

Theory and Methodology

Regenerative pull (Kanban) production control policies

Abraham SEIDMANN

W.E. Simon Graduate School of Business Administration, University of Rochester, Rochester, NY 14627, USA

Abstract: A production system consists of a set of parallel robotic cells manufacturing parts for several distinct work stations. The stations order parts from these cells and withdraw parts from their buffers only at the rate and at the time of consumption. The desired decision vector provides for the instantaneous number of cells assigned to produce parts for each work station. Two novel tractable and optimal regenerative pull ('Kanban') control policies are formulated: one policy minimizes the weighted starvation penalty, while the other maximizes the weighted throughputs per unit time. Following these regenerative policies the production schedules are re-evaluated at each decision epoch to mitigate the effects of processing time variability.

Several important properties regarding the inherent interaction between the structure of the optimal policy, the performance of the system and the desired allocation of productive capabilities among the manufacturing resources are exemplified. It is shown that the optimal policy attempts to marginally assign as much of the cells capacity as possible to certain critical part types. Substantial changes in the structure of the optimal policy, resulting either from incrementing the number of cells or from increasing their capacity, are also identified. More generally, attention is drawn to the qualitative behavior of the optimal pull control policy in certain manufacturing systems with stochastic processing rates.

Keywords: Production, computers, manufacturing industries, Markov decision programming, queues

1. Introduction

A robotic manufacturing cell is a part of a production system designed to fabricate families of parts having similar geometry and process requirements. In order to produce one of the members of the part family at the cell, operations are added and deleted according to the specified attributes of that particular part. Efficient flow of work parts through the cell is facilitated by the use of industrial robots, flexible fixtures, and programmable machine tools. Jablonowski (1987) describes such a manufacturing cell recently in-

stalled at the McDonnell Douglas Astronautics Co(MDAC) in St. Charles, Montana. It includes two production machine tools, a coordinate measuring machine (CMM) and a single automated guided vehicle. All the cell activities are orchestrated by a single supermini computer. For an overview of cellular manufacturing technology see Black (1983), Groover and Zimmers (1984), Koren (1985), Nof (1985), Seidmann, Schweiter and Nof (1985) and recent articles in the *American Machinist & Automated Manufacturing* journal.

The system at the focus of interest in this paper consists of a set of robotic cells manufacturing one part at a time for several distinct work stations (Wilheim et al., 1986). Each of these work stations

Received July 1986; revised June 1987

has a limited input buffer capacity. In this dynamic flow environment, buffer storage is used to improve the overall productivity by decoupling the cells and the lines (Okamura and Yamashina, 1979; Gershwin and Berman, 1981). Unlike a conventional job shop, the limited work in processing storage capacity in automated systems requires stringent flow control policies to minimize the blocking effects and to attain the desired operational objective (Buzacott and Hanifin, 1978). Hahne (1981) studied this problem in an unreliable system having one cell feeding two work stations.

The seminal papers by Kimura and Terada (1981) and by Monden (1981) have pointed to the practical significance of using the interstage buffer stocks as a mean for driving a decentralized pull production control system. Monden (1981) outlines the various Kanbans (or production and withdrawals cards) used in the Toyota Motor Co. plants in Toyota-Shi, Japan. In the case of a serial and automated production system no cards are used and the inprocess inventories are limited by the size of the interstage buffers denoted as 'electric Kanbans', or as 'full work-systems'. Limit switches turn off the preceding machines when the buffers are full (the 'target levels'). Thereby, a bounded quantity of input parts is placed in each station, preventing unnecessary processing in the preceding stage. The production control problem becomes more involved in those flexible facilities where several succeeding work stations compete for the same preceding cell capacity (Figure 1). Such a case is presented in our paper along with two applicable control policies. Mathematical formulation of the Toyota Kanban system is given by Kimura and Terada (1981). A deterministic Kanban model for assembly-tree-plant structure is studied by Bitran and Chang (1987). The desired

buffer allocation problem for a pull production system with variable production rates, machine breakdowns and demand fluctuations is studied by So and Pinault (1987).

Yao and Shanthikumar (1987) studied the optimal input rate of parts into a system of manufacturing cells. Their objective was to maximize the total throughput of the cells without causing excessive blocking. Recent studies related to manufacturing cells with limited buffers include Dubois (1983), Stidham and Altiok (1984), Sarker (1984), Costa and Garetti (1985), Gershwin (1985) and Herer (1985). Watanbe and Sakamoto (1985) used computer simulation to evaluate several heuristic policies for on-line scheduling of adaptive control machine tools. Seidmann and Schweitzer (1984) used Markov Renewal Programming concepts to develop a part selection policy for a *single* manufacturing cell producing different parts for several distinct work stations. Other models of adaptive part routing were analyzed by Yao and Buzacott (1985, 1986ab). Elsayed and Hwang (1986) presented a Markov chain approach to investigate a two-stage production system where a common buffer storage is provided between the two stages. Greene and Sadowski (1986) applied a mixed integer program for loading and scheduling a system of multiple flexible manufacturing cells; their objective function was based on minimizing makespans or mean lateness and flowtimes. A comprehensive survey paper discussing these issues has been written by Kusiak (1986).

The study reported here was motivated by investigating a glass lens manufacturing plant. This plant produces several hundred varieties of glass lenses for prescription spectacles. These varieties are classified into about one dozen technological groups according to the desired manufacturing processes and the specified optical properties. The

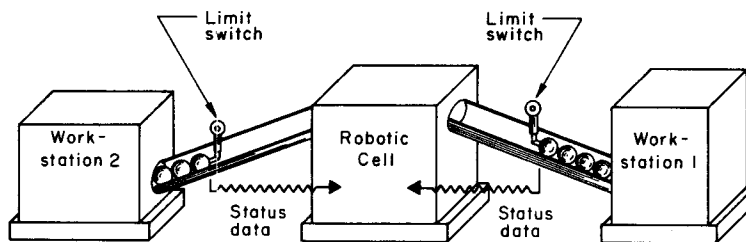


Figure 1. An 'electric Kanban' or a full work system (Monden, 1981). A single robotic cell feeds two workstations. Workstations 1 and 2 have 4 and 3 Kanbans, respectively.

machines used at the cells are general purpose coarse grinder. These cells comprise the first production phase. The second phase is carried out by dedicated high-precision fine grinding work stations. Each one of these work stations is setup periodically to handle only a certain group of the parts variety. Limited buffers were imposed between the cells and the work stations. These buffers constitute the 'physical Kanbans' discussed above. The decision variable is the allocation of product types to the individual cells such that the work stations will be best utilized while cells blocking is controlled.

