INT. J. SYSTEMS SCI., 1989, VOL. 20, NO. 4, 699–714

Queueing analysis of manufacturing systems with set ups

PETER B. LUH† and DEBRA J. HOITOMT†

A manufacturing system with one server (machine), two classes of jobs, finite buffer sizes, and non-negligible set-up times is analysed. A cycling service discipline with a 'wait and see' set-up policy is invoked by the server. The state of the machine, whether in set up or in processing, is explicitly considered in the model. Utilization, mean queue length, and cycle time are derived using markovian analysis, first under a fixed lot size environment and then with random lot sizes. With lot sizes fixed, increasing arrival rates, set-up times, and service times generally increase the utilization, cycle time, and queue lengths. When lot sizes are random, there are negligible variations in the cycle time result, with somewhat smaller queue lengths compared to the fixed lot size case. An approximate mean value analysis is also developed and results are compared to those of the markovian analysis. Numerical studies show the robustness of the mean value analysis for most of the system parameters tested.

Nomenclature

L_i lot size in parts per lot for class i ($i = 1, 2$)
a_i arrival rate in parts per hour for class i
b_i service rate in parts per hour for class i
A_i arrival rate in lots per hour for class i
B_i service rate in lots per hour for class i
N_i buffer size in lots for class i
H shift factor (one shift has 8 hours and one working day with shift factor H has $8 \cdot H$ hours)
s_{ij} interclass set-up rate from class i to class j ($i \neq j$) in set ups per hour
s_{ii} intraclass set-up rate in set ups per hour for class i
s_{ij}^r reduced interclass/intraclass set-up rate $= (1/s_{ij} - 1/s_{ii})^{-1}$
U machine utilization

1. Introduction

A manufacturing system with one server (machine), two classes of jobs, finite buffer sizes and non-negligible set-up times is analysed. Set ups become necessary when a machine is serving more than one job class, while each job class requires a particular machine configuration. For a machine with long set-up times, it may become economical to serve many members of a job class when the machine is thus set up. The classes are then served in a fixed order. Service disciplines with these characteristics are generally referred to as cycling or alternating priority in the case of two queues. Two types of cycling have been discussed previously. Non-exhaustive cycling involves serving a fixed number of jobs from the queue of each class (Cooper 1970, Kuehn 1979, Servi 1985). Exhaustive cycling takes place when all the members

Received 27 August 1987.
† Department of Electrical and Systems Engineering, University of Connecticut, Storrs, Connecticut 06269-3157, U.S.A.

of a particular class in the queue are processed before setting up (or changing over) to process another class (Cooper 1970, Eisenberg 1971, Doshe 1985). Generally, results have been obtained by assuming a markovian or semi-markovian process and using a combination of probability generating functions, Laplace–Stieltjes transforms, and renewal theory.

Set-up policies have been categorized by Mevert (1968). The system considered in this paper employs an exhaustive cycling service discipline with a 'wait and see' set-up policy. That is, when the system is idle, it is left in the orientation determined by the last job. If the next arriving job is of the same class as the previous one, it enters service directly; otherwise, a set up is required. This system is similar to that considered by Eisenberg (1971). The models vary in the definition of state and in the methodology used to derive performance characteristics. The solution methodology presented herein was developed independently of the methods used by Eisenberg (1971). Our methodology using the markovian analysis gives a better picture about the set-up and cycling processes, and the results offer additional insights into the effects of random lot sizes. The mean value analysis developed here also provides an effective technique for the approximate analysis of a system.

This paper is based upon an existing production system. There are two major classes of jobs, and—of less concern to us—each class in turn consists of many part types. The parameters utilized, such as lot sizes, service times, set-up times, and arrival rates, are based upon data from this system. For lot sizing, the simplifying assumption that lot sizes are deterministic is imposed first. A markovian analysis similar to those described by Kleinrock (1975) is developed, equilibrium equations are derived, and a Gauss–Seidel numerical method is utilized to solve the equations for state probabilities. Markovian analysis is then performed assuming random variation of the lot sizes in the arrival process. We find that the length of a cycle remains approximately equal to the fixed lot size case.

Markovian analysis provides exact results. However, it suffers from the 'curse of dimensionality' in that the size and computational requirements of the problem increase drastically with buffer sizes (and other system parameters such as number of machines, classes of jobs, etc. for a more general problem). Mean value analyses (MVAs) have been found to adequately represent many types of product-form systems with much less computational requirements (Sauer and Chandy 1981, Lavenberg 1983, Lazowska *et al.* 1984). The MVA developed here is an approximation method, with assumptions similar to those given by Sykes (1970), and it follows the more heuristic concepts developed by Lazowska *et al.* (1984) for product-form networks.

The paper is organized as follows. Section 2 defines key variables and presents working assumptions. Under the fixed lot size assumption, § 3 sets out the markovian analysis, with associated state transitions and equilibrium equations. The results presented in § 4 include sensitivity studies with respect to set-up times and arrival rates. In § 5, we relax the fixed lot size assumption and allow a number of different sized lots to arrive in a probabilistic fashion. The mean value analysis is presented in § 6.

2. Modelling the production system

2.1. *Modelling the cycling service discipline*

We wish to obtain a mathematical description that accounts for the cycling service discipline and the 'wait and see' set ups between classes—the 'interclass set ups'. With two classes of jobs, the machine alternates periods of setting up and processing. A cycle spans the four possible machine states, which are given as follows:

(1) processing or ready to process Class 1 jobs;

(2) processing or ready to process Class 2 jobs;

(3) setting up for Class 2; and

(4) setting up for Class 1.

In addition, we define

$$m = \text{machine state} \quad (m = 1, 2, 3, 4 \text{ as defined above})$$

$$x = \text{number of Class 1 lots} \quad (x \leqslant N_1)$$

$$y = \text{number of Class 2 lots} \quad (y \leqslant N_2)$$

where N_1 is the buffer size for Class 1 jobs and N_2 is the buffer size for Class 2 jobs. The cycling service discipline is depicted in Fig. 1. The state of the system is completely described by the ordered triple (x, y, m). We note that there are a maximum of $4N_1N_2$ possible states. Some of them, however, are undefined—$(0, y, 1)$ cannot exist with $y > 0$, since we immediately proceed to setting up for Class 2 ($m = 3$) when $x = 0$ and $y > 0$.

