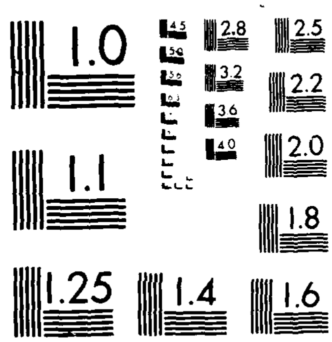
AD-A175 193 TRANSIENT BEHAVIOR OF LARGE MARKOVIAN MULTI-ECHELON 1/1
REPAIRABLE ITEM INVEN. (U) GEORGE WASHINGTON UNIV
WASHINGTON D C DEPT OF OPERATIONS RESE.
UNCLASSIFIED D GROSS ET AL 16 OCT 86 N00014-83-K-0217 F/G 15/5 NL



ROCOPY RESOLUTION TEST CHART

Contract N00014-83-K-0217

(1)

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Department of Operations Research

16 Oct 1986

TRANSIENT BEHAVIOR OF LARGE MARKOVIAN MULTI-ECHELON REPAIRABLE ITEM INVENTORY SYSTEMS USING A TRUNCATED STATE SPACE APPROACH

by

Donald Gross
Leonidas C. Kioussis
Douglas R. Miller

Calculations fo· .arge Markovian finite source, finite repair capacity two-echelon repairable item inventory models are shown to be feasible using the randomization technique and a truncated state space approach. More complex models (involving transportation pipelines, multiple item types and additional echelon levels) are also considered.

1. INTRODUCTION

Let $\{X(t),\ t \geq 0\}$ be a continuous-time time-homogeneous Markov process (CTMP) on a finite state space $S = \{1,2,\ldots,m\}$. All such Markov processes can be characterized by an initial distribution $\underline{\pi}(0)$ and an infinitesimal generator

$$Q = \begin{pmatrix} -q_1 & q_{12} & q_{13} & \cdots & q_{1m} \\ q_{21} & -q_2 & q_{23} & \cdots & q_{2m} \\ q_{31} & q_{32} & -q_3 & \cdots & q_{3m} \\ \vdots & \vdots & \vdots & & \vdots \\ q_{m1} & q_{m2} & q_{m3} & \cdots & -q_m \end{pmatrix}$$

where

$$q_{ij} = \lim_{\Delta t \to 0} \frac{P(X(t+\Delta t)=j \mid X(t)=i)}{\Delta t} , \quad i \neq j$$

and

$$q_i = \sum_{j \neq i} q_{ij} .$$

The q_{ij}'s are the transition rates. The infinitesimal generator Q seems to be the most natural way to describe the stochastic nature of continuous time Markov models with denumerable state spaces. The state probability vector at time t is denoted $\underset{\sim}{\pi}(t) = (\pi_1(t), \pi_2(t), \ldots, \pi_m(t))$, where $\pi_s(t) = P(X(t) = s)$, $s \varepsilon S$. These transient probabilities satisfy the Kolmogorov forward equation

$$\underset{\sim}{\pi}'(t) = \underset{\sim}{\pi}(t)Q \,, \quad t \geqslant 0 \,. \tag{1}$$

This is an initial value system with $\underset{\sim}{\pi}(0)$ given. (See [1,7,14] for more background on Markov processes.) It is apparent that for processes with large state spaces, we need to solve very large systems of differential equations. For example, a two-base, one-depot, single-item type, two-echelon system with 24 units at each base and two spare units at the depot has a state space of 106,875 possible states, requiring the solution of 106,874 simultaneous, linear, first-order differential equations in 106,874 unknowns.

This paper presents a method for solving Equation (1) for certain models with large state spaces: the randomization numerical technique is used to solve a truncated version of Equation (1). This makes it possible to calculate measures of interest for systems which are even too large to be handled by applying the efficient randomization technique to the full state space. This then, would allow the modeling of such systems as an aircraft wing, with three squadrons and two echelons of supply and repair, or a fleet of gas turbine engine ships with both shipboard and shore repair capability.

The paper is structured as follows: The randomization algorithm is reviewed in Section 2. Multi-echelon repairable item systems are discussed in Section 3; the primary goal of this paper is to investigate the transient behavior of such systems, however they also provide good examples for illustrating the general truncation approach. The full state space of a particular multi-echelon system is described in Section 4. The truncated state space is given in Section 5. The problem of determining a good level of truncation is discussed in Section 6. Some numerical results are given in Section 7. More complex models are discussed in Section 8; these include systems with transportation delays, multiple items and higher echelons. Section 9 contains conclusions. This paper extends the earlier work of Gross and Miller [5].

2. THE RANDOMIZATION ALGORITHM

Any Markov process X on a finite state space can be represented as a discrete time Markov chain (the uniformized embedded chain) "randomized" by a Poisson process. Define

$$P = Q/\Lambda + I \ , \quad \text{where} \quad \Lambda = \max_{i \in S} q_i \tag{2}$$

and I is the identity matrix; P is a stochastic matrix. Let $\{Y_n, n = 0,1,2,\ldots\}$ be a Markov chain on S with transition matrix P and initial distribution $\underset{\sim}{\pi}(0)$. Let $\{N(t), t \geq 0\}$ be a Poisson process with rate Λ which is independent of $\{Y_n, n = 0,1,2,\ldots\}$. Then $\{Y_{N(t)}, t \geq 0\}$ is a Markov process with generator Q and initial distribution $\underset{\sim}{\pi}(0)$ and hence is probabilistically identical to $\{X(t), t \geq 0\}$. This construction makes it possible to compute transient probabilities of a Markov process with generator Q from transient prob ilities of a Markov chain Y with transition maxtrix P and a Poisson process N with rate Λ. The transient probabilities of Y are denoted $\underset{\sim}{\phi}(n) = (\phi_1(n), {}_2(n), \ldots, \phi_m(n))$, where $\phi_s(n) = P(Y_n=s)$, $s \in S$. The randomization formula is

$$P(X(t) = s) = \sum_{n=0}^{\infty} P(X(t)=s \mid N(t)=n)P(N(t)=n)$$

$$= \sum_{n=0}^{\infty} P(Y_n = s)P(N(t) = n)$$

or equivalently,

$$\underset{\sim}{\pi}(t) = \sum_{n=0}^{\infty} \underset{\sim}{\phi}(n) \frac{e^{-\Lambda t}(\Lambda t)^n}{n!} . \tag{3}$$

See Gross and Miller [4] for additional discussion and details. (Equation (3) can also be found in Çinlar [1, p. 259] and Keilson [10, Eqn. 2.1.5].)

The infinite series in Equation (3) must be truncated for computational purposes. Let

$$T(\varepsilon,t) = \min \left\{ k: \sum_{n=0}^{k} \frac{e^{-\Lambda t}(\Lambda t)^n}{n!} > 1 - \varepsilon \right\} \tag{4}$$

where ε equals an acceptable error (specified by the user). The computational version of Equation (3) is

$$\underset{\sim}{\pi}^{\varepsilon}(t) = \sum_{n=0}^{T(\varepsilon,t)} \underset{\sim}{\phi}(n) \frac{e^{-\Lambda t}(\Lambda t)^n}{n!} . \tag{5}$$

Truncation of the infinite series involves a probability loss of at most ε; thus all probabilities (of states or subsets of states) will have an error between $-\varepsilon$ and 0. Note that the randomization formula (5) reduces the calculation of transient probabilities of a Markov process to those of a Markov chain and underlying Poisson process, both of which are more amenable to exact numerical evaluation.