The process time for each individual lens is not deterministic due to the control structure being used. Typical coefficient of variation on these CNC adaptive control grinders is about 0.25; the differences between the processing time requirements of the various product types belonging to the *same* group generate the appearance of randomly distributed processing times with an averaged coefficient of variation close to one. Similar observations have lead others to various successful applications of the stochastic-time control approach to traffic (Bullen and Hummon, 1987), data communications (Kleinrock, 1976), and manufacturing systems (Yao and Buzacott, 1986). In addition, this approach is helpful for layout planning and evaluation concerning the allocation of machines to cells and work stations or to the determination of Kanban allocation priorities.

During the plant operation the workstations pull their input parts from the cells. According to this pull system the quantity produced by the cell is not based on estimated future requirements but simply serves to replace the parts actually withdrawn by the workstations. In order to replenish these parts the production control system has to determine the manufacturing assignments for each one of the parallel cells. As a result it computes a decision vector that determines the number of cells assigned to each one of the available part types. The cells are not allowed to remain idle when there exists an empty input buffer at the stations. Furthermore, the total number of cells assigned to each part type can't exceed the current number of empty buffers at the designated station (for that part type).

Two tractable and optimal discrete pull control policies are formulated for evaluating part selection policies at the cells: one policy minimizes the

weighted starvation penalty, and the other maximizes the weighted throughputs per unit time. In Section 2 the functional equations of the optimization model are presented. The computational scheme of major performance measures is provided in Section 3. Several numerical examples illustrating the structure of the optimal policy are given in Section 4.

2. The manufacturing system model

2.1. General structure

The model represents a system which consists of $S (\geq 1)$ parallel and identical robotic manufacturing cells producing parts for $R (\geq 1)$ work stations. Each part is first processed at one of the cells and then it is routed for additional processing at one of the work stations. There is always a supply of raw production units available to the cells and the processing time for each part at the cell depends on its designated downstream station. The time required for a *cell* to process one part for station i is exponentially distributed with a mean of $1/\mu_i$, $1 \leq i \leq R$. Station i has a processing time which is exponentially distributed with a mean of $1/\lambda_i$. Each station has a finite local input buffer with room for $B_i (\geq 1)$ parts, one in processing plus up to $B_i - 1$ on queue. Transport times between the cells and the workstations are assumed to be negligible.

Parts leave the system after being processed at the stations. Station i , when idle because of the absence of parts, incurs a shortage (starvation) penalty of C_i dollars per hour idle. Production control decisions are made before each part is processed by the cell and are made with a full knowledge of the instantaneous buffer status at all the R stations. These assumptions are valid for many automated production systems such as PCB assemblies, precision machining, canning, food processing, injection molding, and packaging (Schonberger, 1983; Seidmann, Arbel and Shapira, 1984; Holmqvist, 1984; Nof, 1985). Figure 2 depicts the schematic layout of such a robotic manufacturing system.

2.2. State space

Let $n = \{(n_1, n_2, \dots, n_R) | 0 \leq n_i \leq B_i, 1 \leq i \leq R\}$ be the state of the system, where $n_i (i =$

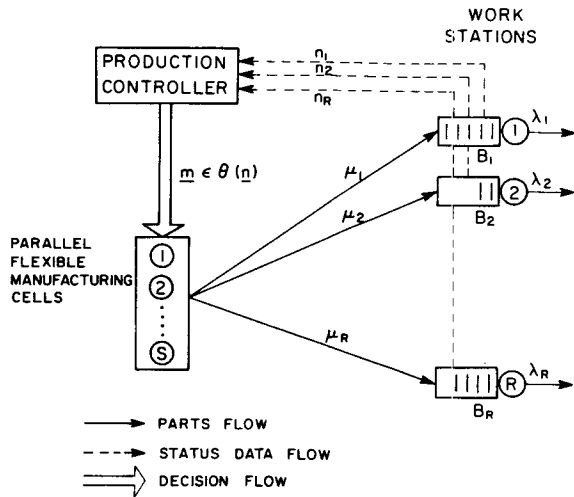


Figure 2. Schematic layout of the robotic manufacturing system with several parallel cells.

1, 2, ..., R) denotes the number of parts on queue at station i plus possibly one part in process. The state space is then

$$\Omega = \{ \mathbf{n} = (n_1, n_2, \dots, n_R) \mid 0 \leq n_i \leq B_i \text{ for } 1 \leq i \leq R \}. \tag{1}$$

The total number of states is given by

$$NS = \prod_{i=1}^R (B_i + 1). \tag{2}$$

Here the set of all *active* stations at state \mathbf{n} is given by

$$A(\mathbf{n}) = \{ i \mid 0 < n_i \leq B_i, 1 \leq i \leq R \}. \tag{3}$$

The cumulative production rate of all active stations at state \mathbf{n} is

$$\lambda(\mathbf{n}) = \sum_{i \in A(\mathbf{n})} \lambda_i. \tag{4}$$

2.3. State transitions

Using a time-continuous model, transitions between system states start and terminate at an *end of processing event*—either at one of the *cells* or at one of the *stations*. Operating this so-called *regenerative* (or continuous) flow control policy the production schedule is re-evaluated at each decision epoch to mitigate the effects of the processing time variability and to obtain the best possible resource utilization strategy. It seems, therefore,

that the cells operate with a preemptive sequencing scheme such that each time a part processing (re)starts at a cell, a new processing time (μ_i) is sampled. This is a reasonable assumption, since the memoryless property of the exponential distribution implies that operating with *preempt-repeat* scheme with resampling yields the same system performance as with *preempt-resume* which assumes that the processing of a preempted job can be taken up where it was preempted without any loss of time (Conway et al., 1967; Hahne, 1981). In most manufacturing systems, regenerative control presents the expected upper bound on system performance for a given industrial configuration. Our numerical experimentations indicate that implementing regenerative control results in a relatively *small* number of preempt-replace instances. This observation is explained by two typical attributes of the system: (i) the system tends to evolve from one state to a neighboring state (i.e., no abrupt changes), and (ii) the optimal decision regions are simply closed-connected sets. (See the Numerical Examples in Section 4.)

The immediate applicability of regenerative control policies is also clear in such cases as dynamic routing in buffered communication systems (i.e., packet switching networks) or in memory management units of large computer systems (i.e., interleaved access memory banks).

The cells can be either blocked or unblocked, depending on the state of the buffers. Since the cells in this production system can be either blocked or unblocked the decision epochs are defined accordingly.

