i=1}^{R} C_i U(n_i, \mu(k_0, k)) \tag{D1}$$

If $n_i \geqslant 1$, let t denote the time for line i to consume n_i of its parts. Since the line's processing times are exponential, it means that t has an Erlang probability density. For a given $t(t < 1/\mu(k_0, k))$, line i starves for $(1/\mu(k_0, k) - t)$ time units. This leads to

$$U(n_i, \mu(k_0, k)) = \int_0^{1/\mu(k_0, k)} \frac{\lambda_i^{n_i}}{(n_i - 1)!} \times \exp(-\lambda_i t) t^{n_i - 1} \left(\frac{1}{\mu(k_0, k)} - t \right) dt \qquad t \geqslant 0 \tag{D2}$$

To compute $U(n_i, \mu(k_0, k))$ define the following auxiliary function

$$\begin{aligned} UX(t^{n_i}) &= \int_0^{1/\mu(k_0, k)} t^{n_i} \exp [(-\lambda_i t)\, dt] \\ &= \frac{1}{\lambda_i} \left(\frac{1}{\mu(k_0, k)} \right)^{n_i} \exp [-(\lambda_i/\mu(k_0, k))] + \frac{n_i}{\lambda_i} UX(t^{n_i - 1}) \qquad n_i \geqslant 1 \end{aligned} \tag{D3}$$

For $t^{n_i} = 1$ we get the seed of the integrals

$$UX(1) = \frac{1}{\lambda_i} \{1 - \exp [-(\lambda_i \mu(k_0, k))]\} \tag{D4}$$

Performing the integration leads to the final result

$$U(n_i, \mu(k_0, k)) = \begin{cases} \dfrac{\lambda_i^{n_i}}{(n_i - 1)!} \left[\dfrac{1}{\mu(k_0, k)} UX(t^{n_i - 1}) - UX(t^{n_i}) \right] & n_i \geqslant 1 \\ \dfrac{1}{\mu(k_0, k)} & n_i = 0 \end{cases} \tag{D5}$$

L'analyse porte sur le problème du contrôle de la production en direct pour une cellule de fabrication robotique produisant différents types de pièces. L'approche décrite dans le présent article développe les conclusions auxquelles avait abouti une étude antérieure de sorte à prévoir les temps de traitement (déterministes ou exponentiels) dépendant des séquences au niveau de la cellule. En outre, une stratégie innovatrice de contrôle de la production permettant de suspendre provisoirement les activités cellulaires lors de certaines périodes de décision est formulée. Plusieurs exemples numériques illustrent le potentiel de productivité de cette formulation et semblent indiquer que cette nouvelle stratégie est supérieure aux stratégies de contrôle optimales antérieures dans lesquelles la cellule devait être active tant qu'elle n'était pas bloquée.

Es wird das Problem der On-line-Regelung und Steuerung von Fertigungsverfahren an einer automatisierten flexiblen Fertigungszelle untersucht, die Teile verschiedener Art herstellt. Der hier beschriebene Lösungsweg erweitert die Ergebnisse einer früheren Untersuchung, und ermöglicht es damit, auch arbeitsfolgebedingte (deterministische oder exponentielle) Bearbeitungszeiten an der Zelle zu berücksichtigen. Außerdem wird eine neue Fertigungsstrategie vorgeschlagen, die zu gewissen Zeitpunkten, an denen Entscheidungen getroffen werden müssen, eine zeitweilige Unterbrechung der Zellentätigkeit erlaubt. An Hand einiger numerischer Beispiele wird das Fertigungspotential einer solchen neuformulierten Strategie veranschaulicht. Die Ergebnisse lassen darauf schließen, daß die hier vorgestellte neue Strategie früheren Steuerstrategien überlegen ist, denen zufolge die Fertigungszelle in Betrieb sein soll, solange sie nicht besetzt ist.

References

Buzacott, J. A., and Hanifin, L. E., 1978, Models of transfer lines with inventory banks—a review and comparison. *AIIE Transactions*, **10**, 197–209.

Cinlar, E., 1967, Decomposition of a semi-markovian process under a state dependent rule. *SIAM Journal of Applied Mathematics*, **15**, 252–263.