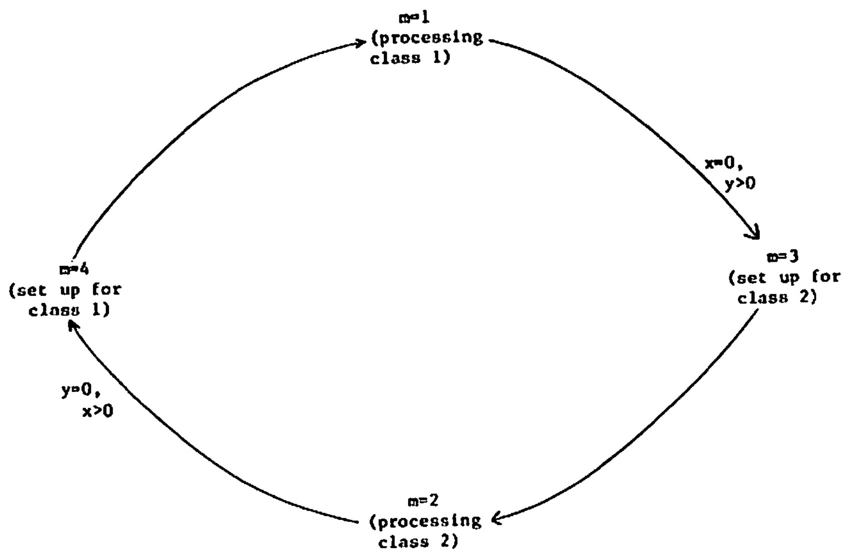

Figure 1. Cycling.

Consider the sample path shown in Figure 2. The machine finishes setting up for Class 1 at time t_0, and from t_0 to t_1, it is processing Class 1 jobs ($m = 1$). Thus, we see that the number of Class 1 jobs is generally decreasing, although arrivals continue. From t_1 to t_2, the machine is setting up to process Class 2 ($m = 3$). From t_2 to t_3, the machine is busy processing Class 2 jobs ($m = 2$), and from t_3 to t_4, the machine is setting up for Class 1 ($m = 4$). Note that from t_1 to t_4, Class 1 jobs are non-decreasing. Also, at times t_1 and t_5, we have no Class 1 jobs in the system, as we are approaching a set up to process Class 2 jobs. This interval t_1–t_5 is referred to as a 'cycle'. A cycle can be defined as a time interval that spans all the machine states ($m = 1, 2, 3, 4$).

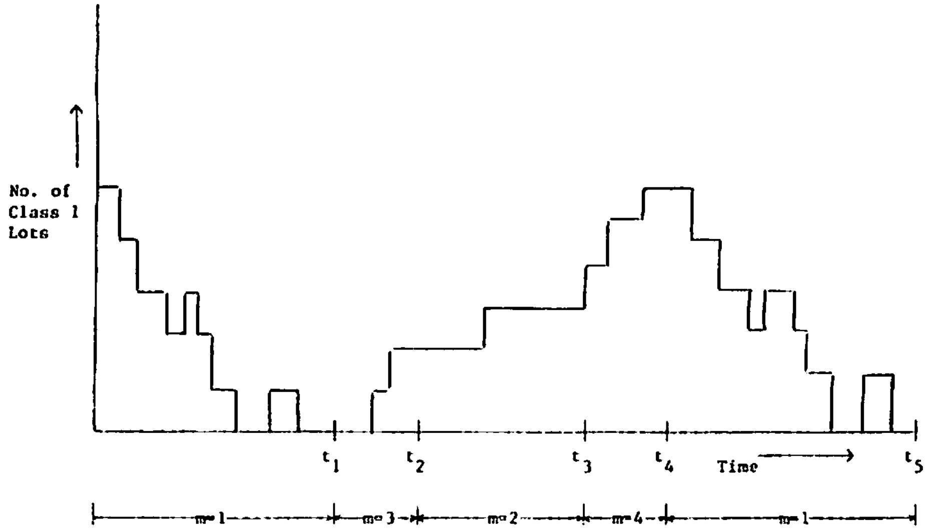

Figure 2. Sample path of Class 1 lots in system over time.

2.2. *Assumptions and notation*

As noted before, there are two job classes, each class in turn consisting of many part types. An interclass set up is required between processing lots of different classes. Another kind of set up, called the intraclass set up, is required between processing two Class 1 lots (or between processing two Class 2 lots) if they are of different part types. Since adjacent lots are generally of different part types, intraclass set up is required for most lots. To simplify the analysis, intraclass set-up time is lumped with processing time, and thus the service times are augmented. In the meantime, we have the reduced interclass set-up times, as a portion of an interclass set-up time has already been included in the augmented service time.

The lot arrival processes for the two job classes are assumed to be Poisson distributed. The reduced interclass set-up times are assumed to be exponentially distributed. Augmented service time, which is the sum of an intraclass set-up time and a lot processing time, is also assumed to be exponentially distributed. Feasible states are mutually exclusive and the evolution of states is a Markov process. These characteristics allow us to model the system as a Markov chain. The variables used are as defined in the Nomenclature section.

We measure the performance of the system in terms of c and z_i, where c is the average cycle time in hours (maximum turnaround time, as depicted in Fig. 1), and z_i is the output in lots per cycle for class i ($i = 1, 2$).

The lot arrival and lot service rates are related to part arrival and part service rates. First, the 'per lot' arrival rate A_i is given by

$$A_i = \frac{a_i}{L_i} \text{ lots/hours} \tag{2.1}$$

The 'per lot' service rate is given by the shift factor divided by the augmented service time:

$$B_i = \frac{H}{\left[\dfrac{L_i}{b_i} + \dfrac{1}{s_{ii}}\right]} \text{ lots/hour} \tag{2.2}$$

Note that the shift factor H is handled by modifying the service rates. The quantity A_i/B_i is the machine utilization (or percentage of time for servicing, including intraclass set ups) for class i. The total utilization is given by

$$U = \frac{A_1}{B_1} + \frac{A_2}{B_2} \tag{2.3}$$

which should satisfy

$$U < 1 \tag{2.4}$$

We refer to this definition as the standard utilization.

The proportion of a cycle dedicated to real processing—i.e. excluding all types of set ups—is called the real processing utilization U_{rp}, and is given by

$$U_{rp} = \frac{1}{H} \frac{\left[z_1 L_1 \frac{1}{b_1} + z_2 L_2 \frac{1}{b_2} \right]}{c} = \frac{1}{H} \left[\frac{a_1}{b_1} + \frac{a_2}{b_2} \right] \tag{2.5}$$

as $z_i L_i / c = a_i$, assuming $U < 1$.