(a) The unblocked case: $\mathbf{n} \neq \mathbf{B}$

In such a decision epoch, we must determine what set of parts the cells are to produce next. That decision is a *vector* $\mathbf{m} = (m_1, m_2, \dots, m_R)$ where m_i denotes the number of cells assigned to produce parts for station i . The number of cells assigned to each station can't exceed the number of empty buffers: $0 \leq m_i \leq B_i - n_i$, and clearly it satisfies:

$$\sum_{i=1}^R m_i \leq S.$$

We impose the rule that no cells remain idle if there exists an empty buffer which can expect a

new part. It means that:

$$\sum_{i=1}^R m_i = \min \left[S, \sum_{i=1}^R (B_i - n_i) \right].$$

The action space (or the admissible decision space) in state n is given by

$$\theta(n) \left\{ m \begin{cases} 0 \leq m_i \leq B_i - n_i & \text{and } \sum_{i=1}^R m_i = S, \\ \text{if } \sum_{i=1}^R (B_i - n_i) > S, \\ m_i = B_i - n_i \\ \text{otherwise,} \end{cases} \right. \quad (5)$$

$i = 1, 2, \dots, R$.

Note that $m_i = B_i - n_i$ when the number of cells exceeds the total number of available buffer spaces. Then, the number of cells that will be kept idle is equal to:

$$S - \sum_{i=1}^R (B_i - n_i).$$

The cumulative production rate of the cells is

$$\mu(m) = \sum_{i=1}^R m_i \mu_i. \quad (6)$$

The mean time that the system will spend in state $n \in \Omega$ following decision $m \in \theta(n)$ is

$$\frac{1}{(\lambda(n) + \mu(m))}.$$

The probability that the next end of processing event following state (n) and decision $m \in \theta(n)$ will occur at a cell producing part type i ($i = 1, 2, \dots, R$), is given by

$$\frac{m_i \mu_i}{\lambda(n) + \mu(m)}.$$

Similarly, the probability that the next end of processing event following state (n) and decision $m \in \theta(n)$, will occur at station i ($i = 1, 2, \dots, R$) is

$$\frac{\lambda_i}{\lambda(n) + \mu(m)}.$$

The mean cost of one transition is denoted by $q(n, m)$. It is the mean shortage penalty cost incurred by all R stations, starting with state n , decision m , and extending over the time interval until the next end of processing event. The set of

starving (empty) stations at state n is

$$E(n) = \{ i \mid n_i = 0, 1 \leq i \leq R \}. \quad (7)$$

From this definition, the following holds:

$$q(n, m) = \sum_{i \in E(n)} \frac{C_i}{\lambda(n) + \mu(m)}. \quad (8)$$

(b) *The blocked case: $n = B$*

When all the buffers at the stations are full, the cells must remain idle (blocked) until the next system state (the first buffer becomes empty); then one of the cells begins to process a part for that station. All other ($S - 1$) cells will be kept idle. The mean holding time in the blocked case is equal to

$$1/\lambda(B) = 1/\sum_{i=1}^R \lambda_i. \quad (9)$$

2.4. The functional equations

The production control problem is stated here as a Semi-Markovian Decision Program (SMDP). Given a manufacturing system as described above, we wish to specify a production plan that *minimizes* the expected shortage penalty cost per unit time. (The Appendix presents the functional equations for *maximizing* the weighted throughputs). This leads to the following NS + 1 dynamic programming functional equations (Jewell, 1963; Cinlar, 1967; Howard, 1971):

$$v(n) = \min_{m \in \theta(n)} \left\{ q(n, m) - g/(\mu(m) + \lambda(n)) + \sum_{i \in A(n)} \frac{\lambda_i}{\lambda(n) + \mu(m)} v(n - e^i) + \sum_{i=1}^R \frac{m_i \mu_i}{\lambda(n) + \mu(m)} v(n + e^i) \right\},$$

$$n_i < B_i, \quad (\text{unblocked case}) \quad n \in \Omega, n \neq B, \quad (10)$$

$$v(B) = -g/\lambda(B) + \sum_{i=1}^R \frac{\lambda_i}{\lambda(B)} v(B - e^i), \quad (\text{blocked case}) \quad n = B, \quad (11)$$

$$v(0) = 0, \quad (12)$$

where we define

\mathbf{B} = the state vector of the buffers ($\mathbf{B} \triangleq (B_1, B_2, \dots, B_R)$),

$\tau = \text{Min}[1/(\mu(\mathbf{m}) + \lambda(\mathbf{n}))]$,

$\mathbf{m} \in \theta(\mathbf{n})$,

$\mathbf{n} \in \Omega$,

\mathbf{e}^k = the unit vector in the k -th direction, $1 \leq k \leq R$,

g = the long-run expected shortage penalty cost per unit time following the optimal production control policy (the 'gain rate' of the SMDP),

\mathbf{m} = the decision vector whose optimal value is to be determined.

In equation (10), $q(\mathbf{n}, \mathbf{m})$ is the one-transition immediate cost and $g/(\mu(\mathbf{m}) + \lambda(\mathbf{n}))$ is the time average cost incurred. The first and the second sums in (10) give the expected value of the next state if the next decision epoch is at the end of a processing event at one of the stations, or at one of the cells, respectively.

The mean cost of one transition is zero in equation (11) since no stations are starving: the probability that station i will be the first to become unblocked is $\lambda_i/\lambda(\mathbf{B})$. Equation (12) is used to fix the arbitrary additive component of the relative value vector ($v(\mathbf{n})$).

2.5. Computational algorithm

Under any policy it is possible to reach state $\mathbf{n} = \mathbf{0}$ if the cells happen to be temporarily sluggish and the workstations clear all their input buffers. Every Markovian transition probability matrix is, therefore, ergodic with $\mathbf{0}$ as a regeneration point, plus possibly a few transient states feeding it as well. This is sufficient to ensure that equations (10)–(12) possess a unique solution with real relative values (Schweitzer, 1971, 1984). Schweitzer's iterative solution algorithm can be adapted to this problem. The constant step size used in our finite-difference scheme to ensure fast, but stable, convergence is:

The initial guess used was $v(\mathbf{n}) = 0$ for all $\mathbf{n} \in \Omega$. The solution algorithm was halted when g could be estimated with $\pm 0.1\%$.