The $\underset{\sim}{\phi}$'s are computed recursively using the relation from standard Markov chain theory;

$$\underset{\sim}{\phi}(0) = \underset{\sim}{\pi}(0) \; ; \quad \underset{\sim}{\phi}(n+1) = \underset{\sim}{\phi}(n)P \; , \quad n \geq 0 \; . \tag{6}$$

(Note that Equation (6) involves only nonnegative numbers, a fact that contributes to numerical stability of the algorithm.) The matrix P is usually sparse and thus the above matrix multiplication should be performed by an appropriate algorithm. Such a multiplication algorithm (called *SERT*) is described by Gross and Miller [4]. The number of operations in this algorithm is proportional to the sum of the number of states and the number of transitions.

In short, the standard randomization computational algorithm computes Λ and P from the generator Q using (2). It computes the truncation point $T(\varepsilon,t)$ from (4), then the $\underset{\sim}{\phi}(n)$'s using (6) recursively, accumulating in Equation (5) to give $\underset{\sim}{\pi}^{\varepsilon}(t)$.

Gross and Miller [5] have computed transient probabilities for Markov process models of multi-echelon inventory systems with 20,000 states and 200,000 transitions using the randomization algorithm. Melamed and Yadin [11] have applied the method to queuing networks with a large number of states. Miller [12] has adapted the randomization algorithm to efficiently handle certain stiff systems which arise in the reliability analysis of fault-tolerant systems.

There are other numerical approaches to the solution of the Kolmogorov equation (1) which we will not consider. Two general approaches are: (i) numerical integration techniques such as Runge-Kutta, predictor-corrector, etc.; and (ii) exponentiation $[\underset{\sim}{\pi}(t) = \underset{\sim}{\pi}(0)e^{Qt}]$ by computing the spectrum, computing the Taylor series, or other means. The randomization technique has a distinct advantage over these approaches in that a bound on the global error can be set by the user, and it is achieved. (The only other source of error is the influence of rounding and truncations by the machine performing the calculations; by noting that the randomization algorithm mainly involves multiplication and addition of positive numbers, Grassmann [3] has bounded this error.) Furthermore, Grassmann [2] has shown empirically that randomization is more efficient for some queuing systems.

3. MULTI-ECHELON REPAIRABLE ITEM SYSTEMS

Multi-echelon repairable item provisioning systems are generalizations of the classic machine repair model. We consider a system consisting of two bases and a depot. Each base has a certain number of "machines" (or key replaceable components of "machines") assigned to it and a certain desired number of these which should be operating. Machines fail (independently of each other) after being operated for an exponentially distributed length of time. There are repair shops at each base and at the depot. When a machine fails it has a certain probability of going to the base repair shop for repair; otherwise it goes to the depot repair shop. Each repair shop has a certain number of repair channels. Repair times are exponentially distributed. If there are more machines requiring repair than repair channels at a given repair shop, a queue forms. The depot repair shop stocks spare machines which are used on a one-to-one ordering basis to replace failed machines coming from the bases. If the depot spares pool is empty when a replacement is required, a backorder is created which will be filled when repair is completed on one of the items in depot repair. If both bases are awaiting backorders, a repaired item is sent to the base with the maximum depot backorders (ties are broken by flipping a fair coin). When neither base is awaiting a backorder, a machine completing depot repair is placed in the depot spares pool. Thus any given machine may be in any of six states (or equivalently at any of six nodes in a "network"): failed and in base repair shop at either base (BR1,BR2); failed and in depot repair shop (DR); operational and at either base (BU1,BU2); operational and in the depot spares pool (DU). These six states and the possible transitions a machine can make between them are illustrated in Figure 1. The parameters and variables of the system are described in Table 1. Note that transportation times from bases to base repair shops and from bases to depot repair shop are assumed to be negligible. We consider adding transportation nodes to the network in Section 8.

We shall introduce the following classification symbology for these systems: (#bases, #levels of repair, #levels of supply). So the system of Figure 1 is considered a (2,2,2) system, since there are two bases, repair facilities at both base and depot levels, and spares stockage at both base and depot levels.

Steady-state models and behavior of multi-echelon inventory systems are presented by Sherbrooke [15] and Muckstadt [13]. In these models, they assume an "infinite" population of machines, so that the system failure rate is constant, regardless of the number of machines actually in operation (state-independent failure rate). Further, they also assume that "ample" repair facilities exist so that failed items

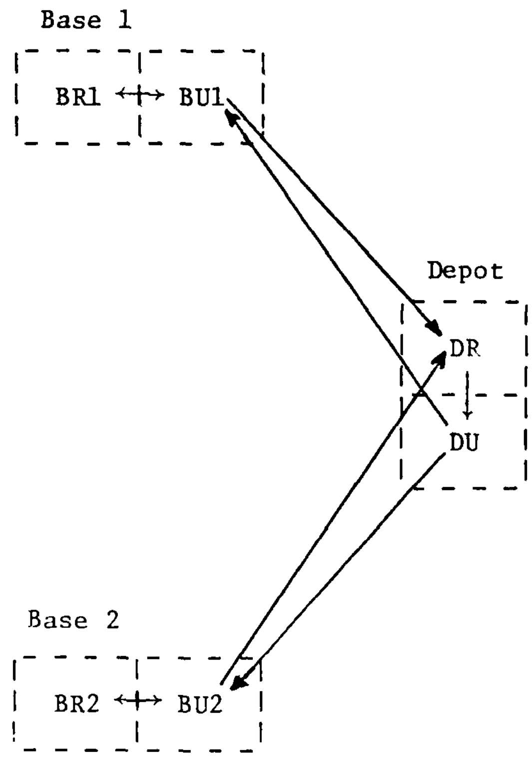


Figure 1. General schematic for a two-base multi-echelon repairable item system.

never queue in the repair shops but go immediately into service. A method for computing approximate transient performance measures of the above multi-echelon system is presented by Hillestad and Hillestad and Carillo [8,9], again with the limiting assumptions of state-independent failure rate and ample service. These two assumptions make it considerably easier to obtain both steady-state and transient results. Gross and Miller [5] compute exact transient probabilities using the randomization algorithm, explicitly accounting for state dependent failure rates and queueing at the repair facilities. For further background on multi-echelon inventory systems, see the above references.

TABLE 1

PARAMETERS AND VARIABLES OF MULTI-ECHELON SYSTEM

Symbol	Definition
	a. Parameters
BUi	Denotes network node: Base i operational (working and spares) units
BRi	Denotes network node: Base i repair facility
DU	Denotes network node: depot spares
DR	Denotes network node: depot repair facility
BSi	Allocation of total stock to Base i (operating machines plus spares), i = 1,2
MSi	Desired number of working machines at Base i
BCi	Number of repair channels in repair shop at Base i
α_i	Probability machine failing at Base i is base repairable
λ_i	Mean failure rate, Base i items
μ_i	Mean repair rate, Base i items
DS	Number of depot spares
DC	Number of depot repair channels
μ_D	Mean depot repair rate
	b. Variables
#BUi	Number of operational units currently at Base i
#BRi	Number of units currently in or awaiting Base i repair
#DU	Number of spares currently available at depot
#DR	Number of units currently in or awaiting depot repair
#DBi	Number of depot backorders from Base i

4. STATE SPACE DESCRIPTIONS OF (2,2,2) SYSTEMS

Let us define the state of the system as the number of machines at each node:

$$\underset{\sim}{s} = (\#BU1,\#BR1,\#BU2,\#BR2,\#DR,\#DU) \ .$$

The number of backorders at the depot from Bases 1 and 2, respectively. are:

$$\#DB1 = BS1 - (\#BU1 + \#BR1) \ ,$$

$$\#DB2 = BS2 - (\#BU2 + \#BR2) \ ,$$

where, as given in Table 1, BS1 and BS2 are the allocation of total stock to Bases 1 and 2, respectively.