Finally, the total proportion of busy time including both types of set-up times, called the processing-plus-set up utilization U_{ps}, is given by

$$U_{ps} = U + P(m=3) + P(m=4) = 1 - P(0, 0, 1) - P(0, 0, 2) \tag{2.6}$$

where $P(m=3)$ [$P(m=4)$] is the steady-state probability of the system being in machine state 3 [4], and $P(0, 0, 1)$ [$P(0, 0, 2)$] is the steady-state probability of the system being idle while in machine state 1 [2]. ($P(x, y, m)$ is the steady-state probability of the state (x, y, m).) U_{ps} can be obtained from the markovian analysis discussed in the next section. Generally, we have

$$U_{rp} < U < U_{ps} \tag{2.7}$$

3. Markovian analysis

Having defined the operational rules and key variables, the Markov model may be set out. State transition diagrams for the various states are described and classified according to whether the system is in processing or in set up. Note that at a boundary (i.e. where at least one of the buffers is either empty or full) certain types of state transitions cannot take place. For example, with a buffer of a certain class full, arrivals of that class cannot be accommodated; when a buffer is empty, a change of the machine state to set up may occur, but no direct processing is allowed.

Representing processing states

In any processing state ($m = 1$ or 2), jobs of the set-up class may increase (as jobs of that class arrive) or decrease (as jobs are processed). The alternate class, however, is non-decreasing. After finishing all the jobs of the current class, the machine is switched to set up for the other class, if there are some jobs of that class in its buffer. The state transition thus has the general form depicted in Fig. 3, assuming $m = 1$ and ignoring all boundary conditions. Note that the $m = 2$ situation is analogous, with $m = 1$ replaced by $m = 2$, $m = 3$ replaced by $m = 4$, and s'_{12} replaced by s'_{21}.

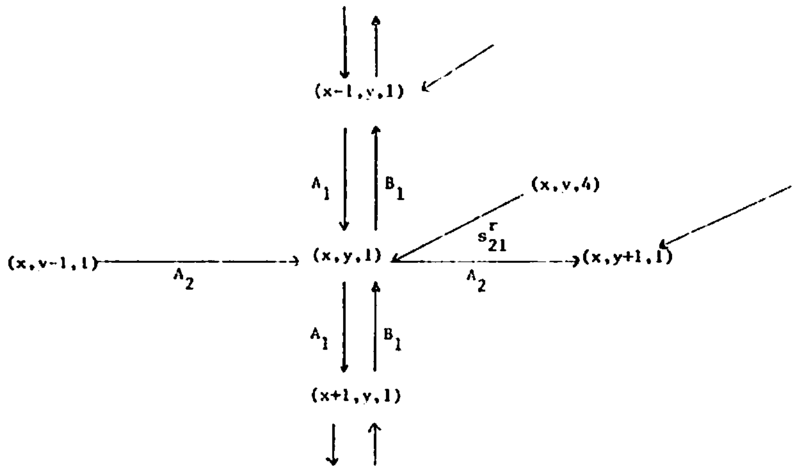

Figure 3. Processing states.

Representing set-up states

During the set-up interval, both job classes may arrive. Jobs of both classes are thus non-decreasing. Figure 4 shows the general form of state transitions in a set-up state, assuming $m = 3$ and ignoring all boundary conditions. Again, the $m = 4$ case is similar.

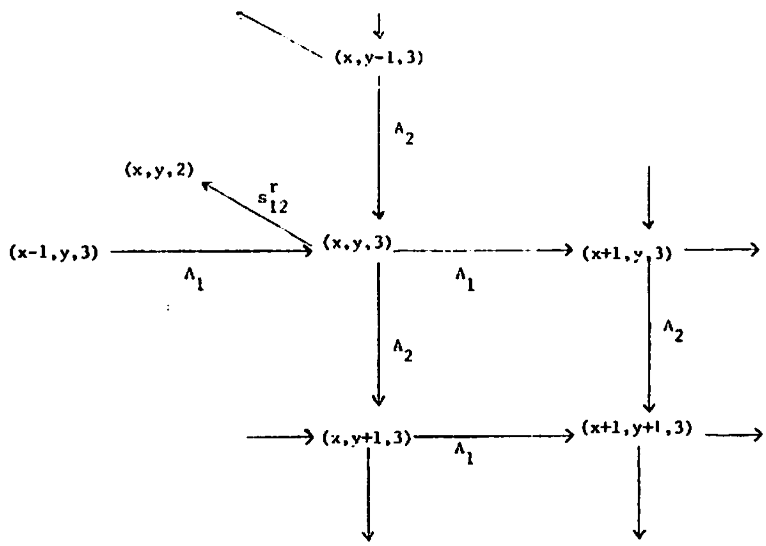

Figure 4. Set-up states.

Global balance equations

For the system to be in equilibrium, the rate going into a particular state should equal the rate out of the state (balance or equilibrium condition). The possible transitions out of the general Class 1 processing state $(k, j, 1)$ include a Class 1 job arrival, a Class 1 departure, and a Class 2 arrival. Transitions into the state are from the state $(k-1, j, 1)$ for a Class 1 job arrival, from the state $(k+1, j, 1)$ for a Class 1 departure, from the state $(k, j-1, 1)$ for a Class 2 arrival, and from $(k, j, 4)$ for the completion of a set up. The general form of the global balance equation for a Class 1 processing state is therefore (again ignoring boundary conditions)

$$(A_1 + B_1 + A_2)P(k,j,1) = A_1 P(k-1,j,1) + B_1 P(k+1,j,1) + A_2 P(k,j-1,1) + s^r_{21} P(k,j,4) \tag{3.1}$$

A similar equation holds for the Class 2 processing state:

$$(A_2 + B_2 + A_1)P(k,j,2) = A_2 P(k,j-1,2) + B_2 P(k,j+1,2) + A_1 P(k-1,j,2) + s^r_{12} P(k,j,3) \tag{3.2}$$

Likewise, the following equations hold for set-up states with $m=3$ and $m=4$:

$$(A_1 + A_2 + s^r_{12}) P(k,j,3) = A_1 P(k-1,j,3) + A_2 P(k,j-1,3) \tag{3.3}$$

$$(A_1 + A_2 + s^r_{21}) P(k,j,4) = A_1 P(k-1,j,4) + A_2 P(k,j-1,4) \tag{3.4}$$

Given $a_i, b_i, s_{ii}, s_{ij}, L_i$, and H, the parameters A_i, B_i and s^r_{ij} can be calculated using (2.1) and (2.2). We can then determine $P(x, y, m)$ by solving a set of simultaneous linear equations derived as above. For a large problem with many zero coefficients, it is often advantageous to use an iterative technique instead of inverting a large matrix for obtaining the solution. The idea is similar to that of fixed point iteration, starting with an initial guess. Two such methods are the Jacobi and Gauss–Seidel (Burden *et al.* 1981) methods. From the form of (3.1)–(3.4), the Stein–Rosenberg theorem says that the Gauss–Seidel method converges faster than the Jacobi method. The Gauss–Seidel method is thus used to solve the set of balance equations for the steady-state probabilities.