For computational simplicity the NS states $\mathbf{n} \in \Omega$ are mapped into the set of integers $\{1, 2, \dots, \text{NS}\}$

via the function $\psi: \Omega \rightarrow \{1, 2, \dots, \text{NS}\}$, where

$$\psi(\mathbf{n}) = 1 + n_R + \sum_{i=1}^{R-1} n_{R-i} \prod_{j=1}^i (B_{R-j+i} + 1). \quad (14)$$

The implementation of the solution algorithm is given at Appendix B. A computer program using this algorithmic approach is used to generate the optimal routing policy and the desired performance measures. The observed CPU time is roughly proportional to the square of the size of the state space since for each state one must compute the transition probabilities and the associated relative values for all (reachable) future states. For instance, solving a model having 64 states required 6.95 CPU sec. and 128K bytes of core memory on a CDC 855 computer with the NOS 2.3 operating system. In this instance convergence was observed after 76 iterations; increasing the state space of this model to 125 states required 94 iterations and 29.43 CPU sec. for solution.

3. Performance measures

3.1. Inputs

The solution of equations (10), (11) and (12) determines the optimal policy—denoted as

$$\mathbf{m}^*(\mathbf{n}) = (m_1^*(\mathbf{n}), m_2^*(\mathbf{n}), \dots, m_R^*(\mathbf{n})). \quad (15)$$

At this stage it becomes feasible to compute the performance measure of the system as a function of $\mathbf{m}^*(\mathbf{n})$. The detailed procedure for computing several performance measures is outlined here. A general scheme for evaluating additional measures is given by Dreyfus and Law (1977) and by Seidmann and Tenenbaum (1985). The computation of the following performance measures is based on the general structure of the functional equations (10)–(12). In this case the cost of one transition is substituted for the desired measure. It is then revealed through the solution of the augmented models as detailed below.

3.2. The throughput of the stations

Let r_t ($t = 1, 2, \dots, R$) denote the production rate of station t following policy $\mathbf{m}^*(\mathbf{n})$. Next,

define a new semi-Markovian process with the same state space producing a fractional ‘reward’ whenever the cells produce parts for station t : i.e., a process having a mean immediate reward in state $\mathbf{n} \neq \mathbf{B}$ of:

$$Q(\mathbf{n}) = \frac{m_i^*(\mathbf{n})\mu_i}{\lambda(\mathbf{n}) + \mu(m^*(\mathbf{n}))}, \quad \mathbf{n} \in \Omega; \mathbf{n} \neq \mathbf{B}. \tag{16}$$

This reward is the probability that the cells will complete processing a part for station t before the next decision epoch.

The NS + 1 value equations for computing the production rate of station t are:

$$v(\mathbf{n}) = Q(\mathbf{n}) - r_t/(\lambda(\mathbf{n}) + \mu(m^*(\mathbf{n}))) + \sum_{i \in A(\mathbf{n})} \frac{\lambda_i}{\lambda(\mathbf{n}) + \mu(m^*(\mathbf{n}))} v(\mathbf{n} - \mathbf{e}^i) + \sum_{i=1}^R \frac{m_i^*(\mathbf{n})\mu_i}{\lambda(\mathbf{n}) + \mu(m^*(\mathbf{n}))} v(\mathbf{n} + \mathbf{e}^i) \tag{17}$$

$$\mathbf{n} \in \Omega; \mathbf{n} \neq \mathbf{B}, \text{ (where } m^*(\mathbf{n}) \in \theta(\mathbf{n}), \text{)}$$

$$v(\mathbf{B}) = -\frac{r_t}{\lambda(\mathbf{B})} + \sum_{i=1}^R \frac{\lambda_i}{\lambda(\mathbf{B})} v(\mathbf{B} - \mathbf{e}^i), \tag{18}$$

$$v(\mathbf{0}) = 0. \tag{19}$$

The iterative algorithm of Schweitzer (1971) is applied to solve these equations. The long run expected reward per unit time (the ‘gain rate’) is r_t —which is also the effective throughput rate of station t ($t = 1, 2, \dots, R$).

3.3. The utilization of the stations

Given r_t , the utilization of station t is given by

$$U_t = r_t/\lambda_t, \quad t = 1, 2, \dots, R. \tag{20}$$

3.4. The utilization of the cells

The state transitions in this system commence, and terminate, at the end of processing events—either at the cells or at one of the stations. The utilization of the cells, therefore, cannot be computed directly from the stations throughputs.

Let CU denote the expected utilization of the cells. To compute CU define a new semi-Markovian process similar to (17)–(19) with a one

transition reward:

$$Q(\mathbf{n}) = \begin{cases} \frac{1}{\lambda(\mathbf{n}) + \mu(m^*(\mathbf{n}))} \sum_{i=1}^R \frac{m_i^*(\mathbf{n})}{S} & \text{if } \mathbf{n} \neq \mathbf{B}, \mathbf{n} \in \Omega, \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

The immediate reward $Q(\mathbf{n})$ is equal to the mean holding time at state \mathbf{n} multiplied by the ratio of active cells at that state.

The gain rate of this semi-Markovian process is CU (rather than r_t in (17)–(19)). Solving these equations reveals the value of CU. Clearly, $(1 - \text{CU})$ is the fraction of time that all the cells are blocked.

3.5. The effective production rate of the cells

The effective throughput of the cells is given directly by

$$\text{CEPR} = \sum_{t=1}^R r_t. \tag{22}$$

4. Numerical examples

In the first example, a system having three stations ($R = 3$) is considered. All stations have identical production rates of 6 units per hour ($= \lambda_1 = \lambda_2 = \lambda_3$) and five buffer spaces ($= B_1 = B_2 = B_3$). The shortage penalties are \$120, \$370 and \$210 per hour idle ($= C_1, C_2, C_3$). These stations are fed by S (≥ 1) cells. Each cell can produce $16/S$ units per hour ($= \mu_1 = \mu_2 = \mu_3$), so aggregate productive capacity at the cells is 16 parts per hour.

Examining the structure of the optimal policy, the number of the parallel cells has been incremented from one to six. From equation (2) the state space for this problem has $(5 + 1)^3 = 216$ states. Table 1 illustrates several system states and the optimal policies $m^*(\mathbf{n})$ for $S = 1, 3$ and 6. For instance, when each station has two parts [$\mathbf{n} = (2, 2, 2)$, $\psi(\mathbf{n}) = 87$] and $S = 6$ then three cells are assigned to produce part type 2 and additional three cells are assigned to part type 3 [$m^*(\mathbf{n}) = (0, 3, 3)$].

