December, 1983 LIDS-P-1309
Revised January, 1985, November, 1985, and July, 1986

An Efficient Decomposition Method for the Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking

by

Stanley B. Gershwin

35-433
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

Keywords: 343 inventory levels and throughput in transfer lines, 570 Markov chain model of transfer lines, 721 reliability and storage, 683 decomposition approximation of queuing networks

To appear, Operations Research

ABSTRACT

This paper presents an efficient method for the evaluation of performance measures for a class of tandem queuing systems with finite buffers in which blocking and starvation are important phenomena. These systems are difficult to evaluate because of their large state spaces and because they may not be decomposed exactly. The approximate decomposition approach described here is based on system characteristics such as conservation of flow. Comparisons with exact and simulation results indicate that it is very accurate.

ACKNOWLEDGMENTS

This research has been supported by the U. S. Army Human Engineering Laboratory under contract DAAK11-82-K-0018. I am grateful for the support and encouragement of Dr. Benjamin E. Cummings and Mr. Charles Shoemaker. I am also grateful for the suggestions and comments of Ms. Ellen L. Hahne, Professor Rajan Suri, Professor Loren K. Platzman, Dr. Daniel Heyman, and a reviewer.

This paper presents a method for the analysis (*i.e.*, the calculation of throughput and average buffer levels) of a class of tandem queuing systems with finite buffers. Such systems are difficult to treat because of their large state spaces and their indecomposability. The method is based on a model which approximates a (k-1)-buffer system by k-1 single buffer systems. The parameters of the single-buffer systems are determined by relationships among the flows through the buffers of the original system. A simple algorithm is developed to calculate the parameters. Numerical and simulation experience indicate that the method determines thoughput and average buffer levels accurately.

The tandem queuing system in Figure 1 consists of a series of k servers or machines (M_1, M_2, ..., M_k) separated by buffers (B_1, B_2, ..., B_{k-1}) of finite capacity. Material flows from outside the system to M_1, then to B_1, then to M_2, and so forth until it reaches M_k, after which it leaves.

In general, in systems of this type, the machines are assumed to spend a random amount of time processing each item. In this paper, the randomness is due to the failures and repairs of machines. When machine M_i is down, buffer B_{i-1} tends to accumulate material and buffer B_i tends to lose material. If this condition persists, B_{i-1} may become full or B_i may become empty. In that case, machine M_{i-1} is blocked and prevented from working or M_{i+1} is starved and also prevented from working.

While a machine is working (*i.e.*, operational and neither starved nor blocked), a fixed amount of time is required to process a part. This time is the same for all machines and is taken as the time unit. During a time unit when machine M_i is working, it has probability p_i of failing. Its mean time between failures (MTBF) in working time is thus $1/p_i$. After a machine has failed, it is under repair and it has probability r_i of being repaired during a time unit. Its mean time to repair (MTTR) is therefore $1/r_i$. This is measured in clock time, however, not in working time.

The problem is difficult because of the great size of the state space. Each machine can be in two states: operational or under repair. Buffer B_i can be in N_i+1 states: $n_i=0, 1, \ldots, N_i$, where n_i is the amount of material in B_i and N_i is its capacity. As a consequence, the Markov chain representation of a k-machine line with k-1 buffers has a state space of cardinality

$$2^k \prod_{i=0}^{k-1} \left(N_i+1\right).$$

A 20-machine line with 19 buffers each of capacity 10, for example, has over 6.41×10^{25} states.

Decomposition

We approximate the single k-machine line of Figure 1 by a set of k-1 two-machine lines L(i), i = 1, ..., k-1 (Figure 2). The buffer in L(i) has the same capacity as buffer B_i of the k-machine line. Both the upstream ($M_u(i)$) and downstream (($M_d(i)$) machine in each of the lines are assumed to have a geometric working time distribution, described by parameters $p_u(i)$ and $p_d(i)$, respectively. Their repair times are assumed to be geometrically distributed with parameters $r_u(i)$ and $r_d(i)$, respectively. The four parameters are chosen (by the algorithm developed in Section 2) so that the performance of machine $M_u(i)$ ($[M_d(i)]$) closely matches that of the line upstream (downstream) of buffer B_i.

That is, the rate of flow into and out of buffer B_i in line L(i) approximates that of buffer B_i in line L. The probability of the buffer of line L(i) being empty or full is close to that of B_i in L being empty or full. The probability of resumption of flow into (and out of) the buffer in line L(i) in a time unit after a period during which it was interrupted is close to the probability of the corresponding event in L. Finally, the average amount of material in the buffer of line L(i) approximates the material level in buffer B_i in L. In order to find such parameter values, we use relationships among parameters and measures of a transfer line.

Machine $M_u(i)$ models the part of the line upstream of B_i and $M_d(i)$ models the part of line downstream from B_i. There are four parameters per two-machine line (*i.e.*, per buffer in L): $r_u(i)$, $p_u(i)$, $r_d(i)$, $p_d(i)$. The decomposition method requires the determination of four parameters per two-machine line, or a total of $4(k-1)$ parameters. To this end, a system of $4(k-1)$ polynomial equations is derived in Section 2.

Literature Survey

The Gershwin-Schick [1983] model is based on the transfer line model of Buzacott [1967]. Simulation results for models of this type appear in Ho *et al.* [1979], Law [1981], and Vanderhenst *et al.* [1981]. In the last paper, the authors refer to historical data that justifies the assumption of exponentially distributed times between failures and until repairs, although Hanifin, Buzacott and Taraman, in a simulation paper [1975], disagree.

Gershwin and Schick [1983] derive an exact solution for three-machine versions of this system. However, it is difficult to program, ill-behaved, and is not extendable to larger problems. Coillard and Proth [1983] describe a method for a related three-machine model, but again their method does not appear to be useful for longer lines. Ohmi [1981] provides an approximate method for larger systems, but he makes certain restrictive assumptions, such as that no more than one machine is down at any time.

The concept of approximate decomposition of tandem queuing models was discussed by Hunt [1956], Hillier and Boling [1966], Takahashi *et al.* [1980], Altiok [1982], Jafari [1982], Suri and Diehl [1983], and others. In all papers surveyed, the numerical method sweeps from one end of the line (generally the upstream end) to the other. The symmetry of the line (as observed by Muth [1979] and Ammar [1980]) has not previously been exploited. In addition, the interrupted nature of both the arrival and service processes in the decomposed line has not been considered.

The method of this paper has been extended to transfer lines with random processing times in Choong and Gershwin (1985), and to lines with machines that have different processing speeds in Gershwin (1985). It has been generalized to networks involving assembly and disassembly in Gershwin (1986).

A summary of the decomposition technique appears in Gershwin (1984). That publication also includes proofs of two key equations of this paper ((5) and (10)).

Outline of Paper

Section 1 describes two characteristics of production lines that are used in the decomposition method: conservation of flow and the relationship between flow rate and idle time. Section 2 estimates the probability of resumption of flow into and out of a buffer after a period in which material was not flowing. Section 2 also contains the decomposition method and an outline of an algorithm. Numerical results appear in Sections 3 and 4. Section 3 is focussed on the behavior of the method, while Section 4 is concerned with the performance of large transfer lines. Conclusions and new research directions are discussed in Section 5. The Appendix contains formulas for the probability distribution of the two-machine line.

1. TRANSFER LINE CHARACTERISTICS

Model Assumptions

A detailed description of the model is presented in Gershwin and Schick (1983). The assumptions that are relevant to the present method are:

1. Let α_i indicate the repair state of machine M_i. If $\alpha_i = 1$, the machine is operational (sometimes called up); if $\alpha_i = 0$, it is under repair (also called failed or down). All operational machines require the same, fixed amount of time for their operations. That length of time is the time unit.

2. A machine is starved if its upstream buffer is empty. It is blocked if its downstream buffer is full. The first machine is never starved and the last machine is never blocked.

3. The amount of material in a buffer at any time is n, $0 \le n \le N$. A buffer gains or loses at most one piece during a time unit. One piece is inserted into the buffer if the upstream machine is working (*i.e.*, operational and neither starved nor blocked). One piece is removed if the downstream machine is working.

4. When machine i is under repair, it has probability r_i of becoming operational during each time unit. That is,

$$\text{prob}\left[\, \alpha_i(t+1)=1 \;\middle|\; \alpha_i(t)=0 \,\right] = r_i.$$

5. When machine i is working, it has probability p_i of failing. That is,

$$\text{prob}\left[\ \alpha_i(t+1)=0 \ \middle|\ n_{i-1}(t)>0,\ \alpha_i(t)=1,\ n_i(t)<N_i\ \right] = p_i.$$

6. By convention, repairs and failures are assumed to occur at the beginnings of time units, and changes in buffer levels take place at the end of the time units. Thus, during periods while starvation and blockage do not influence buffer i,

$$n_i(t+1) = n_i(t) + \alpha_i(t+1) - \alpha_{i+1}(t+1).$$

More generally,

$$n_i(t+1) = n_i(t) + I_{ui}(t+1) - I_{di}(t+1),$$

where $I_{ui}(t+1)$ is the indicator of whether flow arrives at buffer i from upstream. That is,

$$I_{ui}(t+1) = \begin{cases} 1 & \text{if } \alpha_i(t+1)=1 \text{ and } n_{i-1}(t)>0 \text{ and } n_i(t)<N_i, \\ 0 & \text{otherwise.} \end{cases}$$

The indicator $I_{di}(t+1)$ of flow leaving buffer i is defined similarly.