In general, the state space appears to have six dimensions, but because of one-for-one ordering and conservation of the total number of items in the system, the dimensionality of the state space can actually be reduced.

The description of the state space breaks into two situations: (i) no depot spares available, and (ii) some depot spares available. Thus we break S into two parts, namely

$$S = S_0 \cup S_+$$

where

S_0 = states with depleted depot spares pool

S_+ = states with nondepleted depot spares pool.

First consider S_0 . The state of the system can be described with four numbers:

$$(\#DB1,\#BR1,\#DB2,\#BR2) \ ,$$

since here $\#DU = 0$, and thus

$$\#BU1 = BS1 - \#DB1 - \#BR1$$

$$\#BU2 = BS2 - \#DB2 - \#BR2$$

$$\#DR = DS + \#DB1 + \#DB2 \ .$$

The feasible states of S_0 are subject to two constraints:

$$\#DB1 + \#BR1 \leqslant BS1$$

$$\#DB2 + \#BR2 \leqslant BS2 \ .$$

Therefore S_0 can be represented as a Cartesian product,

$$S_0 = T_1 \times T_2 \ ,$$

where

$$T_1 = \{(\#DB1, \#BR1): \ \#DB1 + \#BR1 \leqslant BS1\}$$

$$T_2 \quad \{(\#DB2, \#BR2): \ \#DB2 + \#BR2 \leqslant BS2\} \ .$$

Figure 2 shows a schematic of the set of states of S_0 . The notation T is used because the spaces T_1 and T_2 are triangular. The number of points in T_1 and T_2 are

$$|T_1| = \frac{(BS1+1)(BS1+2)}{2}$$

and

$$|T_2| = \frac{(BS2+1)(BS2+2)}{2} \ ,$$

respectively, and the number of states in S_0 is

$$|S_0| = \frac{(BS1+1)(BS1+2)(BS2+1)(BS2+2)}{4} \ .$$

Now, let us consider the states where the spares pool at the depot is not empty, S_+ . In this case the state of the system can be described by three numbers:

$$(\#BR1, \#BR2, \#DU) \ ,$$

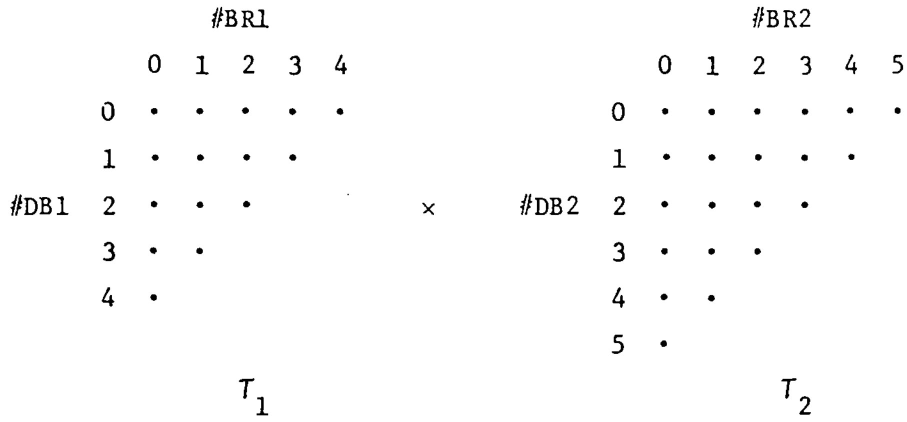


Figure 2. An example of state space S_0 describing individual bases; here BS1 = 4 and BS2 = 5.

with constraints on these given by

$$\#BR1 \leqslant BS1$$

$$\#BR2 \leqslant BS2$$

$$0 \leqslant \#DU \leqslant DS \ .$$

We can condition on the value of #DU to get S_+ as follows.

Let

$$S_+ = S_1 \cup S_2 \cup \ldots \cup S_{DS} \ ,$$

where S_i consists of states with exactly i machines in the depot spares pool. Note that each S_i is a rectangle and its size is

$$|S_i| = (BS1+1)(BS2+1) \ .$$

Thus

$$|S_+| = (BS1+1)(BS2+1)DS$$

and the total number of states is

$$|S| = \frac{(BS1+1)(BS1+2)(BS2+1)(BS2+2)}{4} + (BS1+1)(BS2+1)DS \ . \qquad (7)$$

So, we can describe the entire state space as

$$S = T_1 \times T_2 \bigcup \left(\bigcup_{i=1}^{DS} S_i \right) .$$

Examples of state space sizes are given in Table 2.

In order to compute the probability distribution for all the states at any continuous time point, $p_{\underset{\sim}{s}}(t)$ $(\underset{\sim}{s} \in S \ , \ t \in [0,\infty))$, we use the randomization method to provide the computational formulas and a technique called *SERT* (see [4]) which takes advantage of a sparse Q matrix as computational machinery. There are two ways to compute the probability distribution for the CTMP when using the *SERT* technique:

(i) Table look-up

(ii) Algorithmic approach.

When using the table look-up, we construct all the target and rate vectors and store them. Any time that an event occurs, the algorithm goes through these vectors and gets the necessary information to compute the next discrete time probability vector. This procedure has the undesirable feature that a huge amount of the main memory is needed (in fact the memory needed is approximately twice the product of the number of different event types and the size of the state space). Consequently, we are limited in the size of problem that we are able to run.

When using the algorithmic approach, we calculate the transition rates and the target states each time an event of the underlying Poisson process occurs. While this algorithmic approach does not consume the same vast amounts of memory space as the table look-up method, the algorithmic approach becomes quite uneconomical with respect to CPU time.

TABLE 2

SIZE OF THE STATE SPACES OF (2,2,2) SYSTEMS
FOR SELECTED VALUES OF BS1, BS2, AND DS

BS1	BS2	DS	$\lvert S_0 \rvert$	+ $\lvert S_+ \rvert$	= $\lvert S \rvert$
2	2	2	36	18	54
4	4	2	225	50	275
6	6	2	784	98	882
8	8	2	2025	162	2187
10	10	2	4356	242	4598
12	12	2	8281	338	8619
18	18	2	36100	722	36822
24	24	2	105625	1250	106875

It becomes clear that for this class of problem (Markovian with large state-space), if we want to get exact solutions (within a prespecified error tolerance) using the existing tools, we confront either the computer's main memory restrictions, or the high cost of CPU time.

The question that naturally comes up is: Can we find another way to estimate (within a prespecified error tolerance) measures of interest such as the availability of a desired number of machines at time t ? It would be reasonable to expect that, for a system that initially starts at full strength (all machines operational), the probability will be concentrated among only a relatively small part of the whole state space during a mission period (or period of interest), if the traffic intensities are low and the mission periods are relatively short. In other words, we should be able to truncate the state space and consider only those states among which we believe almost all the probability to be distributed.

The truncated states, on the other hand, are "lumped" into one or more states, which can be treated as absorbing states. If the probability of visiting any of those truncated states (probability of being absorbed) is negligible during the mission period, then the analytical measures estimated are almost exact. Further, we will know the additional error introduced in our measures, as this will be the absorption probability. However, there are two major questions that must be addressed: How will we truncate the state space? In which cases will the truncation procedure perform satisfactorily?