Denote by $m_i^*(\mathbf{n})^S$ the value of the i -th element of the optimal decision vector $m^*(\mathbf{n})$ computed for a system having S parallel cells. Observ-

Table 1
Optimal decision vectors for a varying number of parallel cells
($B = 5, 5, 5$)

$\psi(n)$	n	$m^*(n)$		
		$S=1$	$S=3$	$S=6$
1	0 0 0	0 1 0	0 3 0	0 5 1
2	0 0 1	0 1 0	0 3 0	0 5 1
3	0 0 2	0 1 0	0 3 0	0 5 1
...				
18	0 2 5	0 1 0	0 3 0	3 3 0
19	0 3 0	0 0 1	0 0 3	0 1 5
20	0 3 1	0 0 1	0 0 3	0 2 4
21	0 3 2	0 1 0	0 2 1	1 2 3
22	0 3 3	0 1 0	1 2 0	4 2 0
23	0 3 4	1 0 0	3 0 0	5 1 0
24	0 3 5	1 0 0	3 0 0	5 1 0
25	0 4 0	0 0 1	0 0 3	0 1 5
26	0 4 1	0 0 1	0 0 3	1 1 4
27	0 4 2	1 0 0	3 0 0	5 0 1
28	0 4 3	1 0 0	3 0 0	5 1 0
29	0 4 4	1 0 0	3 0 0	5 1 0
...				
85	2 2 0	0 0 1	0 0 3	0 1 5
86	2 2 1	0 1 0	0 3 0	0 3 3
87	2 2 2	0 1 0	0 3 0	0 3 3
88	2 2 3	0 1 0	0 3 0	1 3 2
...				
214	5 5 3	0 0 1	0 0 2	0 0 2
215	5 5 4	0 0 1	0 0 1	0 0 1
216	5 5 5	0 0 0	0 0 0	0 0 0

ing the structure of the optimal decision vectors in Table 1 it is clear that the values of $m_i^*(n)^S$ are a monotone function of S :

$$m_i^*(n)^S \leq m_i^*(n)^{S+1} \quad \text{for } S = 1, 2, \dots$$

In other words, increasing the number of parallel cells results in incremental (or additive) allocation of cells to part types already processed by the cells. This is certainly what one would expect. As an example of this 'incremental allocation' property consider state $\psi(n) = 25$. The values of $m_3^*(n)^S$ in this state are $(m_3^*(n)^1 = 1) \leq (m_3^*(n)^3 = 3) \leq (m_3^*(n)^6 = 5)$.

Furthermore, in most cases it seems that at each decision epoch the optimal policy attempts to assign as much of the cells capacity as possible to a certain 'critical' part type. For instance, denoting the 'critical' type by i then the number of cells assigned to it is equal to $\min(S, B_i - n_i)$. The remaining cells, which number $\max(0, S - (B_i - n_i))$, are optimally distributed among the other

part types. This 'critical assignment' property is clearly evidenced for all system states in Table 1: Whenever $m_i^*(n)^1 = 1$ then i implies the critical type at that state. This leads to the following relationship observed in Table 1 for $S = 3$ or 6:

$$m_i^*(n)^S = \min(S, B_i - n_i) \quad \text{if } m_i^*(n)^1 = 1; \\ n \in \Omega; 1 \leq i \leq R.$$

Note, for instance, that $m_3^*(2, 2, 0)^1 = 1$; since type 3 is critical in this state it results in $m_3^*(2, 2, 0)^3 = \min(3, 5 - 0) = 3$ and $m_3^*(2, 2, 0)^6 = \min(6, 5 - 0) = 5$.

It is instructive to note that in certain cases the optimal policy will be to produce more parts for a station already having parts in its input buffer rather than to produce for an idle station. This feature is clearly depicted at state $\psi(n) = 18$: Having $S = 1$ or $S = 3$ it was preferred to reduce the chances of starvation at station 2 as compared to reducing the current starvation of station 1. This was observed in a previous paper using a non-regenerative control scheme (Seidman and Schweitzer, 1984).

Several performance measures are displayed in Table 2. The measures presented there are: work stations production rates (r_j), work stations utilizations (U_j), the utilization of the cells (CU), the cells effective production rate (CEPR), and the expected starvation cost per time unit (g).

The results presented in Table 2 refer to three systems operating with one, three and six parallel cells. The aggregate productive capacities of the cells in all three systems were identical. A slight

Table 2
Summary of performance measures for a varying number of parallel cells.

Performance measure	j	Number of parallel cells		
		$S=1$	$S=3$	$S=6$
r_j	1	3.68	3.68	3.80
	2	5.93	5.89	5.66
	3	5.65	5.59	5.43
U_j	1	61.45%	61.45%	63.35%
	2	98.85%	98.28%	94.46%
	3	94.13%	93.20%	90.63%
CU		99.41%	97.58%	95.23%
CEPR		15.26	15.16	14.89
g		62.83	66.98	84.13

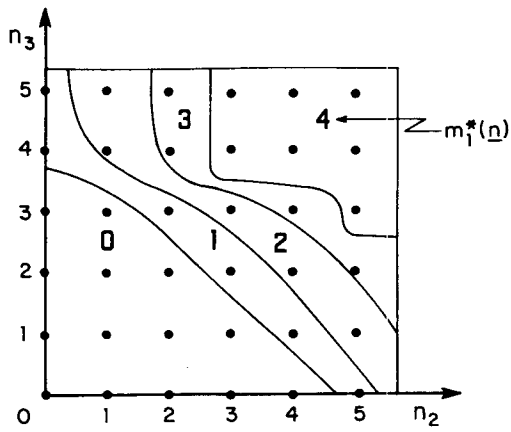


Figure 3. The first component $[m_1^*(n)]$ of the optimal decision vector $m^*(n)$ for the case of $n_1 = 1$ and $S = 6$

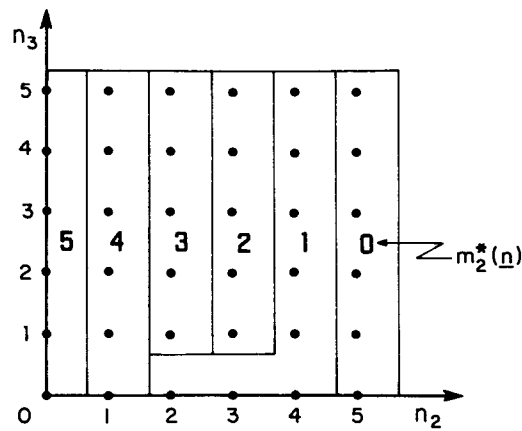


Figure 4. The second component $[m_2^*(n)]$ of the optimal decision vector $m^*(n)$ for the case of $n_1 = 1$ and $S = 6$

reduction in performance is observed as the number of parallel cells increases. This happens because the current number of active cells at each decision epoch cannot exceed the total number of available buffer spaces. This means that as S increases, there's a decline in the effective productivity of the cells, particularly at the high-buffer end of the state space.