Other variables of interest

After obtaining $P(x, y, m)$, many other variables can be obtained. The probability of machine state $m = 3$ is

$$P(m=3) = \sum_{x,y} P(x,y,3) = \frac{1}{s^r_{12}} \Big/ c, \quad \text{or } c = \frac{1}{s^r_{12} P(m=3)} \tag{3.5}$$

Thus, the cycle time can be obtained. Equation (2.6) then becomes

$$U_{ps} = \frac{A_1}{B_1} + \frac{A_2}{B_2} + \frac{1}{s^r_{12} c} + \frac{1}{s^r_{21} c} \tag{3.6}$$

If we think of set ups as jobs that have an arrival rate of $1/c$ and service rates of s^r_{12} and s^r_{21}, then U_{ps} is the total utilization of the four job classes. Clearly,

$$U_{ps} < 1 \tag{3.7}$$

is a stronger non-saturation condition than (2.4). Since the average cycle time c (finite for $U_{ps} < 1$) spans all four machine states, and the output per cycle z_i equals the number of arrivals per cycle, we have

$$z_i = cA_i \quad (i = 1, 2) \tag{3.8}$$

We may also determine the average or expected number of jobs (in lots) in the system

$$E(x) = \sum_{k,j,i} kP(k,j,i), \quad \text{and } E(y) = \sum_{k,j,i} jP(k,j,i) \tag{3.9}$$

4. Results and implications on performance measures

Using the markovian analysis developed, we now examine the numerical result and the sensitivities of performance measures with respect to lot sizes, set-up times (both interclass and intraclass set ups), and arrival rates. In the tables below, we present the average number of lots in the system ($E(x)$ for Class 1 and $E(y)$ for Class 2), the cycle time c, the real processing utilization U_{rp}, the standard utilization U (2.3), and processing plus set-up utilization U_{ps}. The output per cycle z_i, though not included in the tables, can easily be obtained via (3.8). In our analysis, we include the buffer sizes $N_1 = N_2 = 15$, part arrival rates $a_1 = 620{\cdot}44$ parts/hour and $a_2 = 156{\cdot}44$ parts/hour, part service rates $b_1 = b_2 = 1080$ parts/hour, intraclass set-up rates $s_{11} = s_{22} = 1/2$ set up/hour, interclass set-up rates $s_{12} = s_{21} = 1/4$ set up/hour (reduced set-up rate $s'_{12} = s'_{21} = 1/2$ set up/hour), lot sizes $L_1 = 5013$ parts and $L_2 = 2565$ parts, and shift factor $H = 1{\cdot}7$. Parameters A_i and B_i are obtained using (2.1) and (2.2). We then deviate from these nominal values in the ways shown.

The differing definitions of utilization are given by (2.3), (2.5), and (2.6). In examining the effects of decreasing lot sizes on utilizations, the following can be noted. The proportion of cycle time devoted to real processing is independent of the lot sizes if (3.7) is satisfied. On the other hand, both U and U_{ps} increase steadily with decreasing lot sizes. Thus, higher percentages of a cycle are devoted to set ups (mostly intraclass set ups) as the lot sizes decrease. We also see from Table 1 that the average number of Class 1 parts (as opposed to lots) in the system has a local minimum with respect to lot sizes. Figure 5 shows a graph of $E(x)$ versus L_1, demonstrating this effect. This is consistent with the results reported by Karmarkar *et al.* (1985). An intuitive explanation is that, if the lot sizes are small, we expend a higher proportion of the cycle in intraclass set ups, thus increasing the average parts in the system. On the other hand, if the lot sizes are large, we expend longer periods processing large sized lots, thus allowing more parts to collect.

Table 2 examines the effects of intraclass set-up times. ($E(x)$ and $E(y)$ are in lots for all remaining tables in this section.) Again, we observe increases in U and U_{ps} as the intraclass set-up time increases, whereas U_{rp} remains constant.

L_1/L_2	$E(x)$	$E(y)$	c (hr)	U_{rp}	U	U_{ps}
7000/3500	1·010 L 7070 P	0·6345 L 2221 P	44·473	0·4231	0·5800	0·6429
6000/3000	1·122 L 6732 P	0·7372 L 2212 P	39·702	0·4231	0·6061	0·6654
5013/2565	1·295 L 6492 P	0·8855 L 2271 P	35·708	0·4231	0·6404	0·7063
4000/2000	1·640 L 6560 P	1·250 L 2500 P	31·780	0·4231	0·6980	0·7716

Table 1. Effects of changing lot sizes. Note that, for $E(x)$ and $E(y)$, results are shown in lots (L) and parts (P) for comparison across lot sizes.

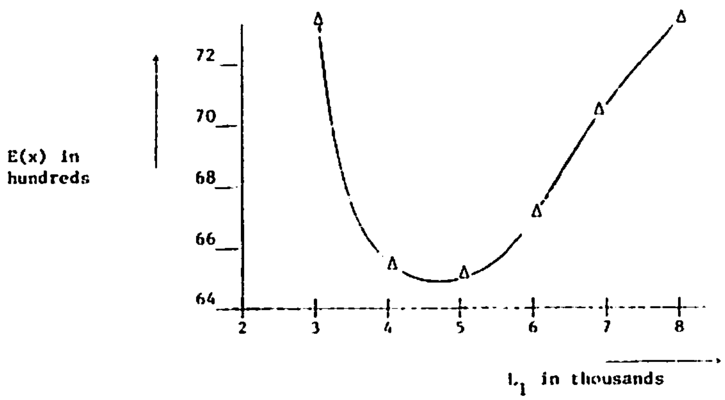

Figure 5. Class 1 work in process minimum.