5. TRUNCATED STATE-SPACE APPROACH

Figure 3 shows a schematic of the state space representation for a (2,2,2) system. Recall that

$$S = T_1 \times T_2 \cup \left(\bigcup_{i=1}^{DS} S_i \right) = S_0 \cup S_+ \ .$$

Assume that we truncate the state space as shown in Table 3. Then the truncated state space will look like that shown in Figure 4. The truncated

TABLE 3

TRUNCATION OF STATE SPACE

For Variable	In Portion of State Space	Truncated Beyond
#DB1	T_1	TDB1
#BR1	T_1	TBR1
#DB2	T_2	TDB2
#BR2	T_2	TBR2
#BR1	S_+	TBR1'
#BR2	S_+	TBR2'

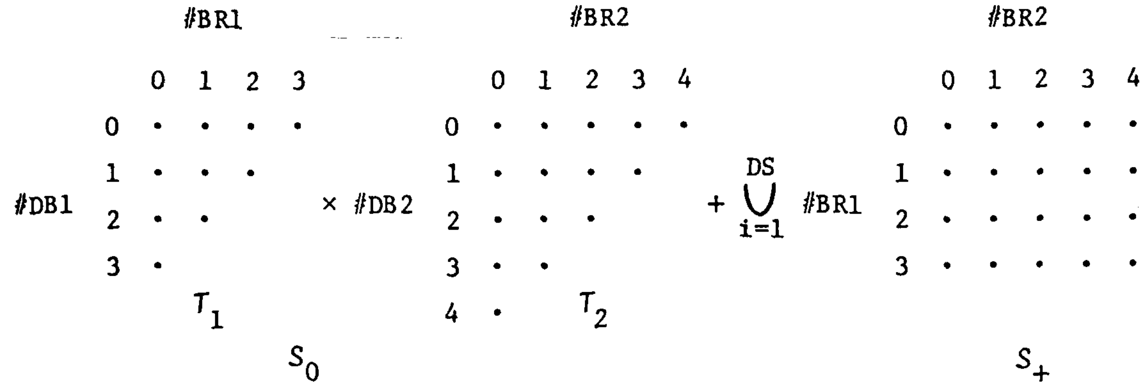


Figure 3. Full state-space representation for a problem where BS1 = 3 and BS2 = 4.

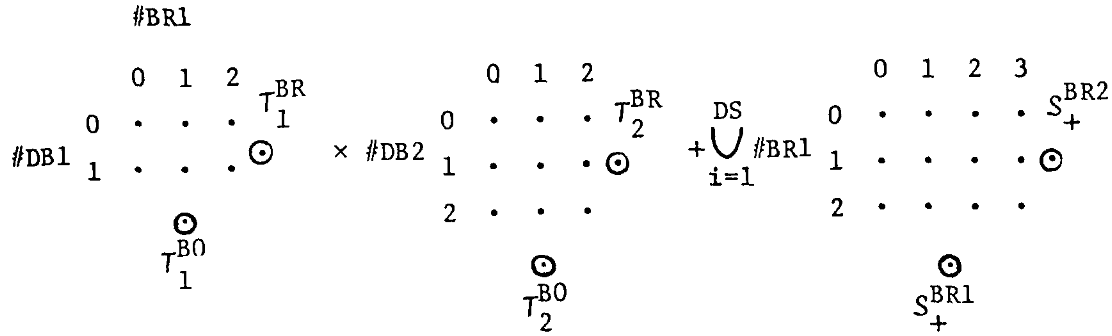


Figure 4. Truncated state space, where TDB1 = 1, TBR1 = 2, TDB2 = 2, TBR2 = 2, TBR1' = 2, TBR2' = 3. The states T_1^{BR}, T_1^{BO}, T_2^{BR}, T_2^{BO}, S_+^{BR2}, S_+^{BR1} are absorbing states.

states are lumped into six absorbing states. Whenever there is a transition from the remaining state space into a truncated state, it will be considered as a transition to a specific (absorbing) state which, once visited by the process, will never be left. The size of the state space (excluding the absorbing states) is now reduced to

$$S = (\text{TDB1}+1)\cdot(\text{TBR1}+1)\cdot(\text{TDB2}+1)\cdot(\text{TBR2}+1) + \text{DS}\cdot(\text{TBR1}'+1)\cdot(\text{TBR2}'+1) \ .$$

The benefits of truncating the state space are readily apparent. First, we decrease significantly the amount of computer memory required (e.g., we need to consider only a fraction of the whole state space). Secondly, by this method we are able to calculate the total absorbed probability or, in other words, the amount of total error added because of absorption. Thirdly, CPU time for the execution of the algorithm is reduced because it depends approximately linearly on the size of the state space. This approach can also be utilized when treating certain infinite state-space CTMP's, for example $M/M/c/\infty$ queues or open queuing networks.

6. ESTIMATION OF INITIAL TRUNCATION POINTS

Now that the algorithm is established, the only step needed before it can be implemented is the selection of the truncation points. To do this, we look at our (2,2,2) system as three independent $M/M/1/\infty$ queues, the traffic intensities of which are made equal to the traffic intensities of the three stations of our original problem. We define traffic intensities for our state-dependent original problem as the maximum possible load, namely, $\lambda_i MS_i \alpha_i / \mu_i BC_i$ for the bases and $\sum_i \lambda_i MS_i (1-\alpha_i)/\mu_D DC$ for the depot. We mention here that three independent queues are not in general a good approximation for our true multi-echelon queueing network, but will serve our purpose for providing reasonable values for initial truncation points, and is used only for this purpose. Once the state space is truncated, randomization *SERT* is applied to the original multi-echelon network. The question that now arises is: What is the smallest n, such that

$$\Pr\{\text{an } M/M/1/\infty \text{ queue visits state } n \text{ or higher during } [0,t_m]\} \leq \delta ,$$

where t_m is the mission time. The n estimated this way will qualify as the appropriate initial truncation point. As a side observation,

$$\Pr\{\text{an } M/M/1/\infty \text{ visits } n \text{ or higher during } [0,t_m]\}$$
$$= \Pr\{\text{an } M/M/1/n \text{ visits } n \text{ during } [0,t_m]\} .$$

Using randomization *SERT*, we calculate the absorbing probabilities as functions of time and truncation point for different traffic intensities, ranging from .2 up to .7 (see the appendix). It turns out that as the truncation point n increases, the absorbing probability as a function of time becomes linear, with slope equal to 1 (on a ℓn-ℓn scale) and intercept dependent on the traffic intensity and the truncation point n .

The graphs calculated for M/M/1/n queues as given in the appendix are based on a randomization cut-off error, ε , of 10^{-6} . These can be used to satisfactorily estimate truncation points even for systems with more than one server, provided that the "match" of traffic intensities is maintained, that is, the single-server service rate is set equal to the sum of the multiple servers service rates. If the traffic intensities are relatively low (which they often are in multi-echelon repairable inventory systems), and the mission period short (t_m small), then the truncation technique should work in the sense that the truncation points decrease the number of states to be considered to a dimension that the computer memory can handle (in Section 9, we discuss this further). Otherwise, system approximation techniques such as decomposition into independent queues or approximation by Jackson networks, or simulation. Gross, Miller, and Plastiras [6] give a comparison of randomization and simulation, including CPU times for comparable accuracies.

Their general conclusion was that for smaller state spaces and for higher precision, randomization is more economical. For example, to obtain a precision of ±1% with 99% confidence using simulation turned out to require three times as much CPU time as randomization for a 15000-state problem.

7. EXAMPLES

We present two examples, one of a (2,2,2) system and another of a (3,2,2) system, which show the effectiveness of the truncation approach.

Example 1.

Let us consider the (2,2,2) system shown in Figure 5. The full state space for this problem is 77,234 states. Using the table look-up algorithm, it is impossible to handle this problem on the VAX 11/780 system without truncation. We desire availability measures, where

$$\text{Availability at time } t \text{ for Base } i \equiv A_i(t) = \sum_{j=MS_1}^{BS_1} p_{j,i}(t) ,$$

where $p_{j,i}(t)$ is the probability that j machines are working at Base i at time t. We also desire the total probability of absorption during the mission time t_m (which for this example equals 30) to be less than .015.