Davis, E., 1977, Optimal control of arrivals to a two-server queueing system with separate queues. Ph.D. dissertation, Program Opera. Research, North Carolina State University, Raleigh, U.S.A.

Disney, R. L., 1975, Random flow in queueing networks: a review and critique. *AIIE Transactions*, **7**, 268–288.

Ephremides, A., Varaiya, P., and Walrand, J., 1980, A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, **25**, 690–693.

French, S., 1982, *Sequencing and Scheduling* (London: Ellis Horwood Ltd.).

Gershwin, S. B., 1985, Representation and analysis of transfer lines with machines that have different processing rates. MIT Laboratory for Information and Decision Systems, Report LIDS: 35-433.

Gershwin, S. B., and Berman, O., 1981, Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. *AIIE Transactions*, **13**, 2–11.

Hahne, E. L., 1981, Dynamic routing in an unreliable manufacturing network with limited storage. MIT Laboratory for Information and Decision Systems, Report No. LIDS-TH-1063.

Hodgson, T. J., King, R. E., Monteith, S. K., and Schultz, S. R., 1985, Developing control rules for an AGVS using Markov decision processes. Technical Report 85-17. Department of Industrial Engineering, North Carolina State University, Raleigh, U.S.A.

Howard, R. A., 1971, *Dynamic Probabilistic Systems, Vol. II: Semi-Markov and Decision Processes* (New York: John Wiley).

Hutchinson, G. K., and Wynne, B. E., 1973, A flexible manufacturing system. *Industrial Engineering*, **5**, 10.

Jewell, W. J., 1963, Markov-renewal programming: I and II. *Operations Research*, **11**, 938–948, 949–971.

Koren, Y., 1985, *Robotics for Engineers* (New York: McGraw-Hill).

Lemoine, A. J., 1975, A queueing system with heterogeneous servers and autonomous traffic control. *Operations Research*, **23**, 681–686.

Lemoine, A. J., 1977, Networks of queues—a survey of equilibrium analysis. *Management Science*, **24**, 464–481.

Lin, W., and Kumar, P. R., 1984, Optimal control of a queueing system with two heterogeneous servers. *IEEE Transactions on Automatic Control*, **29**, 696–703.

Meilijson, I., and Yechiali, U., 1977, On optimal right-of-way policies at a single-server station when insertion of idle time is permitted. *Stochastic Processes and Their Applications*, **6**, 25–32.

Pollock, S. M., Birge, J. R., and Alden, J. M., 1985, Approximation analysis for open tandem queues with blocking: exponential and general service distribution. Technical Report 85-30, Department of Industrial & Operations Engineering, University of Michigan, U.S.A.

SCHWEITZER, P. J., 1971, Iterative solution of the functional equations of undiscounted Markov renewal programming. *Journal of Mathematical Analysis and Applications*, **34**, 495–501.

SEIDMANN, A., and SCHWEITZER, P. J., 1984, Part selection policy for a flexible manufacturing cell feeding several production lines. *IIE Transactions*, **16**, 355–362.

SEIDMANN, A., SCHWEITZER, P. J., and NOF, S. Y., 1985, Performance evaluation of a flexible manufacturing cell with random multiproduct feedback flow. *International Journal of Production Research*, **23**, 1171–1184.

STIDHAM, S. JR., 1985, Optimal control of admission to a queueing system. *IEEE Transactions Automatic Control*, **30**, 705–713.

SURI, R., and DIEHL, G. W., 1986, A variable buffer-size model and its use in analyzing closed queueing networks with blocking. *Management Science*, **32**, 206–224.

TENENBAUM, A., 1985, Dynamic scheduling in FMS network with limited buffers. Research Report, Department of Industrial Engineering, Tel-Aviv University, Israel.

VINOD, B., and SOLBERG, J. J., 1985, The optimal design of flexible manufacturing systems. *International Journal of Production Research*, **23**, 1141–1151.

YAO, D. D., and BUZACOTT, J. A., 1985, Modelling the performance of flexible manufacturing systems. *International Journal of Production Research*, **23**, 945–959.

YAO, D. D., and SHANTHIKUMAR, J. G., 1986, The optimal input rate to a system of manufacturing cells. Working paper Ms-32, School of Business Administration, University of California, Berkeley, U.S.A.