Figures 3, 4 and 5 depict the components of the optimal decision vector $m^*(n)$ for the case of $n_1 = 1$ with $6 (= S)$ parallel cells. Consider, for example, state $n\{1, 4, 2\}$ in these Figures. The optimal decisions for that state are: $m_1^*(n) = 2$ (Figure 3), $m_2^*(n) = 1$ (Figure 4) and $m_3^*(n) = 3$ (Figure 5). The optimal decision regions in each state space rectangle are simply connected sets. In

general, it seems that these sets are not convex. The optimal policies are piece-wise constant functions of intermediate buffer levels and the boundary curves between adjacent sets have no inflection points. Clearly, the values of $m_i^*(n)$ decrease monotonically when viewed as a function of n_i (e.g., $m_2^*(n)$ vs. n_2 in Figure 4 or $m_3^*(n)$ vs. n_3 in Figure 5).

The next example, introduces variations in the productive capacity of the S parallel cells in order to investigate their effect on the structure and performance of the optimal control strategy. Consider the system described above, but assume only three buffer spaces ($B_1 = B_2 = B_3 = 3$) with four parallel cells ($S = 4$). The productive capacity of each cell is μ_i parts per hour and the processing times are identical for the three part types (i.e., $\mu_1 = \mu_2 = \mu_3$). Three cases are evaluated: in the first case $\mu_i = 9$, in the second $\mu_i = 4.5$ and $\mu_i = 2.25$ in the third ($i = 1, 2, 3$). Obviously, the cells are 'over-powered' (relative to the capacity of the work stations) in the first case, and they are 'balanced' and 'under-powered' in the second and third, respectively. Table 3 illustrates the optimal decision vectors $m^*(n)$ for these three cases.

In the 'underpowered' case, where starvation is imminent, one can expect that the optimal decision would tend to be myopic: decisions would be mainly responses to the *greatest* immediate threat. This is clearly evident in Table 3: In the 'underpowered' case the highest priority is given to the production of additional parts for the second work station (i.e., the typical pattern in this 'underpowered' case is $m_2 = \min[S, B_2 - n_2]$). It hap-

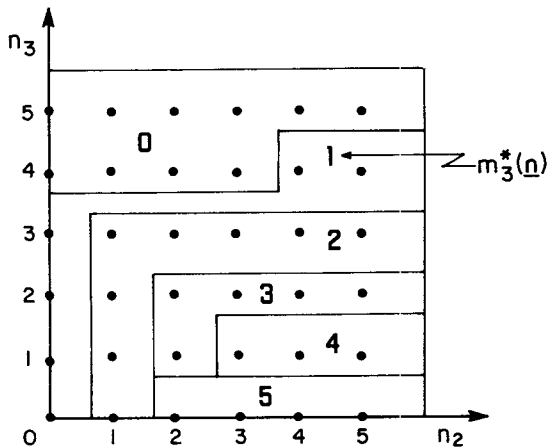


Figure 5. The third component $[m_3^*(n)]$ of the optimal decision vector $m^*(n)$ for the case of $n_1 = 1$ and $S = 6$

Table 3
Optimal decision vectors for varying productive capacities of the parallel cells ($B = (3, 3, 3)$, $S = 4$)

$\psi(n)$	n	$m^*(n)$		
		$\mu_i = 9$ 'over-powered'	$\mu_i = 4.5$ 'balanced'	$\mu_i = 2.25$ 'under-powered'
1	0 0 0	0 3 1	0 3 1	0 3 1
2	0 0 1	1 3 0	0 3 1	0 3 1
3	0 0 2	1 3 0	1 3 0	0 3 1
4	0 0 3	1 3 0	1 3 0	1 3 0
5	0 1 0	1 0 3	0 2 2	0 2 2
6	0 1 1	3 1 0	0 2 2	0 2 2
7	0 1 2	3 1 0	2 2 0	1 2 1
8	0 1 3	3 1 0	2 2 0	2 2 0
9	0 2 0	1 0 3	0 1 3	0 1 3
10	0 2 1	3 0 1	1 1 2	1 1 2
11	0 2 2	3 1 0	3 1 0	2 1 1
...				
21	1 1 0	0 1 3	0 2 2	0 2 2
22	1 1 1	0 2 2	0 2 2	0 2 2
23	1 1 2	2 2 0	1 2 1	1 2 1
...				
37	2 1 0	0 1 3	0 2 2	0 2 2
38	2 1 1	0 2 2	0 2 2	0 2 2
...				
62	3 3 1	0 0 2	0 0 2	0 0 2
63	3 3 2	0 0 1	0 0 1	0 0 1
64	3 3 3	0 0 0	0 0 0	0 0 0

pens probably because the second work station is associated with the highest starvation penalty ($= C_2$). In the 'overpowered' case, on the other hand, the optimal policy tries to respond to *most* immediate threats by allocating at least one cell for each starving work station; while doing so, a smaller number of cells is allocated to feeding a single 'critical' station (i.e., the typical pattern in this 'overpowered' case is to set $m_i \geq 1$ whenever $n_i = 0$). This disparity between the response patterns of the optimal policy is for the most part observed at the low-buffer region of the state space (e.g., $\psi(n) = 5 - 11$).

The decisions in the 'balanced' case constitute an 'averaged' response mode between the two extreme cases discussed above. For example, in state $\psi(n) = 3$ the 'overpowered' and the 'balanced' decisions are identical, but in state $\psi(n) = 6$ only the 'underpowered' or the 'balanced' decisions coincide. Performance measures for these three cases are presented in Table 4. Increasing the capacity of the cells allows for higher work

Table 4
Summary of performance measures for varying production capabilities of the parallel cells

Performance measures	j	Production rate of the cells		
		$\mu_i = 9$ 'over-powered'	$\mu_i = 4.5$ 'balanced'	$\mu_i = 2.25$ 'under-powered'
r_j	1	5.81	4.28	1.19
	2	5.84	5.34	4.17
	3	5.83	5.13	3.46
U_j	1	96.92%	71.48%	19.90%
	2	97.42%	89.07%	69.58%
	3	97.26%	85.58%	57.79%
CU		48.60%	82.04%	98.18%
CEPR		17.48	14.75	8.82
g		19.00	104.95	297.32

stations utilizations (U_j) that results in lower penalty (g). This increase also leads to a higher blocking probability, which results in having progressive reduction in the utilization of the cells. Furthermore, it can be observed that the differences between the work stations throughputs (r_j) become less evident as the production rate of the cells increases. In summary, it appears that the examples discussed in this section emphasize the inherent interaction between the structure of the optimal policy, the performance of the system and the desired allocation of productive capabilities among the manufacturing resources.