To evaluate the initial truncation points, for Base 1 the traffic intensity ρ_1 is

$$\rho_1 = \frac{\alpha_1 \lambda_1 MS1}{BC1\mu_1} = \frac{.6(.15)20}{4.5} = 0.4 .$$

When we calculate the graphs of the absorbing probabilities for different cases of traffic intensity ρ, we assume $\lambda = \rho$ and $\mu = 1$ (we refer to $\mu = 1$ as the "nominal" value). In general, μ is not 1 and we must adjust our time scale accordingly to match event rates. The absorbing probabilities at time t for an M/M/1 system with traffic intensity $\rho = k\lambda/k\mu$ is the same as the absorbing probability at time kt for a

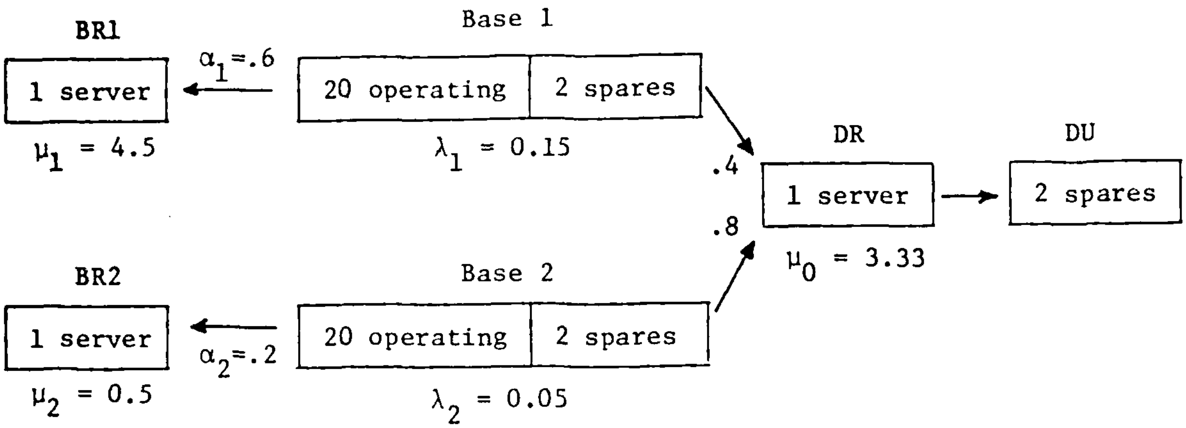


Figure 5. (2,2,2) example.

system with $\rho = \lambda/\mu$, since these two systems yield the same expected number of events over their respective time intervals, namely, $t(k\lambda + k\mu) = (kt)(\lambda + \mu)$. Thus, while we are interested in a mission time of 30, our μ is 4.5, not 1, so $k = 4.5$ and the "equivalent" mission time is $4.5(30) = 135$. Using our graphs in the appendix, given that $\rho = 0.4$, we wish to find the smallest n , such that Pr{*an M/M/1/n visits n in time period* [0,135]} $\leqslant .005$. We obtain an n of 9, so that TBR1 = 9 , and TBR1' = 9 will serve as our initial truncation points for the state variable #BR1 for the S_0 and S_+ portions of the state space, respectively.

Similarly, at Base 2 the traffic intensity ρ_2 is:

$$\rho_2 = \frac{\alpha_2 \lambda_2 MS2}{BC2\mu_2} = \frac{.2(.05)20}{.5} = 0.4 .$$

Now here the event rate is 1/2 × nominal (since $\mu_2 = .5$); therefore, we look at the graphs and tables where $\rho = .4$ and seek the smallest n such that Pr{an M/M/1/n visits n in time period [0,15]} $\leqslant .005$. We obtain $n = 6$; thus TBR2 = 6 , and TBR2' = 6 will serve as our initial truncation points for the state variable #BR2 for the S_0 and S_+ portions of the state space, respectively.

We now consider the depot. The traffic intensity ρ_D is

$$\rho_D = \frac{(1-\alpha_1)\lambda_1 MS1 + (1-\alpha_2)\lambda_2 MS2}{DC\mu_D} = \frac{.4(.15)20 + .8(.05)20}{3.333} = 0.6 \ .$$

Here, the rates are 3.333 × nominal (because $\mu = 3.333$) ; therefore, we look at the tables and graphs where $\rho = .6$, and pick the smallest n such that

$$\Pr\{\text{an } M/M/1/n \text{ visits } n \text{ in time period } [0,100]\} \leqslant .005 \ .$$

We obtain $n = 14$, which serves as the truncation point for the number of machines undergoing repair in the repair facility at the depot. Our aim is to fix truncation points for the number of machines that the depot owes to each base. Given that DS = 2 , the total number of machines owed by the depot to the bases is 14 − 2 = 12 . Now, we need to divide this number into two numbers, which will serve as the truncation points for the depot backorders in Base 1 and Base 2, respectively. One way to "allocate" is to divide the total according to the ratio of the failure rates of the bases to the depot. In our example we divide the 12 machines as follows. The failure rate to depot from Base 1 is .4 × 20 × .15 = 1.2 and from Base 2 is .8 × 20 × .05 = 0.8 . The total is 2.0 and the allocation then is

$$TDB1 = (1.2/2)(12) \doteq 7 \quad \text{and} \quad TDB2 = (0.8/2)(12) \doteq 5 \ .$$

The truncated state space now looks like that shown in Figure 6.

In Table 4 we can see the number of states considered for the truncated state space, and their respective absorbing probabilities. The entries in the first row correspond for the case where the truncation points are the same as in Figure 6. After examining the absorbing probabilities we find the probability of T_2^{BO} relatively

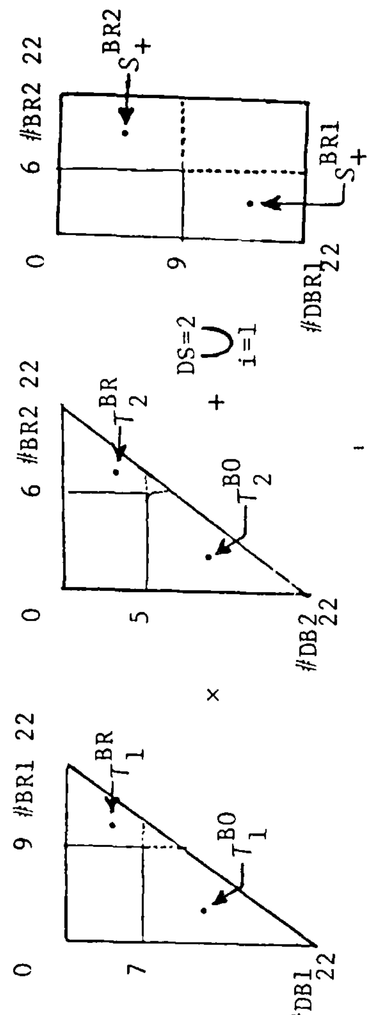


Figure 6. The truncated state space for Example 1.

TABLE 4

Number of States	Total Absorbed Probability	Absorbing States					
		T_1^{BR}	T_1^{BO}	T_2^{BR}	T_2^{BO}	S_+^{BR1}	S_+^{BR2}
3506	.016738	.000253	.002529	.000901	.010140	.000784	.002131
4066	.009190	.000254	.002837	.000903	.002276	.000785	.002135
4066	.010344	.000363	.002572	.001249	.002071	.001122	.002967
4066	.012556	.000558	.002281	.001804	.001859	.001733	.004321
4086	.010906	.000559	.002282	.003896	.001860	.001735	.000573

high. Since we would like the total absorbed probability to be less than .01, we increase TDB2 from 5 to 6, and rerun the algorithm with the results shown in the second row of Table 4.