The state of the system is

$$s = (n_1, ..., n_{k-1}, \alpha_1, ..., \alpha_k).$$

<u>Performance Measures</u>

The <u>production rate</u> (<u>throughput</u>, <u>flow rate</u>, or <u>line efficiency</u>) of machine M_i, in parts per time unit, is

$$E_i = \text{prob}\ [\ \alpha_i = 1,\ n_{i-1} > 0,\ n_i < N_i\ ]. \tag{1}$$

The average level of buffer i is

$$\bar{n}_i = \sum_{s} n_i \text{ prob } (s) \tag{2}$$

Formulas for these and related quantities for two-machine lines are presented in the Appendix.

<u>Conservation of Flow</u>

Because there is no mechanism for the creation or destruction of material, flow is conserved, or

$$E = E_1 = E_2 = ... = E_k. \tag{3}$$

<u>The Flow Rate-Idle Time Relationship</u>

Define e_i to be the <u>isolated production rate</u> of machine M_i. It is what the production rate of M_i would be if it were never impeded by other machines or buffers. It is given by (Buzacott, 1967)

$$e_i = \frac{r_i}{r_i + p_i} \tag{4}$$

and it represents the fraction of time that M_i is operational. The actual production rate E_i of M_i is less because of blocking or starvation. It satisfies

$$E_i = e_i \text{ prob } \Big[\ n_{i-1} > 0 \text{ and } n_i < N_i\ \Big]. \tag{5}$$

For a proof, see Gershwin and Berman (1981), Gershwin (1984).

This result is counter-intuitive. There is no reason to expect that the events of machine failure and adjacent buffers being empty or full are independent. However, failures may occur only while machines are not idle due to starvation or blockage. Furthermore, B_{i-1} can become empty and B_i can become full only when M_i is operational. Therefore, an idle period can be thought of as a hiatus in which a clock, measuring working time until the next machine state change event, is not running. The fraction of non-idle time that M_i is operational is thus the same as the fraction of time it would be operational if it were not in a system with other machines and buffers.

It is useful to observe that

$$\text{prob } [n_{i-1} = 0 \text{ and } n_i = N_i] \approx 0. \tag{6}$$

The probability of this event is small because such states can only be reached from states in which $n_{i-1} = 1$ and $n_i = N_i - 1$ by means of a transition in which

1. Machine M_{i-1} is either under repair or starved, and
2. $\alpha_i = 1$, and
3. Machine M_{i+1} is either under repair or blocked.

The production rate may therefore be approximated by

$$E_i \approx e_i \Big[1 - \text{prob } [\, n_{i-1}=0 \,] - \text{prob } [\, n_i=N_i \,]\Big]. \tag{7}$$

2. DECOMPOSITION METHOD

The decomposition method is based on the equation of conservation of flow (3), the flow rate-idle time relationship (7), and a set of equations ((11) and (12)) developed below. The approach is to characterize the most important features of the transfer line in a simple, approximate way, and to find a solution to the resulting set of equations.

Let $E(i)$ be the efficiency or production rate of two-machine line $L(i)$. Let $p_s(i)$ be the probability of buffer B_i being empty in $L(i)$ and let $p_b(i)$ be the probability of buffer B_i being full in that two-machine line. $E(i)$, $p_s(i)$, and $p_b(i)$ are functions of the four unknowns $r_u(i)$, $p_u(i)$, $r_d(i)$, $p_d(i)$ through the two-machine line formulas in the Appendix.

<u>Conservation of Flow</u>

One set of conditions is related to conservation of flow:

$$E(i) = E(1),\ i=2,\ \dots,\ k-1. \tag{8}$$

<u>Flow Rate-Idle Time</u>

The second set of conditions follows from (7):

$$E(i) = e_i \left[1 - p_s(i-1) - p_b(i)\right],\ i=2,\ \dots,\ k-1. \tag{9}$$

Equation (9), after some manipulation, can be written

$$\frac{p_d(i-1)}{r_d(i-1)} + \frac{p_u(i)}{r_u(i)} = \frac{1}{E(i)} + \frac{1}{e_i} - 2,\ i=2,\ \dots,\ k-1. \tag{10}$$

This is described in Gershwin [1984].

Resumption of Flow

In the following, we derive a set of equations of the form

$$r_u[i] = r_u[i-1]\ X[i] + r_i\ [1-X[i]],\ i = 1, \ldots, k-1 \qquad [11]$$

$$r_d[i-1] = r_d[i]\ Y[i] + r_i\ [1-Y[i]],\ i = 1, \ldots, k-1 \qquad [12]$$

which show the relationship between repair probabilities in neighboring two-machine lines and in the original line. To characterize the repair probabilities in the two-machine lines, we consider the meaning of failure and repair in those systems.

Machine $M_u[i]$ in line $L[i]$ represents, to buffer B_i, everything upstream of B_i in line L. Thus, at time t,

$$\left.\begin{array}{l} \{\alpha_u[i] = 1\} \text{ iff } \{\alpha_i[t] = 1\} \text{ and } \{n_i[t-1] > 0\} \\ \{\alpha_u[i] = 0\} \text{ iff } \{\alpha_i[t] = 0\} \text{ or } \{n_i[t-1] = 0\} \end{array}\right\} \qquad [13]$$

A failure of $M_u[i]$ represents either a failure of machine M_i or the emptying of buffer B_{i-1}. The emptying of B_{i-1}, in turn, is due to a failure of M_{i-1} or the emptying of B_{i-2}. That is, the emptying of B_{i-1} is due to a failure of $M_u[i-1]$. A failure of $M_u[i]$ therefore results from either a failure of machine M_i or a failure of $M_u[i-1]$.

The repair of $M_u[i]$ is the termination of whichever condition was in effect. Consequently, the probability of repair of

$M_u(i)$ is r_i if the cause of failure is the failure of M_i and it is $r_u(i-1)$ otherwise. This leads to equation (11), in which X(i) is the conditional probability that $M_u(i-1)$ is down given that $M_u(i)$ is down.

We now make this more precise. Assuming that $r_u(i)$ is independent of t, the probability that M_i produces a part at time t+1 given that it did not produce and was not blocked at time t is

$$r_u(i) = \text{prob}\Big[n_{i-1}(t)>0,\ \alpha_i(t+1)=1 \;\Big|\; \{n_{i-1}(t-1)=0 \text{ or } \alpha_i(t)=0\} \ \underline{\text{and}}\ n_i(t-1) < N_i\Big]. \tag{14}$$

Breaking down this expression by decomposing the conditioning event, we have

$$r_u(i) = A(i-1)\ X(i) + B(i)\ X'(i), \tag{15}$$

where we define

$$A(i-1) = \text{prob}\Big[n_{i-1}(t) > 0,\ \alpha_i(t+1)=1 \;\Big|\; n_{i-1}(t-1)=0 \text{ and } n_i(t-1) < N_i\Big], \tag{16}$$

$$X(i) = \text{prob}\Big[n_{i-1}(t-1)=0 \text{ and } n_i(t-1) < N_i \;\Big|\; \{n_{i-1}(t-1)=0 \text{ or } \alpha_i(t)=0\} \text{ and } n_i(t-1) < N_i\Big], \tag{17}$$

$$B(i) = \text{prob}\Big[n_{i-1}(t) > 0,\ \alpha_i(t+1)=1 \;\Big|\; \alpha_i(t)=0 \text{ and } n_i(t-1) < N_i\Big], \tag{18}$$

$$X'(i) = \text{prob}\Big[\alpha_i(t)=0 \text{ and } n_i(t-1) < N_i \;\Big|\; \{n_{i-1}(t-1)=0 \text{ or } \alpha_i(t)=0\} \text{ and } n_i(t-1) < N_i\Big]. \tag{19}$$

This decomposition is possible because $\{n_{i-1}(t-1)=0\}$ and $\{\alpha_i(t)=0\}$ are disjoint events. We now evaluate all four conditional probabilities.

The transition in (18) occurs when machine M_i goes from down to up. Therefore,

$$B(i) = r_i.$$

The first quantity, A(i-1), is the probability of buffer B_{i-1} making the transition from empty to non-empty. Buffer B_{i-1} being empty implies that machine M_{i-1} is either down or starved. This is equivalent to saying that $M_u(i-1)$ is down. The only way that B_{i-1} can become non-empty immediately after being empty is for $M_u(i-1)$ to recover. The probability of this event is, by definition, $r_u(i-1)$. Therefore,

$$A(i-1) = r_u(i-1)$$

We now show that

$$X(i) = \frac{p_s(i-1)\ r_u(i)}{p_u(i)\ E(i)} . \tag{20}$$

Equation (17) can be written

$$X(i) = \frac{\text{prob}\left[n_{i-1}(t-1)=0 \text{ and } n_i(t-1) < N_i\right]}{\text{prob}\left[\{n_{i-1}(t-1)=0 \text{ or } \alpha_i(t)=0\} \text{ and } n_i(t-1) < N_i\right]} . \tag{21}$$

We have observed that the probability of B_{i-1} being empty and B_i being full at the same time is small (17). Therefore, the numerator of (21) is approximately the probability of B_{i-1} being empty. We assume B_{i-1} in L(i) has the same probability of being empty as B_{i-1} in L, so the numerator is $p_s(i-1)$.