Interclass set-up times have no effect on the standard utilization, as is evident from (2.3). However, the utilization including interclass set ups, U_{ps}, increases as interclass set-up times increase. The numerical results are summarized in Table 3. In comparing Tables 2 and 3, we see that the system is more sensitive to changes in intraclass set up than to changes in interclass set up. Figure 6 exhibits the difference graphically. In this system, the intraclass set up may recur many times in a cycle, whereas the interclass set up occurs just twice. Intraclass set up, therefore, occupies a greater proportion of the cycle time than interclass set up and has a greater impact upon cycle time.

To conclude this discussion, we now examine sensitivities of the performance measures with respect to arrival rates. The numerical results are summarized in Table 4. Table 4 reflects the general fact that increasing the utilization corresponds to increasing the cycle time and increasing $E(x)$ and $E(y)$. Additional tables showing the effects of service rate and shift factor follow a similar line, and are thus omitted.

$1/s_{11} = 1/s_{22}$	$E(x)$	$E(y)$	c(hr)	U_{rp}	U	U_{ps}
0·0	0·7899	0·4291	29·773	0·4231	0·4231	0·5821
1·0	0·9879	0·5920	32·020	0·4231	0·5318	0·6431
2·0	1·295	0·8855	35·708	0·4231	0·6404	0·7063
3·0	1·835	1·466	42·503	0·4231	0·7491	0·7768

Table 2. Effects of changing intraclass set-up times.

$1/s_{12} = 1/s_{21}$	$E(x)$	$E(y)$	c(hr)	U_{rp}	U	U_{ps}
3·0	1·210	0·8136	34·575	0·4231	0·6404	0·6744
4·0	1·295	0·8855	35·708	0·4231	0·6404	0·7063
5·0	1·392	0·9655	36·819	0·4231	0·6404	0·7358

Table 3. Effects of changing interclass set-up times.

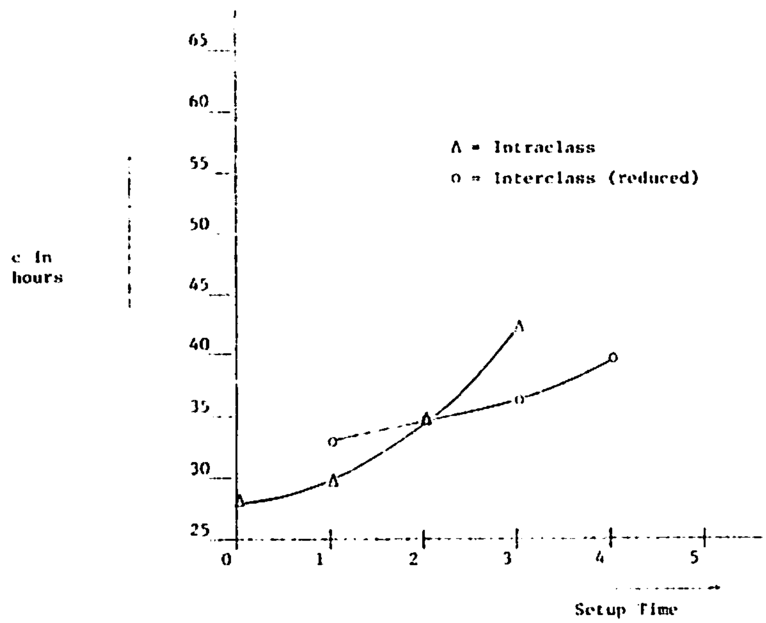

Figure 6. Relative effect of interclass vs. intraclass set-up time on cycle time.

a_1	$E(x)$	$E(y)$	c (hr)	U_{tp}	U	U_{ps}
520	0·9314	0·6435	34·263	0·3684	0·5622	0·6309
620·34	1·295	0·8855	35·708	0·4231	0·6404	0·7063
720	1·809	1·289	39·178	0·4773	0·7181	0·7781
820	2·581	1·994	46·051	0·5321	0·7964	0·8475

Table 4. Effects of changing Class 1 arrival rate ($L_1 = 5013$, $L_2 = 2565$).

5. Random lot sizes

5.1. *System description*

In previous sections, we developed a Markov model based on the assumption of fixed size lots. In reality, however, different sized lots may arrive. We assume now that a finite number of different sized lots may arrive, each with a certain probability. Other assumptions about the manufacturing system remain valid and intact. We wish to determine the effects of this change on results of § 4.

Let L_{ij} be the jth lot size for class i, and K_i be the number of possible lot sizes for class i. We assume that L_{ij} is a multiple of a packet of size p_i, i.e.

$$L_{ij} = jp_i \quad \text{for } i = 1, 2, \quad j = 1, 2, \ldots, K_i$$

Let f_{ij} be the conditional probability of the arrival of L_{ij}, given the arrival of a class i job. With the overall arrival rate for class i lots being A_i, the arrival rate for a class i job with lot size L_{ij} is $A_i f_{ij}$. We also assume that, although an arrival may consist of several packets, processing occurs one packet at a time. Consequently, buffer sizes and state variables are defined in terms of packets. The state of the system is

$$s = (x, y, m)$$

where x is the number of Class 1 packets in system, y is the number of Class 2 packets in system, and m is the machine state as defined before.

Assume that lot arrivals are Poisson distributed, and the set-up time and service time are exponential. The service rate is modified to

$$B_i = \frac{H}{\left[\frac{p_i}{b_i} + \frac{p_i}{M_i}\frac{1}{s_{ii}}\right]} \text{ packets/hour} \tag{5.1}$$

where M_i is the average lot size for class i, i.e.

$$M_i = \sum_{j=1}^{K_i} L_{ij} f_{ij}$$

Comparing this with (2.2), the term p_i/M_i represents the fraction of intraclass set-up time $1/s_{ii}$ allotted for each packet. The generic processing diagram and the set-up diagram are shown in Figs 7 and 8. Note that these diagrams are similar to Figs 3 and 4,