5. Concluding comments

Full production control schemes maintain conformity to dynamic changes by having all processes produce only the necessary parts at the necessary time and have on hand only a limited amount necessary to hold the production stages together. Each preceding process then produces parts as they are withdrawn by the following process. This paper formulates two optimal regenerative pull, control policies for an automated industrial facility. Following the computerized implementation of the optimization model, several examples are presented. These examples illustrate some important properties with respect to the inherent interaction between the structure of the optimal policy, the performance of the system and

the desired allocation of productive capabilities among the manufacturing resources. They indicate that the optimal decision regions are simply closed-connected sets and that, in general, these sets are not convex. These examples also demonstrate that the optimal policy is monotone in the number of cells. Besides, it is shown that the differences between the work stations throughputs become less pronounced with the increase in the capacity of the cells. Comprehending the structural properties of the optimal policy is vital to the design, analysis and control of complex pull production control systems.

A potential extension of these models may take into consideration other factors like dynamic changes in the number of Kanbans and their interaction with such issues as reliability or machine failures and repairs. Generalization of the current functional equations might prove useful in accomodating in these models a broader set of processing time probability distributions.

Appendix A. Maximizing the expected weighted throughput with parallel cells

Consider the case where a weight (value) of W_i ($W_i \geq 0, i = 1, 2, \dots, R$) dollars is associated with each item produced at station i . The expected benefit incurred when the system is in state n and a decision $m \in \theta(n)$ is made, is equal to

$$\sum_{i=1}^R W_i \frac{m_i \mu_i}{\lambda(n) + \mu(m)}. \tag{A.1}$$

This leads to the following dynamic programming functional equations for *maximizing* the expected weighted throughput of the system:

$$v(n) = \max_{m \in \theta(n)} \left\{ \sum_{i=1}^R W_i \frac{m_i \mu_i}{\lambda(n) + \mu(m)} + \sum_{i \in A(n)} \frac{\lambda_i}{\lambda(n) + \mu(m)} v(n - e^i) + \sum_{i=1}^R \frac{m_i \mu_i}{\lambda(n) + \mu(m)} v(n + e^i) \right\}, \tag{A.2}$$

(unblocked case) $n \in \Omega, n \neq B, \tag{A.3}$

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

(blocked case) $n = B, \tag{A.4}$
 $v(0) = 0.$

In these equations g is the expected long run weighted value produced per unit time following the optimal policy. The solution of (A.2)–(A.4) follows the general scheme used earlier for solving (10)–(12).

Appendix B. An iterative solution algorithm for undiscounted semi-Markovian decision process with scalar gain rate

This Appendix outlines the iterative solution procedure as proposed by Schweitzer (1971). It presents a data-transformation which converts a semi-Markovian decision model into a discrete-time Markovian decision model so that each stationary policy has the same mean cost in both models.

For brevity, define the following compact variables over the finite state space Ω having NS states:

Unknowns

- V_i = relative value of state i .
- g = the gain rate or the maximal possible expected reward per unit time achievable by any policy in Ω .

Decisions

- $\theta(i)$ = Finite non-empty set of actions in state i .
- m = decision index.

Problem descriptors

- T_i^m = mean holding time at state i following decision $m, m \in \theta(i)$.
- q_i^m = mean one transition reward at state i following decision m .
- P_{ij}^m = the transition probability from state i to j if decision m is made at state i .

It is assumed that

$$P_{ij}^m \leq 1, \quad \sum_{j=1}^{NS} P_{ij}^m = 1, \quad i, j \in \Omega, \tag{B.1}$$

$$0 < T_i^m < \infty, \quad i \in \Omega, m \in \theta(i).$$

The functional equations are

$$V_i = \max_{m \in \theta(i)} \left\{ g_i^m - gT_i^m + \sum_{j=1}^{NS} P_{ij}^m V_j \right\}, \quad 1 \leq i \leq N. \quad (\text{B.2})$$

Having $NS + 1$ unknown variables and (g and V_i) NS equations we fix arbitrarily $V_1 = 0$. This leads to the following alternate representation:

$$g = \max_{m \in \theta(1)} \left\{ \left(q_1^m + \sum_{j=2}^{NS} P_{1j}^m V_j \right) / T_1^m \right\}, \quad (\text{B.3})$$

$$V_i = \max_{m \in \theta(i)} \left\{ q_i^m - gT_i^m + \sum_{j=2}^{NS} P_{ij}^m V_j \right\}, \quad 2 \leq i \leq NS. \quad (\text{B.4})$$

Denote by g^p and by V_i^p the values of g and V_i at the p -th iteration. Assume also that the initial values are $g^0 = 0$ and $V_i^0 = 0$.

The stable iterative algorithm used for solving (B.3) and (B.4) is:

$$g^p = \max_{m \in \theta(1)} \left\{ \left(q_1^m + \sum_{j=2}^{NS} P_{1j}^m V_j^p \right) / T_1^m \right\} \quad (\text{B.5})$$

and

$$V_i^{p+1} = V_i^p + \tau \left(\max_{m \in \theta(i)} \left\{ \left(q_i^m - g^p T_i^m + \sum_{j=2}^{NS} P_{ij}^m V_j^p - V_i^p \right) / T_i^m \right\} \right), \quad (\text{B.6})$$

$2 \leq i \leq NS$.

The step size τ in (B.6) must satisfy

$$0 < \tau < \tau_{\max} \equiv T_i^m / (1 - P_{ii}^m), \quad i \in \Omega, m \in \theta(i). \quad (\text{B.7})$$

In general (B.7) is satisfied when

$$\tau = \min \{ T_i^m \}, \quad i \in \Omega, m \in \theta(i). \quad (\text{B.8})$$

Recently Tijms and Eikeboom (1980) presented a new approach to accelerate the computational requirements of such models. It is based on the addition of fictitious decision epochs so that sparse

transition matrices are generated. For a more comprehensive and up-to-date treatment of this topic the reader is referred to Tijms (1986).

Acknowledgements

The author thanks Professor P.J. Schweitzer from the W.E. Simon Graduate School of Business Administration at the University of Rochester and two anonymous referees for useful reviews and comments.