The denominator of (21) is, according to (13), the probability of the following event (the time arguments are suppressed):

$$\{\alpha_u(i) = 0\} \text{ and } \{n_i < N_i\} \tag{22}$$

The probability can be calculated by making use of

$$r_u(i) \text{ prob } \left[\{\alpha_u(i) = 0\} \text{ and } \{n_i < N_i\} \right]$$

$$= p_u(i) \text{ prob } \left[\{\alpha_u(i) = 1\} \text{ and } \{n_i < N_i\} \right] = p_u(i)\, E(i) \tag{23}$$

(See Schick and Gershwin (1978), page 114.) Thus, the denominator of (21) is

$$\text{prob } \left[\{\alpha_u(i) = 0\} \text{ and } \{n_i < N_i\} \right]$$

$$= \frac{p_u(i)\, E(i)}{r_u(i)} \tag{24}$$

and (20) is established.

A comparison of (17) and (19) shows that the events are complementary. Therefore

$$X'(i) = 1 - X(i) \tag{25}$$

All quantities in (15) have now been evaluated, and the result is (11), in which X(i) is given by (20). A similiar analysis yields equation (12) for the second machine in the i-1'st line, where

$$Y(i) = \frac{p_b(i)\, r_d(i-1)}{p_d(i-1)\, E(i-1)} \,. \tag{26}$$

Finally, there are boundary conditions:

$$\left.\begin{aligned} r_u[1] &= r_1 \\ r_d[k-1] &= r_k \\ p_u[1] &= p_1 \\ p_d[k-1] &= p_k \end{aligned}\right\} \qquad [27]$$

There are a total of 4[k-1] equations among [8], [10], [11], [12], and [27] in 4[k-1] unknowns: $r_u[i]$, $p_u[i]$, $r_d[i]$, $p_d[i]$, $i=1, \ldots, k-1$.

Summary of Approximations

We have made the following approximations:

1. We have assumed that two machines for each buffer can be described whose behavior adequately summarizes the upstream and downstream parts of the line. We have further assumed that they can be represented by geometric up- and down-time models.

2. We have made use of approximation [6]. This affects equation [10] as well as the calculation of X[i] and Y[i] in equations [11] and [12]. [The approximation in the calculation of X[i] occurs in the simplification of the numerator of [21].]

Numerical Technique

Equations [8], [10], [11], [12], and [27] define a two-point boundary value problem of the form

$$\left.\begin{aligned} &f[x[i],\ x[i+1]] = 0,\ i=1,\ \ldots,\ k-1, \\ &x_1[0],\ x_2[0],\ x_3[k],\ x_4[k] \text{ specified} \end{aligned}\right\} \qquad [28]$$

where x[i] is a 4-vector of the parameters of line L[i]; x[i] = [$r_u[i]$, $p_u[i]$, $r_d[i]$, $p_d[i]$]. The nonlinear function f[·] involves the evaluation of E[i], $p_s[i]$, and $p_b[i]$ by means of the two-machine line formulas of the Appendix.

The following algorithm produced the numerical results of the next sections.

1. Initialization. Guess $r_u(i)$, $p_u(i)$, $r_d(i)$, $p_d(i)$, $i=1, \ldots, k-1$.

2. Evaluate $E(1), \ldots, E(k-1)$ and E, their average.

Loop 3. The outer loop seeks E whose value, used in step 3 and the following steps, produces a nearly equal value of E in step 15.

3. Using E in place of $E(i)$, calculate new values of $r_u(i)$ and $r_d(i-1)$ from (11) and (12).

Loop 2. The middle loop seeks a value of $p_d(1)$ so that $E(k-1)$, as evaluated in Loop 1, is close to $E(1)$.

4. Guess $p_d(1)$. (The most recent value may be used.)

5. Evaluate $E(1)$.

6. Calculate $p_u(2)$ from (10) with $i=2$ and with $E(i)$ replaced by E.

Loop 1. For $i=2, \ldots, k-2$ the inner loop seeks a value of $p_d(i)$ so that $E(i)$ is close to $E(1)$.

7. Set $i \leftarrow 1$.

8. Set $i \leftarrow i+1$.

9. Find the value of $p_d(i)$ so that $E(i)$ is suf-

ficiently close to $E[1]$. This is done by searching and evaluating the two-machine line formulas of the Appendix.

10. Calculate $p_u[i+1]$ from (10) where $E[i]$ is replaced by E.

11. If $i = k-2$, go to step 12. Otherwise, go to 8.

12. $p_u[k-1]$ has been determined in step 10. All other parameters of $L[k-1]$ are already known. Evaluate $E[k-1]$.

13. If $E[k-1]$ is sufficiently close to $E[1]$, go to step 15. Otherwise, step 14.

14. Modify $p_d[1]$ to bring $E[k-1]$ closer to $E[1]$. Go to step 5.

15. The value of $E[1] = \ldots = E[k-1]$ is the new E. If it is sufficiently close to the previous E, stop. Otherwise, modify the old E and go to step 3.

16. The average buffer levels of the long line are simply those of the two-machine lines when convergence is reached.

Each loop in the algorithm is characterized by the solution of a single nonlinear equation in one unknown. For example, Step 9 in Loop 1 (the innermost loop) determines $p_d[i]$ such that $|E[i]-E[1]|<\epsilon$ as follows. Let $F_i^1[\cdot]$ be the function, determined by the two-machine formulas, that represents the relationship between $E[i]$ and $p_d[i]$ when all other parameters of Line $L[i]$ are held constant:

$$E(i) = F_i^1(p_d(i)) \tag{29}$$

(The superscript 1 refers to Loop 1.) Let $p_d^{(n-1)}(i)$ and $p_d^{(n)}(i)$ be successive guesses of $p_d(i)$ and let $E^{(n-1)}$ and $E^{(n)}$ be the corresponding values of E(i). The Newton-Raphson method for choosing the next guess $p_d^{(n+1)}(i)$ is

$$p_d^{(n+1)}(i) = p_d^{(n)}(i) - \frac{E^{(n)}(i) - E(1)}{\left(\dfrac{dF_i^1(p_d(i))}{dp_d(i)}\right)} \tag{30}$$

where the derivative is evaluated at $p_d^{(n)}$. Note that E(1) is treated as a constant here. We approximate the derivative by the difference quotient

$$\frac{dF_i^1(p_d(i))}{dp_d(i)} = \frac{E^{(n)}(i) - E^{(n-1)}(i)}{p_d^{(n)}(i) - p_d^{(n-1)}(i)} \tag{31}$$

and (30) becomes

$$p_d^{(n+1)}(i) = \frac{p_d^{(n-1)}(i)\left(E^{(n)}(i) - E(1)\right) - p_d^{(n)}(i)\left(E^{(n-1)}(i) - E(1)\right)}{E^{(n)}(i) - E^{(n-1)}(i)} \tag{32}$$

Similarly, in Loop 2 we seek a value for $p_d(1)$ so that $|E(k-1)-E(1)|<\epsilon$. In this loop, E(1) is a variable. Here we apply Newton's method to the equation

$$E(k-1)-E(1) = F^2(p_d(1)) \tag{33}$$

so that the iteration formula is

$$p_d^{(m+1)}(1) = \frac{p_d^{(m-1)}(1)\left(E^{(m)}(k-1) - E(1)\right) - p_d^{(m)}(1)\left(E^{(m-1)}(k-1) - E(1)\right)}{E^{(m)}(k-1) - E^{(m)}(1) - E^{(m-1)}(k-1) + E^{(m-1)}(1)}. \tag{34}$$

Finally, Loop 3 (the outermost loop) determines a value for E that satisfies the Flow Rate-Idle Time and Resumption of Flow equations as well as Conservation of Flow. Let $\mathcal{E}^{(\ell)}$ be the value of E used in Step 3 (Resumption of Flow) and Step 6 (Flow Rate-Idle Time) in the ℓth iteration. Then let $\hat{E}^{(\ell)}$ be the value of E(1) to which Loop 2 converges. We seek $\mathcal{E}$ such that $\hat{E} = \mathcal{E}$. One possible iteration process is simply

$$\mathcal{E}^{(\ell+1)} = \hat{E}^{(\ell)} = F^3(\mathcal{E}^{(\ell)}) \tag{35}$$

but again, the Newton-Raphson method appears to be more efficient. The iteration formula is

$$\mathcal{E}^{(\ell+1)} = \frac{\mathcal{E}^{(\ell-1)}F^3(\mathcal{E}^{(\ell)}) - \mathcal{E}^{(\ell)}F^3(\mathcal{E}^{(\ell-1)})}{F^3(\mathcal{E}^{(\ell)}) - F^3(\mathcal{E}^{(\ell-1)})} \tag{36}$$

Note that in all three loops, two initial values are required. In all the examples in the following sections, ϵ was large in the initial iterations and was then reduced to 10^{-8}.

Analytical results on the iteration process are not available. Numerical experience suggests that when it converges, it converges to a unique solution which agrees closely with simulation and is independent of the initial guess. There have been some cases in which the algorithm failed to converge. These were lines in which a few machines had much larger values for r and p than the others, or where the lines were very long (more than 20 machines).

The speed of the algorithm is best measured by the number of evaluations of a two-machine line. This is reported below for many cases.

3. NUMERICAL RESULTS-- BEHAVIOR OF THE ALGORITHM

In this section, the behavior of the algorithm is described. The issues investigated are those of accuracy and computational effort. The behavior of transfer lines, as determined by this method, is presented in Section 4.