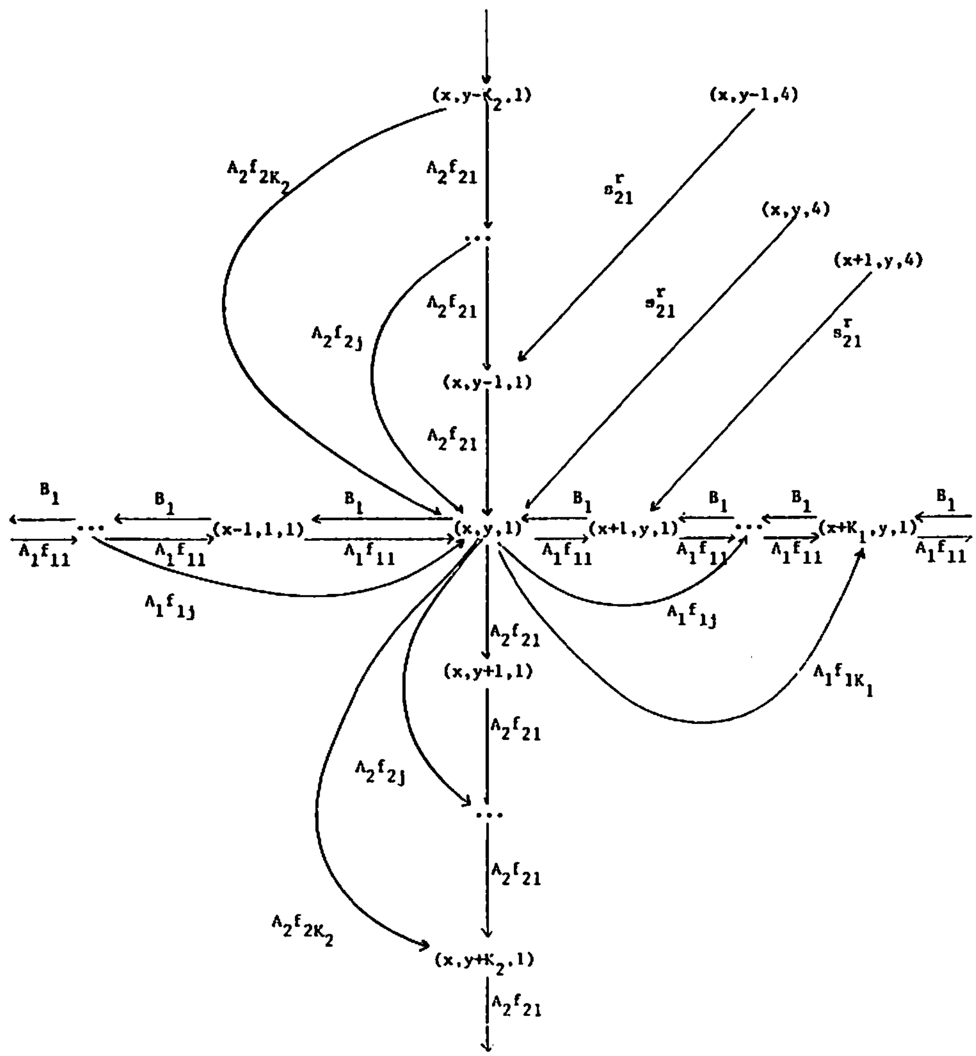

Figure 7. Random lot size, processing states

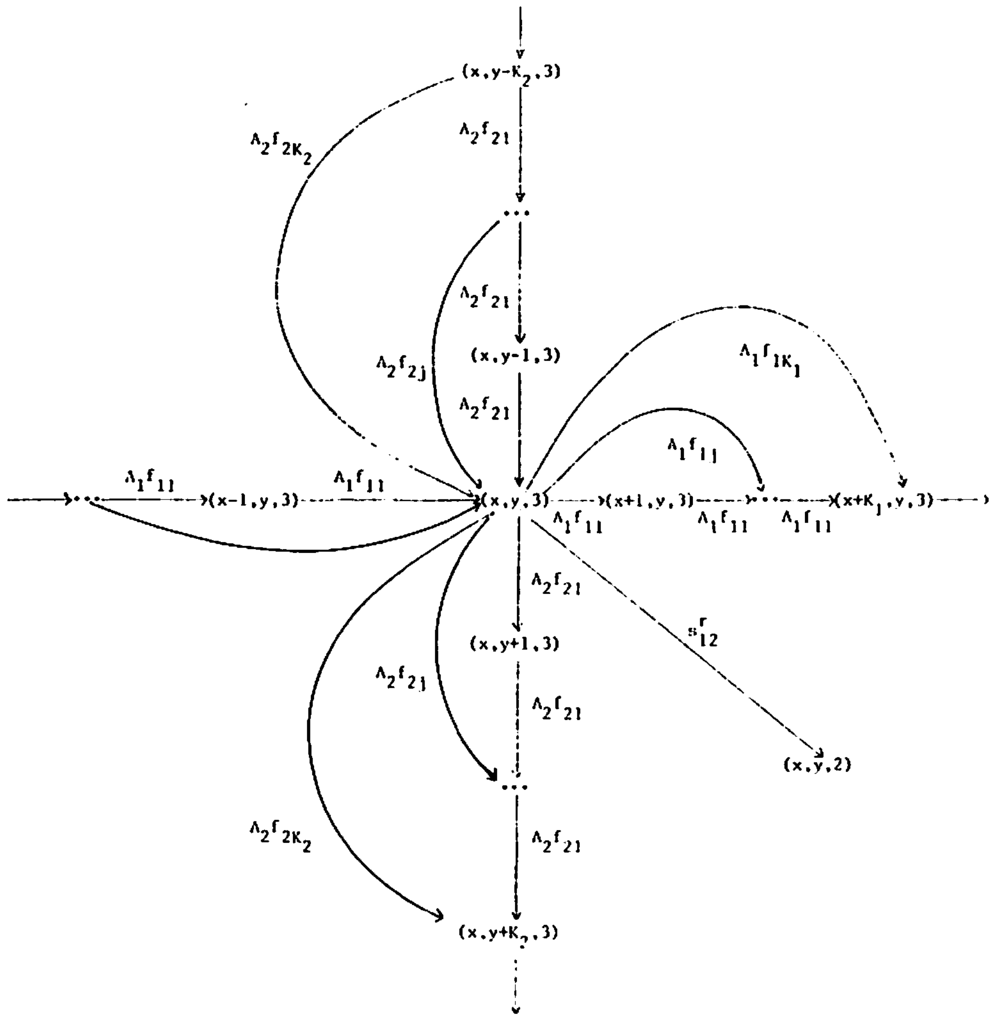

Figure 8. Random lot size, set-up states.

except for the arrival processes. This formulation is consistent with Kleinrock (1975) in the presentation of bulk arrival systems.