References

- Bitran, G.R., and Chang, L. (1987), "A mathematical programming approach to a deterministic Kanban system", *Management Science* 33/4, 427-441.
- Black, J.T. (1983), "An overview of cellular manufacturing systems", *Industrial Engineering* 15/11, 36-47.
- Bullen, A.G.R., and Humman, N.P. (1987), "An optimization model for a fully actuated traffic signal", presented at TIMS/ORSA national meeting, May 1987, New Orleans.
- Buzacott, J.A. and Hanifin, L.E. (1978), "Models of transfer lines with inventory banks—A review and comparison", *AIIE Transactions* 10/2, 197-209.
- Cinlar, E. (1967), "Decomposition of a semi-Markovian process under a state dependent rule", *SIAM J. Appl. Math* 15/2, 252-263.
- Conway, R.W., Maxwell, W.L., and Miller, L.W. (1967), *Theory of Scheduling*, Addison-Wesley, Reading, MA.
- Costa A., and Garetti, M. (1985), "Design of a control system for flexible a manufacturing cell", *Journal of Manufacturing Systems* 4/1, 65-84.
- Dreyfus, S.E. and Law, A.M. (1977) *The Art and Theory of Dynamic Programming*, Academic Press, New York.
- Dubios, D. (1983), "A mathematical model of a flexible manufacturing system with limited in-process inventory", *European Journal of Operations Research* 14/1, 66-78.
- Elsayed, E.A. and Hwang, C.C. (1986), "Analysis of two-stage manufacturing systems with buffer storage and redundant machines", *International Journal of Production Research* 24/1 187-201.
- 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* 13/1, 2-11.
- 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.
- Greene, T.J., and Sadowski, R.P. (1986), "A mixed integer program for loading and scheduling multiple flexible manufacturing cells", *European Journal of Operational Research* 24/3, 379-386.
- Groover, M.P., and Zimmers, E.W. (1984), *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ.

- 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.
- Herer, Y. (1986), "Buffer placement in sequential production lines: further studies", working paper, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.
- Holmqvist, V. (1984), "Programmable assembly station with new ASEA robot", In: Martensson, N. (Ed.), *Proc. 14th ISIR, Gothenburg, Sweden, October, 2-4, 1984*, North-Holland, Amsterdam and IFS Bedford, UK, 617-622.
- Howard, R.A. (1971), *Dynamic Probabilistic Systems, Vol. II: Semi-Markov and Decision Processes*, Wiley, New York.
- Jablonowski, J. (1987), "The advantages of phasing-in an FMS in measured steps", *American Machinist* 131/4, 69-72.
- Jewell, W.J. (1963), "Markov-renewal programming: I and II", *Operations Research* 11/6, 938-948, 949-971.
- Kimura, O., and Terada, H. (1981), "Design and analysis of pull system, a method of multistage production control", *International Journal of Production Research* 19/3, 241-253.
- Kleinrock, L. (1976), *Queuing Systems, Volume 2: Computer Applications*, Wiley, New York.
- Koren, Y. (1985), *Robotics for Engineers*, McGraw-Hill, New York.
- Kusiak, A. (1986), "Application of operational research models and techniques in flexible manufacturing systems", *European Journal of Operational Research* 24, 336-345.
- Monden, Y. (1981), "Adaptable Kanban system helps toyota maintain just-in-time production", *Industrial Engineering* 13/5, 29-46.
- Nof, S.Y. (1985), *The Handbook of Industrial Robotics*, Wiley, New York.
- Okamura, K., and Yamashina, H. (1979), "Justification for installing buffer stocks in unbalanced two stage automatic transfer lines", *AIIE Transactions* 11/4, 308-312.
- Schonberger, R.J. (1983), "Plant layout becomes product-oriented with cellular just-in-time production concepts", *Industrial Engineering* 15/11, 66-71.
- Sarker, B.R. (1984), "Some comparative and design aspects of series production systems", *IIE Transactions* 16/3, 229-240.
- Schweitzer, P.J. (1971), "Iterative solution of the functional equations of undiscounted Markov renewal programming," *Journal of Mathematical Analysis and Applications* 34, 495-501.
- Schweitzer, P.J. (1984), "On the existence of relative values for undiscounted Markovian decision processes with a scaler gain rate", *Journal of Mathematical Analysis and Applications* 104/1, 67-78.
- Seidmann, A., and Schweitzer, P.J. (1984), "Part selection policy for a flexible manufacturing cell feeding several production lines", *IIE Transactions* 16/4, 355-362.
- Seidmann, A., and Tenenbaum, A. (1985), "Optimal dynamic routing in flexible manufacturing systems", Working paper, Dept. of Industrial Engineering, Tel-Aviv University 1985.
- Seidmann, A., Arbel, A., and Shapira, R. (1984), "A two-phase analytic approach to robotic system design", *Robotics & Computer Integrated Manufacturing* 1/2, 181-190.
- So, K., and Pinault, C. (1987), "Allocating buffer storage in a pull system", working paper, AT & T Engineering Research Center, Princeton, NJ.
- Stidham, S., and Altiok, R. (1984), "Production lines with unreliable machines and finite buffers", TMS/ORSA Conference, 1984, San Francisco, CA.
- Tijms, H.C. (1986), *Stochastic Modelling and Analysis: A Computational Approach*, Wiley, New York.
- Tijms, H.C., and Eikeboom, A.M. (1986), "A simple technique in Markovian control with applications to resource allocation in communication networks", *Operations Research Letters* 5/1, 25-32.
- Yao, D.D., and Buzacott, J.A. (1986), "Models of flexible manufacturing systems with local buffers", *International Journal of Production Research* 24/1, 107-117.
- Yao, D.D., and Buzacott, J.A. (1985), "Modelling the performance of flexible manufacturing systems", *International Journal of Production Research* 23/5, 945-959.
- Yao, D.D., and Buzacott, J.A. (1986), "Modelling a class of state dependent routing in flexible manufacturing systems", *Annals of Operations Research*, 3, 153-168.
- Yao, D.D., and Shanthikumar, J.G. (1987), "The optimal input rate to a system of manufacturing cells", *INFOR* 25/1 57-65.
- Watanbe, T., and Sakamoto, M. (1985), "On-line scheduling for adaptive control machine tools in FMS", *Robotics and Computer-Integrated Manufacturing* 2/1, 65-73.
- Wilhelm, W.E., Nof, S.Y., and Seidmann, A. (1986), "Analysis of recirculation in robotic systems using feedback models", *Material Flow* 3/1 69-82.