Three-Machine Cases

It is possible to compare the results of this algorithm with exact results by using the method of Gershwin and Schick [1983]. Table 1A lists the parameters of a set of cases. These cases represent a wide range of three-machine systems. Simulations were also performed for comparison. Each simulation was run for a total of 100,000 time units. The results are shown in Table 1B.

The approximate method produces results that are extremely close to the exact values obtained by solving the Markov chain exactly. The error is very small in the production rate E (less than 0.02%) and only a little larger (less than 2.6%) in the average buffer levels. The approximate results are generally as close or closer to exact as the simulation results.

The table also shows how efficient the method is. The evaluation of a two-machine transfer line is a very small computational burden. Eighty-eight evaluations requires orders of magnitude less computer time than that required for the exact solution or for the simulations. Note that Case 6 has buffers that are too large for the exact method, but that the agreement with simulation is good.

Longer Lines

Exact methods are not available for systems of more than three machines or for three-machine cases with very large buffers. Consequently, other techniques are required to assess the accuracy of the approximation. They include simulation and exact qualitative results. The cases considered here represent a wide range of failure probabilities, repair probabilities, and buffer sizes. The results cover a wide range of production rates and average buffer levels.

There is close agreement between the approximation and simulation results. In most cases, production rates and buffer levels agree to within a few percent. This remains true even for large buffer capacities (over 100) and long lines (20 machines.) There is no obvious trend indicating that the accuracy of the approximation decreases as the line length increases.

Cases in which $r_i = p_i = .1$

The number of evaluations of two-machine lines increases with the length of the line. Figure 3 shows this for two sets of cases in which repair and failure probabilities are all equal to .1. Experience indicates that the number of evaluations appears to be $O(k^3)$, where k is the number of machines.

Most of the approximation and simulation results of this and the next section were produced on the MIT Honeywell 68/DPS computer with the Multics operating system. All times indicated were for runs performed on this machine. Experience indicates that the computer time for the analytic approximation method appears to be much less than that of simulation.

Other runs -- both approximate and simulation -- were performed on an IBM PC and a variety of compatible computers.

Table 2 shows the results of a set of cases with repair and failure probabilities all equal to .1 and buffer sizes all equal to 5. Production rates (E) are indicated for all cases; average buffer levels and computer times are indicated for the 20-machine line. Note the close agreement between approximate and simulated production rates and average levels. Table 3 contains similar results but with all buffer capacities equal to 10.

Reversibility and Symmetry.

Several authors have conjectured (Hillier and Boling, 1977) or shown (Dattatreya, 1978; Muth, 1979; Ammar, 1980; Ammar and Gershwin, 1981) that two tandem queueing systems which are the reverse of one another have the same production rates. In addition (Ammar, 1980; Ammar and Gershwin, 1981), the average levels of corresponding buffers are complementary.

Symmetric lines are their own reverses. The complementarity property applies to symmetrically opposite buffers. That is, for $i = 1, \ldots, k-1$,

$$\bar{n}_i + \bar{n}_{k-i} = N_i = N_{k-i}. \tag{37}$$

Furthermore, the average level in the middle buffer of a symmetric line with an odd number of buffers is equal to exactly half the capacity of that buffer [Ammar, 1980]. That is, if k is even and i = k/2, [37] implies

$$\bar{n}_{k/2} = \frac{1}{2}\bar{N}_{k/2} \qquad [38]$$

The results of the decomposition method agree with these observations exactly. Simulation, however, produces only approximate agreement. These properties are exhibited in the last case of the Table 2;, the first case of Table 3; Cases 1-4; 6; 14 and 15; 18 and 23; and 19 and 21.

Perturbed cases

Tables 4A and 4B contain a set of four-machine cases that are close to three-machine cases already considered. It includes machines that are nearly, but not quite, perfect. If those machines were perfect, these cases would be the same as the earlier cases. We expect the behavior of the perturbed cases to be close to that of the others.

Case 7 was chosen to be close to Case 4. Case 7 is a four-machine line whose second machine almost never fails. The first two buffers of Case 7 have a total capacity which is equal to that of B_1 of Case 4, and the other corresponding machines and buffers are the same. The sum of the average levels of the first two buffers of Case 7 is 5.977. This should be compared with the average level of B_1 of Case 4 which is 6.063. In addition, $\bar{n}_3$ of Case 7 is not far from $\bar{n}_2$ of Case 4.

Similarly, Case 8 was chosen to be close to Case 3. Here the highly efficient machine is the last. The first three machines and two buffers of Case 8 are identical to Case 3. The results are also in close agreement. Furthermore, the level in the last buffer can be explained.

Because the last machine nearly never fails, the last buffer never has more than one piece in it. If it were initialized with more than one piece, the number of pieces would diminish whenever the third machine was down or the second buffer was empty. If such a condition persisted, the third buffer would become empty. Eventually, the third buffer would have exactly one piece if the third machine were working and the second buffer were non-empty; and it would be empty otherwise. That is, it would have one piece whenever the three-machine subset was producing parts and no pieces when it was not.

To calculate the expected level in the third buffer, we need to know the fraction of time the subset is producing parts. We do: it is the production rate of the three-machine subset. As a consequence, the average number of parts in the last buffer (when the last machine is almost perfectly reliable) is equal to the production rate, in parts per time unit.

The simulation results also bear this out. Cases 4 and 9 are in the same relationship as Cases 3 and 8 and they demonstrate this behavior.

Comparison with Published Simulations

Cases 10-12 (Table 5A) are a set of 5-machine lines from example (c) of Ho, Eyler, and Chien (1979) and the simulation results (production rates) of Table 5B are theirs (written in a slightly different form). In each of these cases,

r_i=.005, p_i=.001, i=1, ..., 5

The approximate and simulation production rates are in good agreement. Furthermore, both methods calculate production rates for Case 12 which are greater than those for Case 11, which are greater than those for Case 10.

Cases 13-23 (Table 6A) are taken from Law (1981) and again the simulation results (Table 6B) come from the cited paper. These are all 4-machine lines in which

r_i=.05, p_i=.005, i=1, ..., 4

The decomposition results are in close agreement with simulated values. However, Law draws conclusions that are not supported by reversibility or decomposition results. Cases 13-17 are Buffer Allocations 1-5 of Table 4 of Law (1981). Law finds Allocations 1 and 5 to be statistically indistinguishable and to have a greater production rate than Allocations 2, 3, and 4, which are also indistinguishable. However, we find that Allocations 2 and 3 (*i.e.*, Cases 14 and 15) are the same (due to reversibility). Their production rate is less than that of Allocation 4 (Case 16) which is less than that of Allocation 5 (Case 17) which is less than that of Allocation 1 (Case 13).

Cases 18-23 (Table 7) are Patterns 1-6 of Table 2 of Law (1981). Law is seeking to find the best sequence of four machines of which two have high efficiency (H) and two have low efficiency (L). The parameters of the H machines are p=.0025 and r=.05. The L machines have p=.02 and r=.05. The three buffers each have capacity 19 in all cases. Law finds that the patterns, in order of production rate, are

(23,22), 21, 18, 19, 20.

using the case numbers of this paper. That is, Cases 23 and 22 cannot be distinguished, but their production rate is less than that of Case 21, and so forth.

We find instead the following order:

22, (18,23), (19,21), 20.

The cases that we find indistinguishable are the reverses of one another.

Cases with Different Machines

Many of the lines treated here have had identical machines. Table 8B shows the results of two cases with different machines, whose parameters are listed in Table 8A. These are six- and twelve-machine cases that consist of the three machines of Case 5, doubled and quadrupled. The buffers are all of capacity 10. The simulation and approximation results are in close agreement.

4. NUMERICAL RESULTS-- BEHAVIOR OF TRANSFER LINES

Figure 4 demonstrates how the production rate varies with the length of the line for two sets of cases. It is not surprising that production rates decrease as lines grow. This graph does not, however, settle the question of whether the production rate approaches zero as the length goes to infinity.

Figure 5 shows how material is distributed in the buffers for two twenty-machine cases. Note how the average buffer level in the tenth buffer is equal to half the buffer's capacity. This is due to the symmetry of both cases, as discussed above. All the other buffers exhibit complementarity. For example, in the case where $N_i=10$ for all i, $\bar{n}_3+\bar{n}_{17}=10$.

Cases 3, 4, and 6 of Section 3 show that as the buffer sizes increase, the production rate increases. In addition, the average buffer levels, as a fraction of capacities, approach .5. In general, production rate increases with buffer capacity, but buffer levels do not always approach half of buffer capacity as capacity increases.

Figure 6 shows the effect of increasing total buffer space and the effect of varying the distribution of buffer space. A system of nine identical machines with parameters r=.019 and p=.001 is considered. In one set of cases, there are eight buffers with equal capacities. In the other, there are two buffers with equal capacities, located between the third and fourth machines and between the sixth and seventh machines.

[Each set of three machines is equivalent to one machine with parameters r=.01898 and p=.002997.]

The reason the 8-buffer lines are better than the 2-buffer lines when the space is large is that the production rate is limited by the least productive machine. In the former case, the limiting efficiency is .95; in the latter it is .8636 since each set of three machines acts like a single machine. The behavior when the buffer space is small may be due to the model of blocking and starvation used here; with other models, the 8-buffer distribution may always be better than the 2-buffer distribution. Note that the total average buffer levels are the same in the two sets of cases: half the total capacity (due to symmetry).

The facts that Case 20 is the best of Cases 18-23, that 13 and 17 are the best of 13-17, and that 33 is the best of 31-33 may be instances of the "bowl phenomenon" (Hillier and Boling, 1966), which is that, for maximal production rate, more buffer space should be allocated to the middle of a line.