One might ask whether the $M/M/1/\infty$ approximation can help in estimating truncation points for problems where there is more than one server in the repair shop. To show that it can, Rows 3 and 4 in Table 4 correspond to two additional runs in which we retain the truncation points of the Row 2 example based on the $M/M/1/\infty$, but where we try two and then three service channels at each base repair facility, reducing the service rate of each of the multiple channels appropriately to yield an equivalent service rate. Row 5 lists results of an additional run where the number of service channels at Base 1 and Base 2 repair are 3 (the Row 4 case), but where TBR2' has been increased to 7 in order to reduce the absorption error closer to the desired .01 value.

Example 2.

We now consider a (3,2,2) system as shown in Figure 7. The full state space size can be found by generalizing the formula for the two base case. Here we have the Cartesian product of three triangles (one corresponding to each base) for the S_0 part of the state space, and the union of DS cubes for the S_+ part of the state space. Hence,

$$|S| = (BS1+1) \times (BS1+2) \times (BS2+1) \times (BS2+2) \times (BS3+1) \times (BS3+2)/8$$
$$+ DS \times (BS1+1) \times (BS2+1) \times (BS3+1) .$$

Using the above formula for this example, the full state space is 43,278,703 states. Going through the same procedure used for the previous example, we set our initial truncation points using the graphs given in the appendix and obtain values shown in Figure 8.

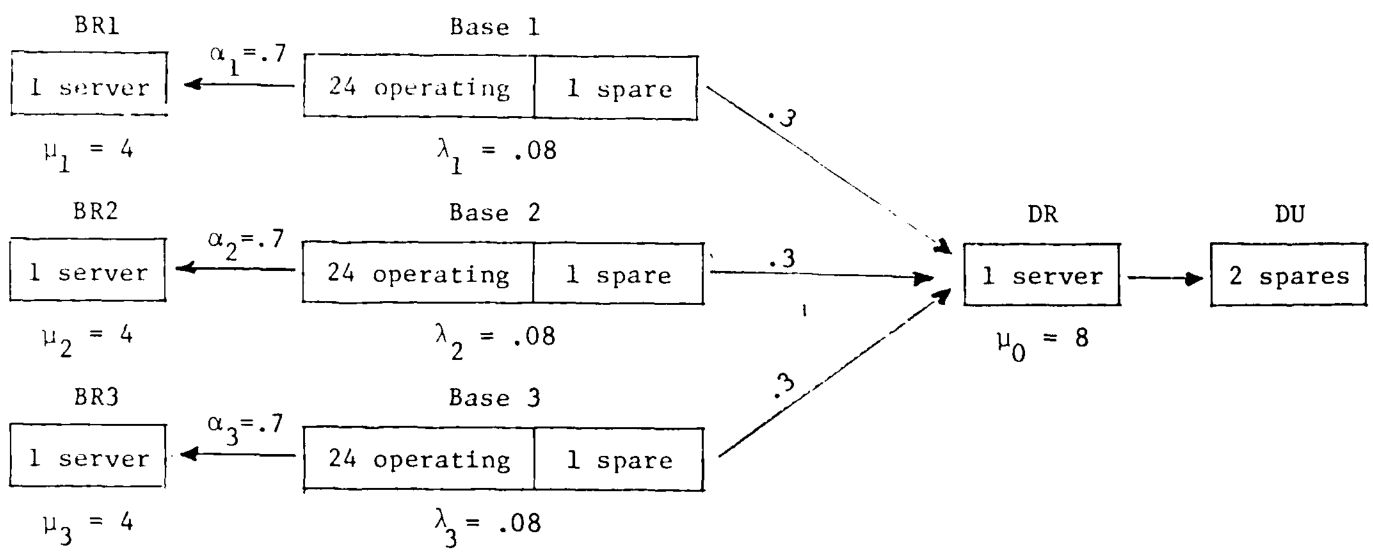


Figure 7. (3,2,2) example.

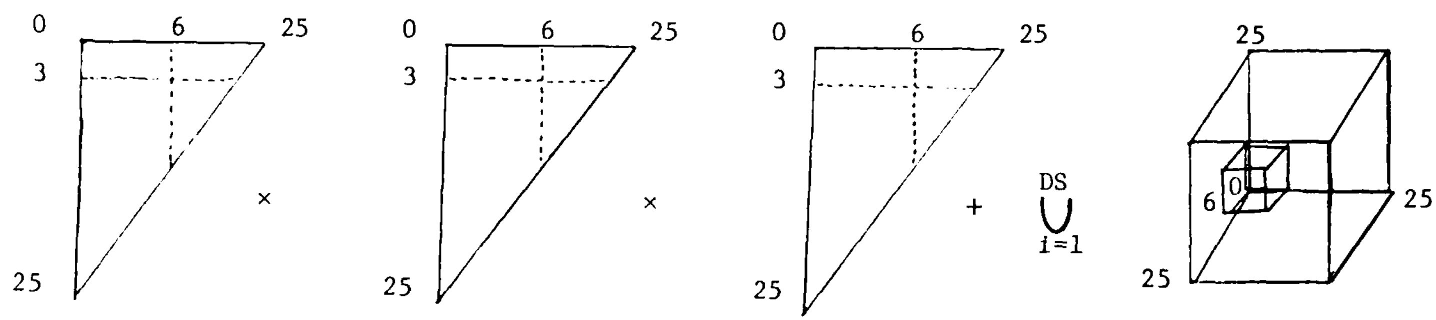


Figure 8. The truncated state space and the truncation points for Example 2.

Table 5 lists the number of states of the truncated state space considered, and the absorbing probabilities. The first row exhibits results based on the truncation points chosen initially (as shown in Figure 8). Observe that S_{+}^{BR1} , S_{+}^{BR2} and S_{+}^{BR3} have the higher absorbing probabilities. After increasing TBR1' , TBR2' and TBR3' from 6 to 8, we reran the problem with the results shown in Row 2 of Table 5 and the total absorbed probability drops below .01.

8. MORE COMPLEX SYSTEMS

We now consider the following three models, which are more complex variations of the models having two bases and two echelons:

(i) Model with transportation pipelines,

(ii) Model with two types of spares,

(iii) Model with intermediate repair shop (three echelons).

Model with Transportation Pipelines

A schematic of such a system is shown in Figure 9. This model has two bases, two levels of supply, and two levels of repair, and serves as a prototype for analyzing transportation pipelines. There is only one additional characteristic that distinguishes the model of Figure 9 from the (2,2,2) system previously studied. Failures from base to the depot enter a single pipeline (denoted by PIN), and repairs of backordered machines enter either one of two pipelines (denoted by P1OUT and P2OUT) depending on to which base they are sent. In order to describe the state space for this model, we will condition on the number of spares available at the depot:

TABLE 5

	Number of States	Total Absorbed Probability	Absorbed probability distributed among regions								
			T_1^{BR}	T_1^{BO}	T_2^{BR}	T_2^{BO}	T_3^{BR}	T_3^{BO}	S_+^{BR1}	S_+^{BR2}	T_+^{BR3}
(1)	22638	.040353	.000540	.000247	.000540	.000247	.000840	.000247	.012664	.012664	.012664
(2)	23410	.007845	.001681	.000250	.001681	.000250	.001681	.000250	.000684	.000684	.000684