5.2. *Comparison with fixed lot sizes*

The following table summarizes the effects of random lot sizes on average queue lengths and cycle time. The nominal values of the parameters are $p_1 = 1700$, $p_2 = 850$, $K_1 = K_2 = 5$, $a_1 = 620{\cdot}34$, $a_2 = 156{\cdot}44$, and $b_1 = b_2 = 1080$ parts/hour. In addition, the f_{ij} used are given in Table 5. Note that $M_1 = 5168$ and $M_2 = 2601$ in comparison to $L_1 = 5013$ and $L_2 = 2565$ of § 4. The interclass set-up time is then varied as shown in Table 6. Note that in the random lot size case, x and y are in terms of packets—we convert them to parts for comparing with the fixed lot size results. In the table, the subscript r indicates random lot size and f indicates fixed lot size.

	Class 1		Class 2	
j	Lot size	f_{1j}	Lot size	f_{2j}
1	1700	0·09	850	0·09
2	3400	0·26	1700	0·18
3	5100	0·28	2550	0·36
4	6800	0·26	·3400	0·27
5	8500	0·11	4250	0·09

Table 5.

$1/s_{21} = 1/s_{12}$	$E_r(x)$	$E_f(x)$	$E_r(y)$	$E_f(y)$	c_r	c_f
3	4894	6066	1634	2087	33·875	34·575
4	5377	6492	1901	2271	35·708	35·708
5	5811	6978	2058	2477	36·495	36·819

Table 6. Random sizes versus fixed lot sizes for varying set-up times.

It can be seen that the use of random lots generally yields a lower queue length. This is somewhat deceiving as it is assumed that packets, instead of lots, leave the system after being processed. The number of parts in the system is therefore smaller. The cycle time, on the other hand, appears to be almost unaffected by this change in arrival and departure patterns. Since the output z_i and variables such as $P(m = 3)$ and U_{rp}—see (3.8), (3.5), and (2.5)—are based upon cycle time, they are also insensitive with respect to the lot sizing assumption. Other numerical testings show similar results. In the next section, we therefore develop the approximate mean value analysis for the fixed lot size case only.

6. Mean value analysis and approximated solutions

The markovian analysis for the fixed lot size case involves the solution of about $4 \cdot N_1 N_2$ balance equations with the same number of unknown variables. For $N_1 = N_2 = 15$, we have about 900 equations with 900 unknowns (using the Eisenberg method, the solution of the problem involves infinite products and sums). This number of equations grows rapidly with N_1 and N_2. To overcome this difficulty, we develop an approximate mean value analysis (MVA). The essential idea is to treat the system as though it were operating in a smooth fashion, with a mean percentage of busy time, mean cycle time, and $E(x)$ and $E(y)$ evolving smoothly over a cycle. The MVA developed here yields satisfactory results for most practical cases (not close to saturation) with greatly reduced computational requirements.

We assume that during Class 1 non-processing times ($m = 2$, 3, and 4), the number of Class 1 jobs in the system experiences a straight-line build up. During processing time ($m = 1$), this build up decreases steadily to a certain level. It then remains constant until the arrival of the first Class 2 job, prompting the next set up (finishing what is there then setting up). This situation is depicted in Fig. 9. Comparing this to Fig. 2 (a sample path of Class 1 jobs), we see that it is a smoothed version of Fig. 2.

Let t_i denote the (mean) time in a cycle where the system is in machine state i, let x_i denote the (mean) number of Class 1 jobs present at the end of t_i, and let y_i denote the

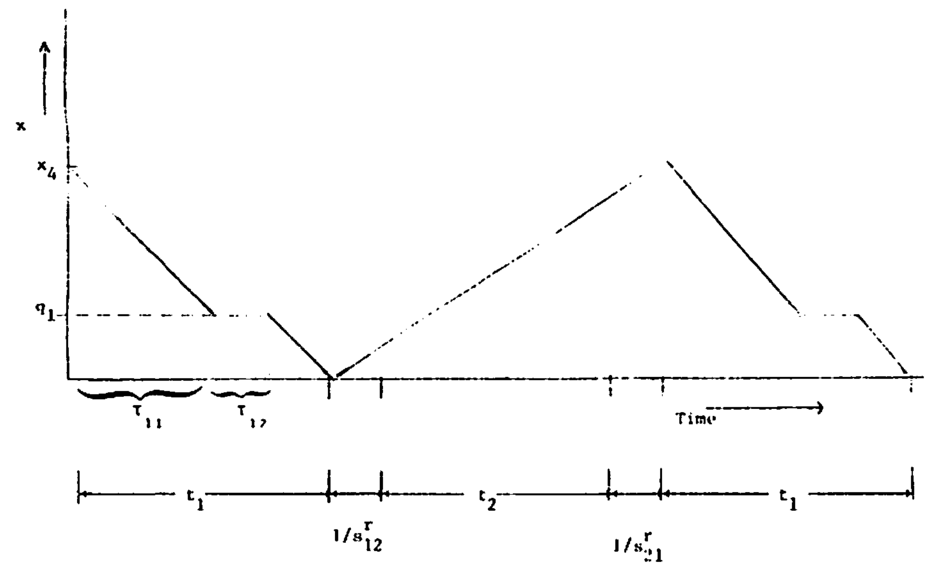

Figure 9. Average cycle.

number of Class 2 lots present at the end of t_i. During t_2, t_3, and t_4, the number of Class 1 lots experiences a straight-line build up starting from $x_1 = 0$ at the end of t_1 (or the beginning of t_3). We therefore have

$$x_4 = A_1(t_3 + t_2 + t_4) = A_1\left(t_2 + \frac{1}{s^r_{12}} + \frac{1}{s^r_{21}}\right) \tag{6.1}$$