5. CONCLUSIONS AND FURTHER RESEARCH

A new method has been found for the analysis of tandem queuing systems with finite buffers in which blocking is important. Exact and simulation results indicate that the method, while approximate, is quite accurate. Current research is aimed at extending this work in two directions: other service process behavior, such as reliable and unreliable machines with exponential and other processing time distributions [Choong and Gershwin, 1985; Gershwin, 1985]; and assembly/disassembly networks [Gershwin, 1986]. Future efforts will be devoted to systems such as Jackson-like networks with blocking.

Research is needed to determine precisely the performance of this method. That is, analytic techniques rather than only numerical experiments should be used to develop bounds on the accuracy of the method and to determine convergence properties of the algorithm.

REFERENCES

T. Altiok (1982), "Approximate Analysis of Exponential Tandem Queues with Blocking," European Journal of Operations Research, Vol. 11, 1982.

M. H. Ammar (1980), "Modelling and Analysis of Unreliable Manufacturing Assembly Networks with Finite Storages," MIT Laboratory for Information and Decision Systems Report LIDS-TH-1004.

M. H. Ammar and S. B. Gershwin (1981), "Equivalence Relations in Queuing Models of Manufacturing," Proceedings of the Nineteenth IEEE Conference on Decision and Control.

J. A. Buzacott (1967), "Markov Chain Analysis of Automatic Transfer Lines with Buffer Stock," Ph.D. Thesis, Department of Engineering Production, University of Birmingham.

J. A. Buzacott (1967), "Automatic Transfer Lines with Buffer Stocks," International Journal of Production Research, Vol. 6 .

Y. F. Choong and S. B. Gershwin (1985), "A Decomposition Method for the Approximate Evaluation of Capacitated Transfer Lines with Unreliable Machines and Random Processing Times," MIT Laboratory for Information and Decision Systems Report LIDS-P-1476; to appear in IIE Transactions.

P. Coillard and J.-M. Proth (1983), "Sur L'Effet des Stocks Tampons Dans Une Fabrication en Ligne," INRIA Rapports de Recherche, Institut National de Recherche en Informatique et en

Automatique, Domaine de Voluceau, Roquencourt, Le Chesnay, France.

E. S. Dattatreya [1978], "Tandem Queueing Systems with Blocking," Ph. D. thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley.

S. B. Gershwin [1984], "An Efficient Decomposition Method for the Approximate Evaluation of Production Lines with Finite Storage Space," in Analysis and Optimization of Systems, Proceedings of the Sixth International Conference on Analysis and Optimization of Systems, Part 2, Volume 63 of Lecture Notes in Control and Information Sciences, Springer-Verlag, 1984.

S. B. Gershwin [1985], "Representation and Analysis of Transfer Lines with Machines that have Different Processing Rates," MIT Laboratory for Information and Decision Systems Report LIDS-P-1446; to appear in Annals of Operations Research.

S. B. Gershwin [1986], "Assembly/Disassembly Systems: An Efficient Decomposition Algorithm for Tree-Structured Networks," MIT Laboratory for Information and Decision Systems Report LIDS-P-1579, July, 1986.

S. B. Gershwin and O. Berman [1981], "Analysis of Transfer Lines Consisting of Two Unreliable Machines with Random Processing Times and Finite Storage Buffers," AIIE Transactions Vol. 13, No. 1, March 1981.

S. B. Gershwin and I. C. Schick [1983], "Modeling and Analysis of Three-Stage Transfer Lines with Unreliable Machines and Finite

Buffers," Operations Research, Vol. 31, No.2, pp 354-380, March-April 1983.

L. E. Hanifin, J. A. Buzacott, and K. S. Taraman [1975], "A Comparison of Analytical and Simulation Models of Transfer Lines," SME Technical Paper EM75-374, Society of Manufacturing Engineers, 20501 Ford Road, Dearborn, Michigan 48128.

F. S. Hillier and R. W. Boling [1966], "The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times," Journal of Industrial Engineering, Vol. 17, No. 12, December, 1966.

F. S. Hillier and R. Boling [1977], "Toward Characterizing the Optimal Allocation of Work in Production Lines with Variable Operation Times," in Advances in Operations Research, Proceedings of EURO II, Marc Reubens, Editor; North-Holland, Amsterdam.

Y. C. Ho, M. A. Eyler, and T. T. Chien [1979], "A Gradient Technique for General Buffer Storage Design in a Production Line," International Journal of Production Research, Vol. 17, No. 6, pp 557-580, 1979.

G. C. Hunt [1956], "Sequential Arrays of Waiting Lines," Operations Research, Volume 4, 1956.

M. A. Jafari [1982], Ph. D. Thesis proposal, Syracuse University.

S. S. Law [1981], "A Statistical Analysis of System Parameters in Automatic Transfer Lines," International Journal of Production Research, Vol. 19, No. 6, pp 709-724, 1981.

E. J. Muth (1979), "The Reversibility Property of Production Lines," Management Science, Vol.25, No. 2.

T. Ohmi (1981) "An Approximation for the Production Efficiency of Automatic Transfer Lines with In-Process Storages," AIIE Transactions, Volume 13, No. 1, March 1981.

I. C. Schick and S. B. Gershwin (1978), "Modelling and Analysis of Unreliable Transfer Lines with Finite Interstage Buffers," Massachusetts Institute of Technology Electronic Systems Laboratory Report ESL-FR-834-6, September, 1978.

R. Suri and G. Diehl (1983), "A Variable Buffer-Size Model and its Use in Analyzing Closed Queuing Networks with Blocking," Harvard University, Division of Applied Sciences, September, 1983.

Y. Takahashi, H. Miyahara, and T. Hasegawa (1980), "An Approximation Method for Open Restricted Queuing Networks," Operations Research, Vol. 28, No.3, Part I, May-June 1980.

P. Vanderhenst, F. V. Van Steelandt, and L. F. Gelders (1981), "Efficiency Improvement of a Transfer Line Via Simulation," Katholieke Universiteit Leuven, Faculteit Toegepaste Wetenschappen Afdeling Industrieel Beleid, Belgium; Report 81-04, January 1981.

APPENDIX

Steady-State Probabilities and Performance Measures for Two-Machine Lines

In the following, $p[n, \alpha_1, \alpha_2]$ is the probability that there are n parts in the buffer and that M_i is in state α_i. These probabilities are taken from Gershwin and Schick [1983].

$$p[0,0,0] = 0$$

$$p[0,0,1] = C\ X\ \frac{r_1 + r_2 - r_1 r_2 - r_1 p_2}{r_1 p_2}$$

$$p[0,1,0] = 0$$

$$p[0,1,1] = 0$$

$$p[1,0,0] = C\ X$$

$$p[1,0,1] = C\ X\ Y_2$$

$$p[1,1,0] = 0$$

$$p[1,1,1] = \frac{C\ X}{p_2}\ \frac{r_1 + r_2 - r_1 r_2 - r_1 p_2}{p_1 + p_2 - p_1 p_2 - r_1 p_2}$$

$$p[N-1,0,0] = C\ X^{N-1}$$

$$p[N-1,0,1] = 0$$

$$p[N-1,1,0] = C\ X^{N-1}\ Y_1$$

$$p[N-1,1,1] = \frac{C\ X^{N-1}}{p_1}\ \frac{r_1 + r_2 - r_1 r_2 - p_1 r_2}{p_1 + p_2 - p_1 p_2 - p_1 r_2}$$

$$p(N,0,0) = 0$$

$$p(N,0,1) = 0$$

$$p(N,1,0) = C\ X^{N-1}\ \frac{r_1 + r_2 - r_1 r_2 - p_1 r_2}{p_1 r_2}$$

$$p(N,1,1) = 0$$

$$p(n,\ \alpha_1,\ \alpha_2) = C\ X^n\ Y_1^{\alpha_1}\ Y_2^{\alpha_2},\ 2 < n < N - 2$$

where

$$Y_1 = \frac{r_1 + r_2 - r_1 r_2 - r_1 p_2}{p_1 + p_2 - p_1 p_2 - p_1 r_2}$$

$$Y_2 = \frac{r_1 + r_2 - r_1 r_2 - p_1 r_2}{p_1 + p_2 - p_1 p_2 - r_1 p_2}$$

$$X = \frac{Y_2}{Y_1}$$

and C is a normalizing constant. Performance measures are given by

$$E = \sum_{\substack{n > 0 \\ \alpha_2 = 1}} p(n,\ \alpha_1,\ \alpha_2) = \sum_{\substack{n < N \\ \alpha_1 = 1}} p(n,\ \alpha_1,\ \alpha_2)$$

$p_s = p(0,0,1)$, the probability of starvation

$p_b = p(N,1,0)$, the probability of blockage

$$\bar{n} = \sum_{\text{all } n,\ \alpha_1,\ \alpha_2} n\ p(n,\ \alpha_1,\ \alpha_2)$$

Note that

$$E = e_1 \, (1 - p_b)$$

$$= e_2 \, (1 - p_s).$$

Case	r_1	p_1	N_1	r_2	p_2	N_2	r_3	p_3
1	.1	.05	5	.1	.05	5	.1	.05
2	.05	.002	5	.05	.002	5	.05	.002
3	.1	.1	5	.1	.1	5	.1	.1
4	.1	.1	10	.1	.1	10	.1	.1
5	.07	.01	10	.1	.013	10	.05	.007
6	.1	.1	100	.1	.1	100	.1	.1