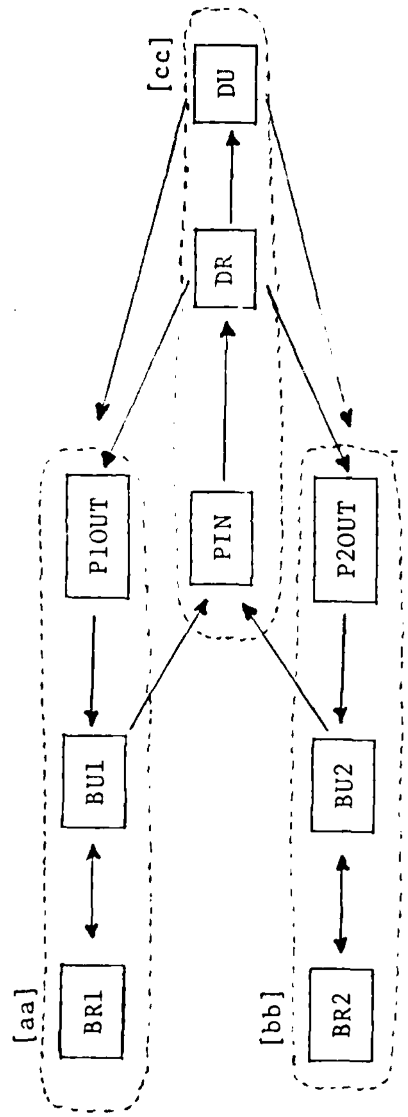


Figure 9. Schematic of pipeline system.

(i) Spares available at depot $\Rightarrow$ DU $>$ 0 .

If spares are available in the depot, then whenever a machine enters the PIN pipeline, another machine enters either the P1OUT or the P2OUT pipeline at the same time. The number of backorders for each base will be at level 0 , so that the total number of machines allocated to each base (BSi) will be distributed in the [aa] or [bb] portions of the network. The conditions for #DU $>$ 0 are

$$\#BR1 + \#BU2 + \#P1OUT = BS1$$

$$\#BR2 + \#BU2 + \#P2OUT = BS2$$

$$\#PIN + \#DR + \#DU = DS ,$$

and the total number of states is:

$$\frac{DS(DS+1)}{2} \times \frac{(BS1+1)(BS1+2)}{2} \times \frac{(BS2+1)(BS2+2)}{2} .$$

(ii) Spares not available in depot $\Rightarrow$ #DU = 0.

For this case,

$$DS \leqslant \#PIN + \#DR \leqslant BS1 + BS2 + DS ,$$

in fact, #PIN + #DR = #DB1 + #DB2 + DS (where #DB1, #DB2 are the numbers of backorders at the depot from Bases 1 and 2, respectively). If we let

$$j = BS1 - \#DB1$$

$$k = BS2 - \#DB2 ,$$

then the total number of states is:

$$\sum_{j=0}^{BS1} \left\{ \frac{(j+1)(j+2)}{2} \times \sum_{k=0}^{BS2} \left\{ \frac{(k+1)(k+2)}{2} \times [DS+(BS1-j)+(BS2-k)+1] \right\} \right\}$$

Thus, the total state space size is given by:

$$|S| = \frac{DS(DS+1)}{2} \times \frac{(BS1+1)(BS1+2)}{2} \times \frac{(BS2+1)(BS2+2)}{2}$$

$$+ \sum_{i_1=0}^{BS1} \sum_{i_2=0}^{BS2} \left\{ \frac{(i_1+1)(i_1+2)}{2} \times \frac{(i_2+1)(i_2+2)}{2} \times [DS+BS1+BS2-i_1-i_2+1] \right\} .$$

Examples of state space sizes for the above model are given in Table 6.

Model with Two Types of Units

We show schematically in Figure 10 an example with two different types of machines or components. This model is the same as the (2,2,2) prototype except that we now have two types of machines (or assemblies, say A and B). Given that they are independent components, we can easily derive the formula for the state space of the model of Figure 10 as follows:

$$|S| = \left[\frac{(BS1A+1)(BS1A+2)}{2} \times \frac{(BS2A+1)(BS2A+2)}{2} + DSA(BS1A+1)(BS2A+1) \right]$$

$$\times \left[\frac{(BS1B+1)(BS1B+2)}{2} \times \frac{(BS2B+1)(BS2B+2)}{2} + DSB(BS1B+1)(BS2B+1) \right] .$$

Table 7 exhibits state space sizes for different sizes of the above model.

Model with Intermediate Repair Shop (Three Levels of Supply and Repair

We show in Figure 11 a (4,3,3) system. We let ISi be the number of spares allocated to intermediate station i , #IUi the number of spares available at intermediate station i , and #IRi be the number in or awaiting repair at intermediate station i .

TABLE 6

STATE SPACE SIZES FOR THE MODEL
WITH TRANSPORTATION PIEPLINES

BS1	BS2	DS	Number of States
1	1	1	49
2	2	2	508
4	4	2	6,800
6	4	2	17,430
6	5	2	28,812
6	6	2	44,688
8	8	3	229,950
16	12	3	4,933,383
24	18	4	60,916,375

TABLE 7

STATE SPACE SIZES FOR TWO ITEM-TYPE SYSTEM

BS1A	BS2A	BS1B	BS2B	DSA	DSB	Number of States
2	3	2	3	2	2	7,006
2	3	3	4	2	3	17,640
4	4	4	4	2	2	75,625
4	4	6	6	2	2	242,550
4	5	6	7	2	3	441,000
8	8	8	8	2	2	4,782,969
8	8	10	10	2	2	10,055,826

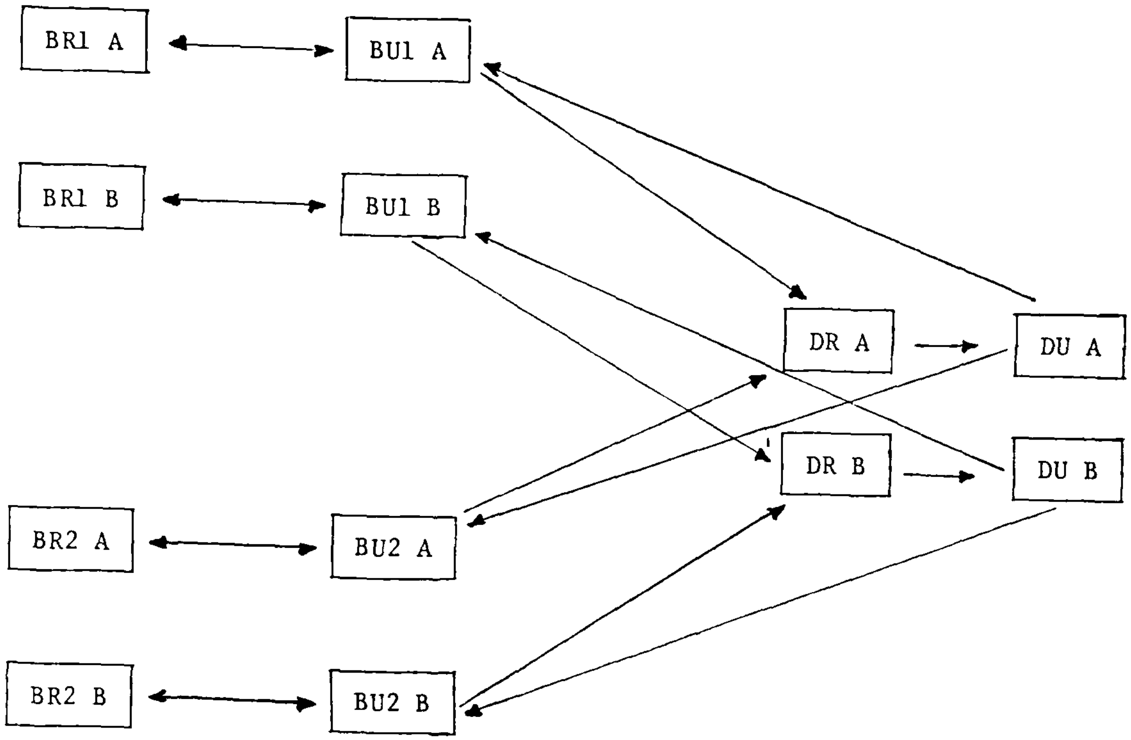


Figure 10. Multiple item type example.