This is the same formula used by Sykes (1970), where a system was analysed under a different set-up policy. During Class 1 processing time ($m = 1$), the average number of Class 1 jobs is assumed to decrease steadily to a certain level q_1. It then remains constant until the arrival of the first Class 2 job, prompting the next set up. To approximate q_1, we use the result of Lazowska *et al.* (1984, Ch. 7) that the average queue length for class k in a system with a FCFS (first come first served) discipline is given by

$$q_{k,\text{FCFS}} = \frac{U_k}{(1-U)} \tag{6.2}$$

where U_k is the utilization of class k and U is the overall utilization, as given in (2.2). We assume that q_1 can be approximated by (6.2) with $k = 1$ if $x_4 > q_1$; otherwise $q_1 = x_4$. That is,

$$q_1 = \min(q_{1,\text{FCFS}}, x_4) \tag{6.3}$$

is the average number of Class 1 jobs in the system until the arrival of the first Class 2 job (see Fig. 9). Taking additional arrivals during processing into account, the time required to reach $x = q_1$ from $x = x_4$ is therefore

$$\tau_{11} = \frac{x_4 - q_1}{B_1 - A_1} \tag{6.4}$$

It takes on the average $1/A_2$ hours for a Class 2 job to arrive. Thus, if

$$\frac{1}{A_2} > \frac{1}{s^r_{21}} + \tau_{11} \tag{6.5}$$

the average idle time during $m = 1$ is greater than zero. To quantify this period of time where $x = q_1$, define

$$\tau_{12} = \frac{1}{A_2} - \frac{1}{s^r_{21}} - \tau_{11} \tag{6.6}$$

During τ_{12}, Class 1 jobs are assumed to be freely arriving and processed, or the system may be idle. We also assume that at the end of τ_{12} the first Class 2 job arrives. It then takes $q_1/(B_1 - A_1)$ amount of time to deplete q_1, taking into account any additional Class 1 arrivals during processing. The system is then switched to $m = 3$ to set up for Class 2 (see Fig. 9). We thus have

$$t_1 = \tau_{11} + \tau_{12} + \frac{q_1}{B_1 - A_1} \tag{6.7}$$

On the other hand, if (6.5) is not satisfied, t_1 should just be the time required to deplete x_4:

$$t_1 = x_4/(B_1 - A_1) \tag{6.8}$$

A set of equations analogous to (6.1)–(6.8) can be derived for Class 2 jobs. The cycle time is then given by

$$c = t_1 + t_2 + \frac{1}{s^r_{12}} + \frac{1}{s^r_{21}} \tag{6.9}$$

The above set of equations can be solved iteratively, starting with an initial guess for t_1 and t_2.

The results obtained using MVA are tabulated in Tables 7(*a*)–(*c*) (with subscript

L_1/L_2	U	U_{ps}	c_{MA}	c_{MVA}	% Error
7000/3500	0·5800	0·6429	44·479	44·249	0·52
6000/3000	0·6061	0·6654	39·705	39·665	0·10
5013/2565	0·6404	0·7063	35·759	35·715	0·12
4000/2000	0·6980	0·7716	31·779	30·866	2·88

(*a*)

$1/s_{11} = 1/s_{22}$	U	U_{ps}	c_{MA}	c_{MVA}	% Error
0·0	0·4231	0·5063	29·773	27·120	8·91
1·0	0·5318	0·6075	32·020	29·998	6·31
2·0	0·6404	0·7063	35·759	35·715	0·12
3·0	0·7492	0·8013	42·503	42·108	0·92

(*b*)

a_1	U	U_{ps}	c_{MA}	c_{MVA}	% Error
520·00	0·5622	0·6309	34·260	33·208	3·07
620·34	0·6404	0·7063	35·759	35·715	0·12
720·00	0·7181	0·7781	39·177	37·638	3·93
820·00	0·7964	0·8475	46·051	41·953	8·90

(*c*)

Table 7. Results from mean value analysis and comparisons: (*a*) for changing lot sizes; (*b*) for changing intraclass set-up times; (*c*) for changing Class 1 arrival rate.

MVA), where comparisons are made with results from the markovian analysis (with subscript MA). U_{ps} is also from the markovian analysis. We see from the variety of situations tested that the MVA estimate appears to be quite robust with respect to the parameters used. It is also acceptably accurate over a variety of utilizations.

7. Conclusions

In this paper, we have put forward a method for modelling non-negligible set-up times and introduced different concepts in defining utilization. A markovian analysis yields a set of balance equations, which are solved via a Gauss–Seidel numerical method. The results obtained are then compared to the more elaborate random lot size model and also to the MVA method developed. The cycle time seems to be insensitive to fixed or random lot sizing assumptions. The MVA generally gives satisfactory results for most of the cases tested.

We have seen in this paper that the modelling and analysis of a manufacturing system with set ups is not straightforward. Even with two classes of jobs and one machine, the performance depends upon many factors in complicated ways. For a general manufacturing network, developing realistic models, understanding the underlying principles, and performing numerical testings may be a costly enterprise. The methods presented in this paper may serve as a starting point for the investigation of such works.

Acknowledgments

The research work reported here was supported in part by a contract with the Emhart Corporation. A preliminary version of the work has been presented in the 1987 I.E.E.E. International Conference on Robotics and Automation, Raleigh, North Carolina, March/April 1987. The authors would like to thank Dr Lester D. Servi, Professor J. George Shanthikumar, Dr Renato Conterno, Ms Anne Spence and two anonymous reviewers for their valuable comments.

References

Burden, R. L., Faires, J. D., and Reynolds, A. C., 1981, *Numerical Analysis*, second edition (Boston, Mass.: PWS).

Cooper, R. B., 1970, *Bell Systems tech. J.*, **49**, 399.

Doshe, B. T., 1985, *J. appl. Probab.*, **22**, 419.

Eisenberg, M., 1971, *Ops Res.*, **19**, 386.

Karmarkar, U. S., Kekre, S., Kekre, S., and Freeman, S., 1985, *Interfaces*, **15**, 1.

Kleinrock, L., 1975, *Queueing Systems, Volume I: Theory* (New York: Wiley).

Kuehn, P. J., 1979, *Bell Systems tech. J.*, **58**, 671.

Lavenberg, S. S., 1983, *Computer Performance Modeling Handbook* (New York: Academic Press).

Lazowska, E. E., Zhorjan, J., Graham, G. S., and Sevcik, K. C., 1984, *Quantitative System Performance: Computer System Analysis Using Computer Network Models* (Englewood Cliffs, NJ: Prentice Hall).

Mevert, P., 1968, *Ops Res.*, **16**, 602.

Sauer, C. H., and Chandy, K. M., 1981, *Computer System Performance Modeling* (Englewood Cliffs, NJ: Prentice Hall).

Servi, L. D., 1985, *I.E.E.E. Trans. Commun.*, **33**, 279.

Sykes, J. S., 1970, *Ops Res.*, **18**, 1182.