Table 1A. Three-Machine Cases

	E	$\bar{n}_1$	$\bar{n}_2$	Evaluations
Case 1				
Approximate Results	.4542	3.088	1.912	70
Exact Results	.4545	3.063	1.937	
Simulation 1	.4567	3.030	1.953	
Simulation 2	.4554	3.034	1.877	
Simulation 3	.4514	3.072	1.969	
Case 2				
Approximate Results	.8993	3.001	1.999	32
Exact Results	.8992	2.987	2.013	
Simulation 1	.9031	3.087	2.024	
Simulation 2	.9030	3.135	2.056	
Simulation 3	.8883	2.998	1.982	
Case 3				
Approximate Results	.3105	3.101	1.899	68
Exact Results	.3109	3.075	1.925	
Simulation 1	.3075	3.060	1.918	
Simulation 2	.3126	3.096	1.912	
Simulation 3	.3109	3.086	1.926	
Case 4				
Approximate Results	.3563	6.063	3.937	64
Exact Results	.3556	6.035	3.965	
Simulation 1	.3544	6.089	3.908	
Simulation 2	.3572	6.109	4.024	
Simulation 3	.3552	6.011	3.903	
Case 5				
Approximate Results	.7546	6.182	3.349	86
Exact Results	.7539	6.120	3.437	
Simulation 1	.7548	6.107	3.443	
Simulation 2	.7478	6.412	3.523	
Simulation 3	.7558	6.074	3.310	
Case 6				
Approximate Results	.4722	56.60	43.40	88
Simulation	.4676	58.59	45.71	

Table 1B. Results of Three-Machine Cases

Line length, k	Quantity	Eval's	Approx.	Sim. 1	Sim. 2
4	E	116	.2806	.2830	.2809
5		238	.2630	.2632	.2647
6		399	.2516	.2514	.2519
7		583	.2438	.2418	.2458
8		996	.2382	.2365	.2332
10		2030	.2309	.2306	.2251
13		4110	.2248	.2212	.2193
17		7705	.2206	.2119	.2132
20	E	9467	.2188	.2107	.2117
	$\bar{n}_1$		3.902	3.850	3.909
	$\bar{n}_5$		2.935	3.106	2.856
	$\bar{n}_{10}$		2.500	2.684	2.543
	$\bar{n}_{19}$		1.098	1.086	1.119
	time		6.897	248.169	247.685

Table 2. Performance of Systems with N=5.

Line length, k	Quantity	Eval's	Approx.	Sim. 1	Sim. 2
4	E	131	.3352	.3341	.3390
	$\bar{n}_1$		6.535	6.604	6.556
	$\bar{n}_2$		5.000	5.133	4.992
	$\bar{n}_3$		3.465	3.637	3.564
5	E	234	.3228	.3232	.3175
6		386	.3149	.3075	.3127
7		614	.3095	.3033	.3057
8		1387	.3056	.2970	.2957
10		1899	.3006	.2934	.2910
13		3521	.2964	.2830	.2861
17		10389	.2936	.2768	.2813
20	E	17185	.2924	.2774	.2759
	time		12.155	262.188	261.079

Table 3. Performance of Systems with N=10.

Case	Parameters
7	$r_i=.1,\ i=1,\ \ldots,\ 4$ $p_1=p_3=p_4=.1,\ p_2=.000001$ $N_1=N_2=5,\ N_3=10$
8	$r_i=.1,\ i=1,\ \ldots,\ 4$ $p_1=p_2=p_3=.1,\ p_4=.000001$ $N_1=N_2=N_3=5$
9	$r_i=.1,\ i=1,\ \ldots,\ 4$ $p_1=p_2=p_3=.1,\ p_4=.000001$ $N_1=N_2=N_3=10$

Table 4A.—Parameters of Perturbed Cases.

Case	Quantity	Eval's	Approx.	Sim. 1	Sim. 2
7	E	133	.3537	.3550	.3619
	$\bar{n}_1$		2.656	2.631	2.632
	$\bar{n}_2$		3.321	3.406	3.403
	$\bar{n}_3$		3.876	3.986	3.975
8	E	128	.3105	.3095	.3093
	$\bar{n}_1$		3.101	3.058	3.086
	$\bar{n}_2$		1.899	1.917	1.882
	$\bar{n}_3$		.3107	.3095	.3093
9	E	115	.3563	.3546	.3619
	$\bar{n}_1$		6.063	6.042	5.897
	$\bar{n}_2$		3.937	3.992	3.843
	$\bar{n}_3$		.3566	.3548	.3619

Table 4B. Results of Perturbed Cases.

Case	Buffer Sizes
10	N_i=100, i=1, ..., 4
11	N_1=80, N_2=122, N_3=126, N_4=72
12	N_1=60, N_2=146, N_3=142, N_4=52

Table 5A. Parameters of Ho, Eyler, Chien [1979] Simulations.

Case	Eval's	Approx.	Sim
10	293	.6007	.5927
11	289	.6034	.5948
12	295	.6045	.5958

Table 5B. Production Rates from Approximation and Simulation.

Case	Buffer Sizes
13	N_i=40, i=1, ..., 3
14	N_1=10, N_2=40, N_3=70
15	N_1=70, N_2=40, N_3=10
16	N_1=50, N_2=40, N_3=50
17	N_1=30, N_2=60, N_3=30

Table 6A. Parameters of Law [1981] Simulations.

Case	Eval's	Approx.	Sim.
13	128	.8337	.8380, .8344
14	133	.8202	.8212
15	94	.8202	.8219
16	117	.8273	.8200
17	137	.8336	.8364

Table 6B. Production Rates from Approximation and Simulation.

Case	Machine Sequence	Eval's	Approx.	Sim
18	LLHH	898	.6363	.6534
19	LHLH	111	.6707	.6651
20	LHHL	122	.6906	.6767
21	HLHL	148	.6707	.6442
22	HLLH	137	.6351	.6192
23	HHLL	146	.6363	.6173

Table 7. Sequences from Law (1981) and Results.

Case	Line Length, k	Parameters
24	6	$r_1=r_4=.07$, $r_2=r_5=.1$, $r_3=r_6=.05$ $p_1=p_4=.01$, $p_2=p_5=.013$, $p_3=p_6=.007$ $N_i=10$, $i=1, ..., 5$
25	12	$r_1=r_4=r_7=r_{10}=.07$ $r_2=r_5=r_8=r_{11}=.1$, $r_3=r_6=r_9=r_{12}=.05$, $p_1=p_4=p_7=p_{10}=.01$, $p_2=p_5=p_8=p_{11}=.013$, $p_3=p_6=p_9=p_{12}=.007$, $N_i=10$, $i=1, ..., 11$

Table 8A. Parameters of Lines with Different Machines.

Case	Eval's	Approx.	Sim
24	1582	.6692	.6684
25	4455	.6098	.5916

Table 8B. Results from Lines with Different Machines.

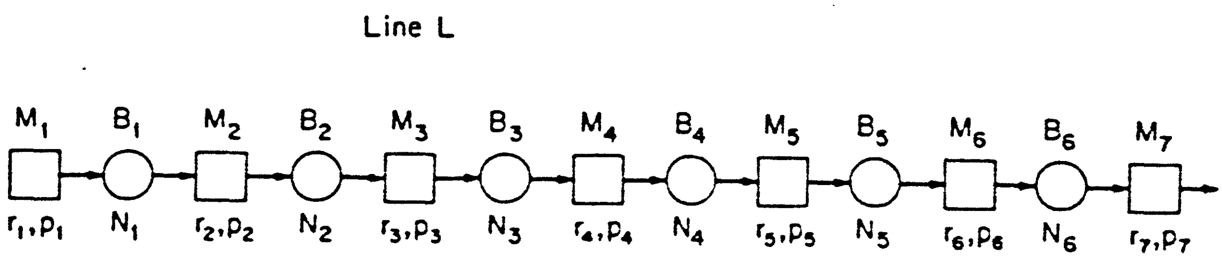


Figure 1

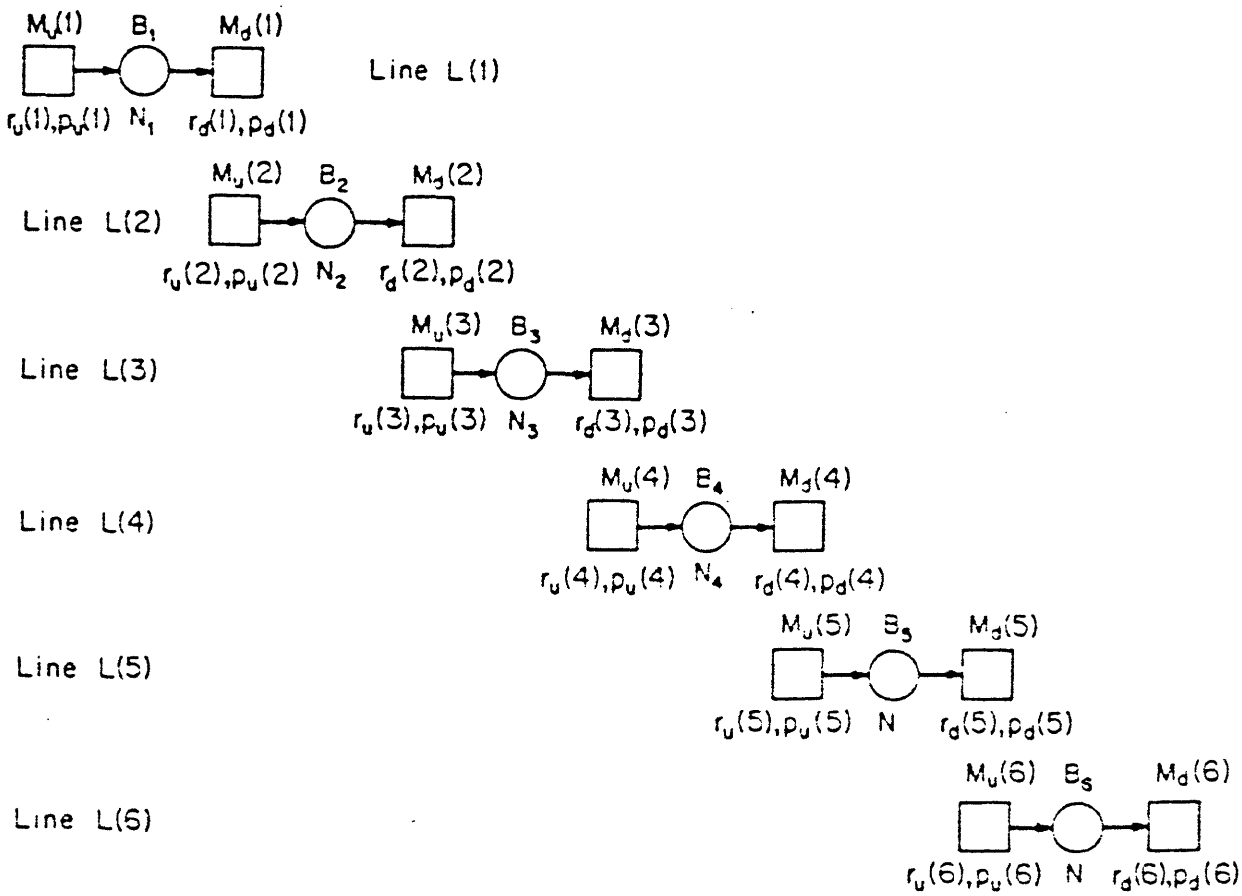


Figure 2

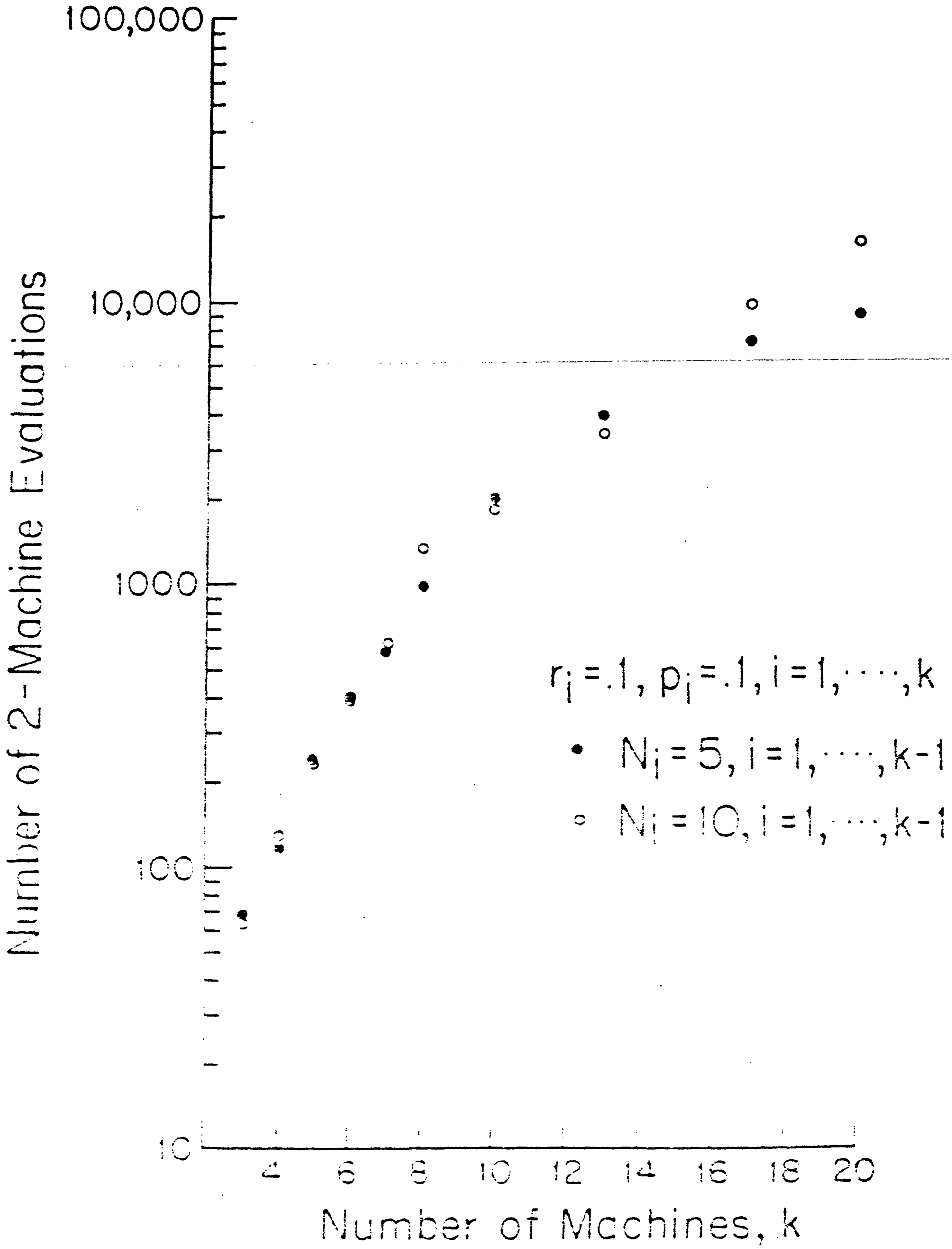

100,000
10,000
1000
100
10
Number of 2-Machine Evaluations
4 6 8 10 12 14 16 18 20
Number of Machines, k
$r_i = .1,\ p_i = .1,\ i = 1, \cdots, k$
• $N_i = 5,\ i = 1, \cdots, k-1$
○ $N_i = 10,\ i = 1, \cdots, k-1$

Figure 3

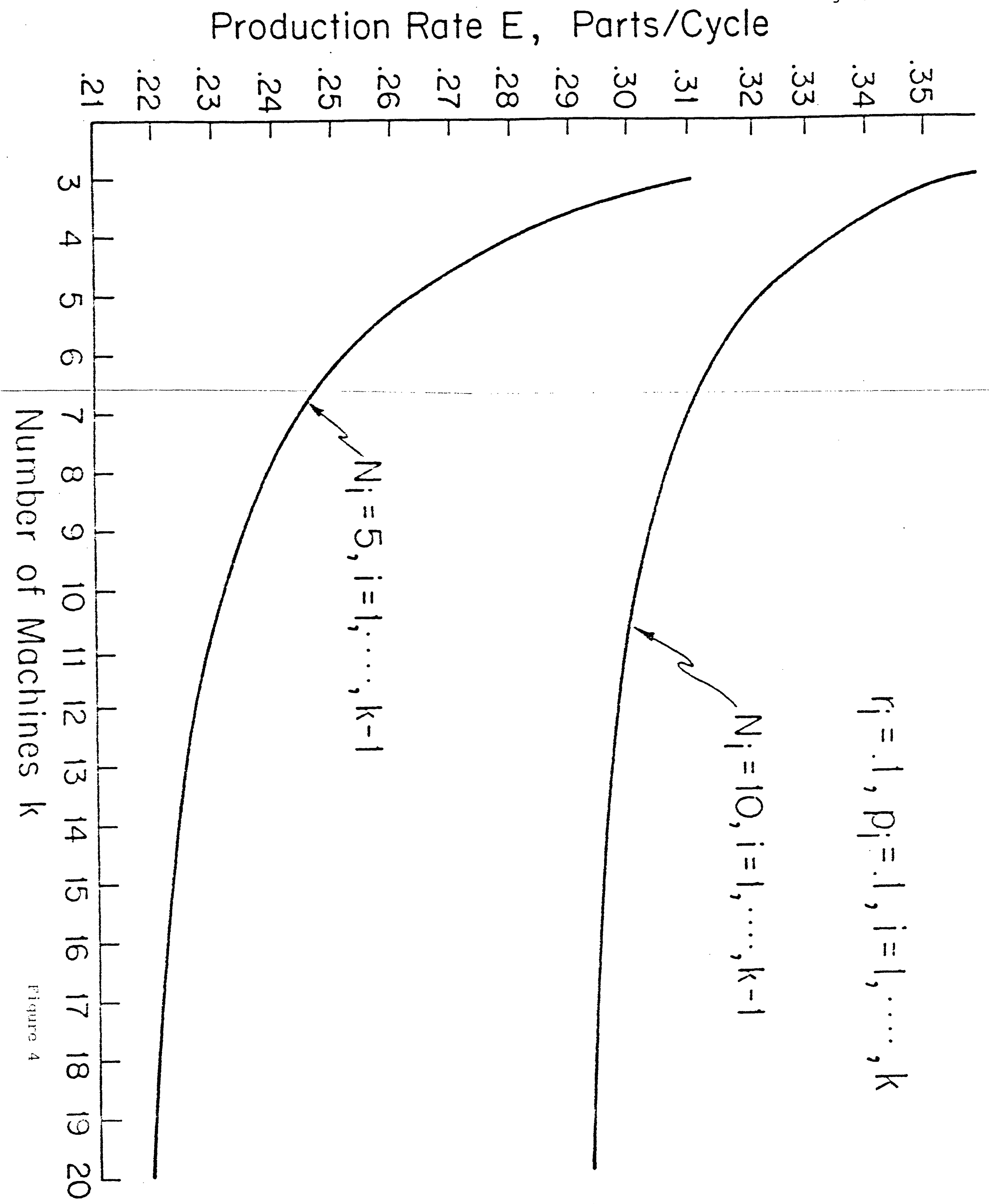
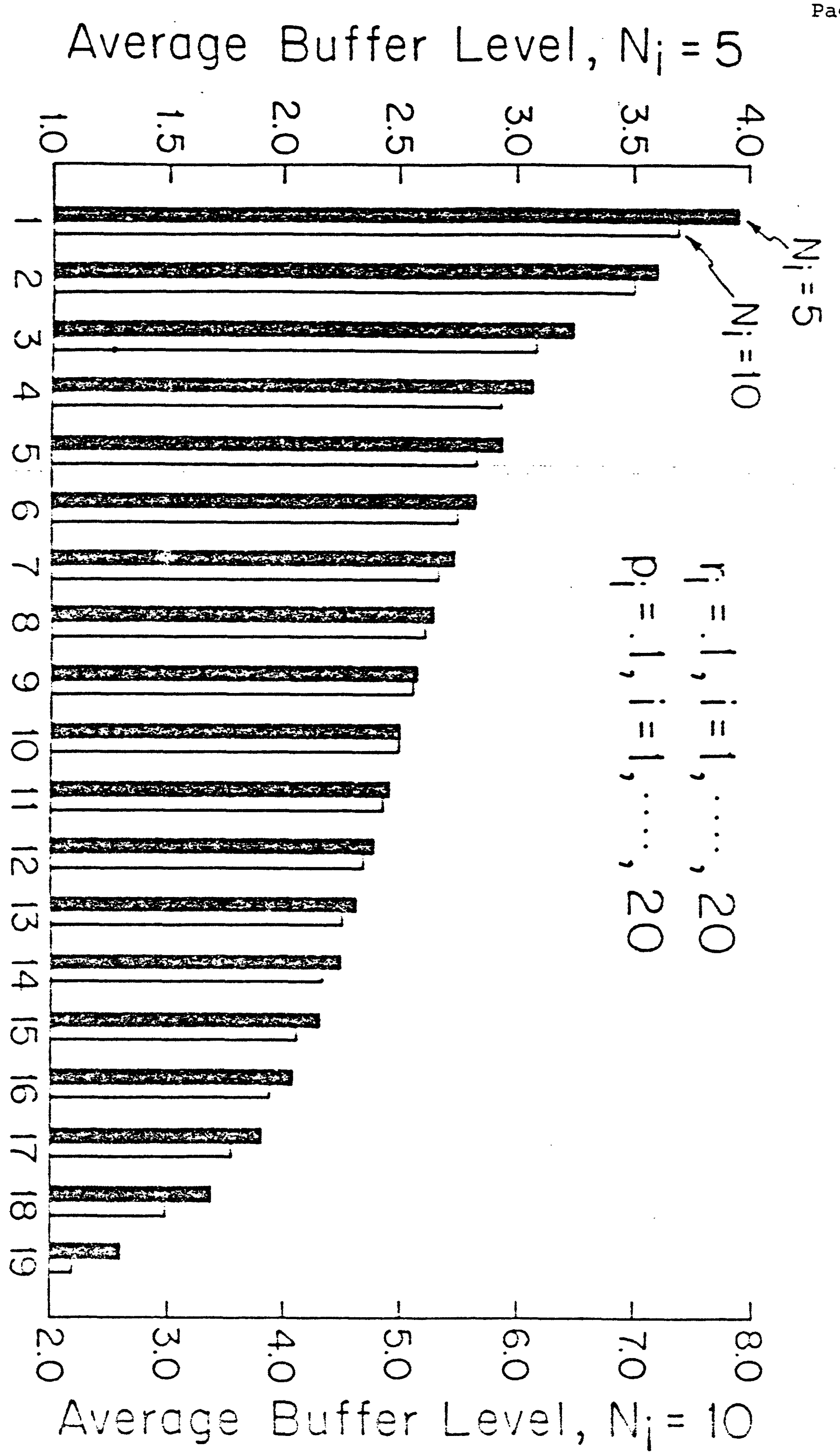

Production Rate E, Parts/Cycle
.21
.22
.23
.24
.25
.26
.27
.28
.29
.30
.31
.32
.33
.34
.35
Number of Machines k
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
$N_i = 5, i = 1, \ldots, k-1$
$N_i = 10, i = 1, \ldots, k-1$
$r_i = .1, p_i = .1, i = 1, \ldots, k$

Figure 4



Average Buffer Level, $N_i = 5$
1.0
1.5
2.0
2.5
3.0
3.5
4.0
$N_i = 5$
$N_i = 10$
$r_i = .1, i = 1, \cdots, 20$
$p_i = .1, i = 1, \cdots, 20$
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Buffer
2.0
3.0
4.0
5.0
6.0
7.0
8.0
Average Buffer Level, $N_i = 10$

Figure 5

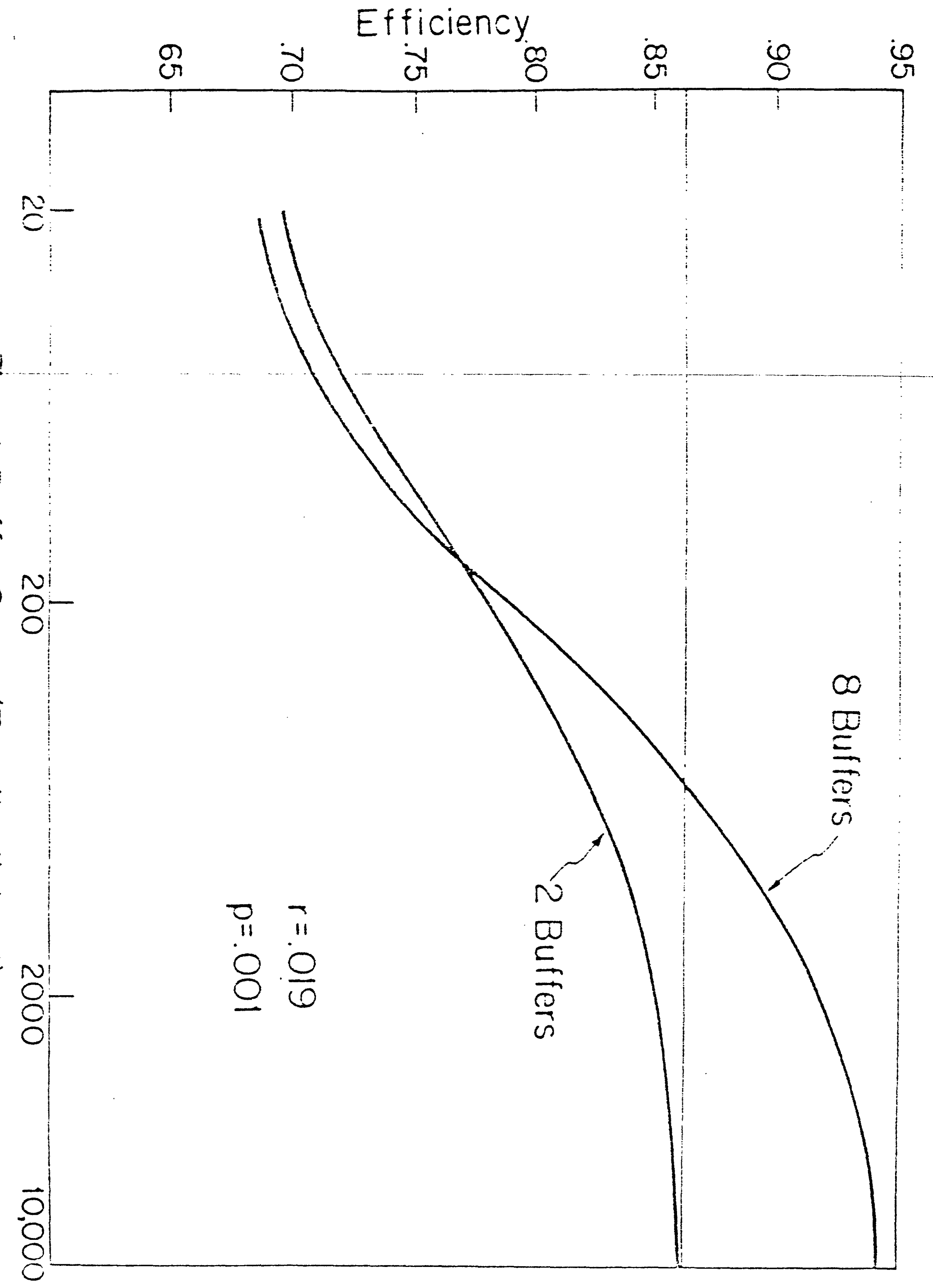

Efficiency
.95
.90
.85
.80
.75
.70
65
20
200
2000
10,000
Total Buffer Space (Equally divided)
8 Buffers
2 Buffers
r=.019
p=.001

Figure 6