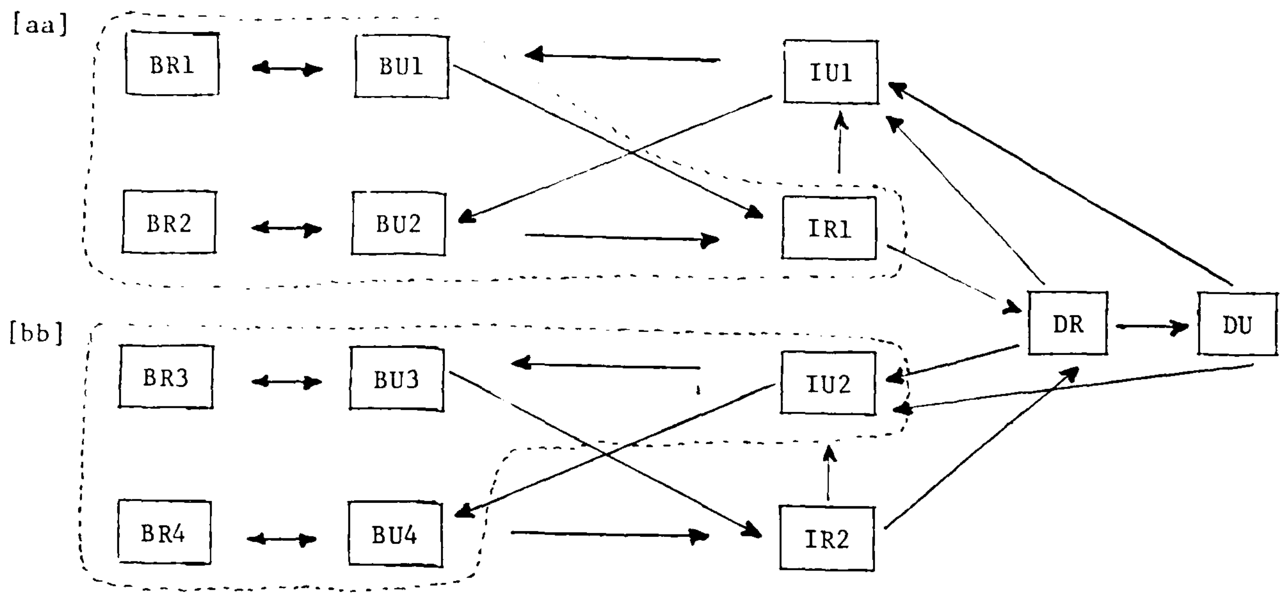


Figure 11. A (4,3,3) intermediate repair level system.

We now condition on the number of spares available at the depot (as before, DS represents the number of spares allocated to the depot and BSi the total number of machines allocated to Base i).

(i) #DU > 0 .

In this case, we have two independent (2,2,2) models, with two echelons and two bases. Thus

$$|S_+| = DS \times [(IS1)(BS1+1)(BS2+1) + (BS1+1)(BS1+2)(BS2+1)(BS2+2)/4]$$
$$\times [(IS2)(BS3+1)(BS4+1) + (BS3+1)(BS3+2)(BS4+1)(BS4+2)/4] .$$

(ii) #DU = 0 (no spares available at depot) and the depot owes i IU1 spares and j IU2 spares.

We first focus on the [aa] portion of the network.

(a) $0 \leq i \leq IS1$

1. #IU1 > 0 .

Then we have to consider $\sum_{i=0}^{IS1-1} (BS1+1)(BS2+1)(IS1-i)$ states.

2. #IU1 = 0 .

Then BS1 + BS2 + IS1 − i machines have to be allocated to the five stations contained in area [aa] of Figure 11. So we consider

$$\sum_{i=0}^{IS1-2} \sum_{k} \sum_{\ell} (k+1)(k+2)(\ell+1)(\ell+2)/4$$

states, where

k = number of machines at Base 1

ℓ = number of machines at Base 2

and the double sum is conditioned on

$$0 \leqslant k + \ell \leqslant BS1 + BS2 + IS1 - i$$

$$0 \leqslant k \leqslant BS1$$

$$0 \leqslant \ell \leqslant BS2$$

(b) $IS1 \leqslant i \leqslant BS1 + BS2 + ISj$

Here, we know that #IU1 = 0 . So the number of states to be considered is

$$\sum_{i=IS1}^{BS1+BS2+IS1} \sum_{k} \sum_{\ell} (k+1)(k+2)(\ell+1)(\ell+2)/4 ,$$

where the double sum is conditioned on

$$0 \leqslant k + \ell \leqslant BS1 + BS2 + IS1 - i$$

$$0 \leqslant k \leqslant BS1$$

$$0 \leqslant \ell \leqslant BS2 .$$

We are now able to write the formula giving the number of states of the entire state space:

$$|S| = DS \times [JS1(BS1+1)(BS2+1) + (BS1+1)(BS1+2)(BS2+1)(BS2+2)/4]$$
$$\times [IS2(BS3+1)(BS4+1) + (BS3+1)(BS3+2)(BS4+1)(BS4+2)/4]$$
$$+\left\{ \sum_{i=0}^{IS1-1} [(BS1+1)(BS2+1)(IS1-i)] + \sum_{0\leqslant i\leqslant BS1+BS2+IS1} \sum_{k=0}^{\min(BS1,BS1+BS2+IS1-i)} \sum_{\ell=0}^{\min(BS2,BS1+BS2+IS1-i-k)} \left[\frac{(k+1)(k+2)(\ell+1)(\ell+2)}{4}\right]\right\}$$
$$\times\left\{ \sum_{i=0}^{IS2-1} [(BS3+1((BS4+1)(IS2-i)] + \sum_{0\leqslant i\leqslant BS3+BS4+IS2} \sum_{k=0}^{\min(BS3,BS3+BS4+IS2-i)} \sum_{\ell=0}^{\min(BS4,BS3+BS4+IS2-i-k)} \left[\frac{(k+1)(k+2)(\ell+1)(\ell+2)}{4}\right]\right\}.$$

Table 9 gives some examples for different cases of the above model.

9. CONCLUSIONS

We can see that the truncation technique works for rather large, single-item, two-echelon Markovian systems. The applicability of the technique basically depends, for each individual case, on the rates and the mission times.

Figure 12 gives an idea of which rates will be acceptable if we wish to have an M/M/1 system probability of "overflowing" a certain n (= number in system) of .005 in 30 time units; that is, points on each contour correspond to points for which n is overflowed in 30 time units with probability .005. Points below this specific contour give absorbing probability less than .005 in 30 time units. For instance,

TABLE 9

STATE SPACE SIZE FOR INTERMEDIATE REPAIR SYSTEM

BS1	BS2	IS1	BS3	BS4	IS2	DS	Number of States
1	1	1	1	1	1	1	689
2	1	0	2	1	0	1	2,074
2	1	1	2	1	1	1	4,426
2	2	1	2	2	1	1	15,525
5	5	2	5	5	2	2	9,460,802
10	8	3	9	12	2	3	1,573,833,690

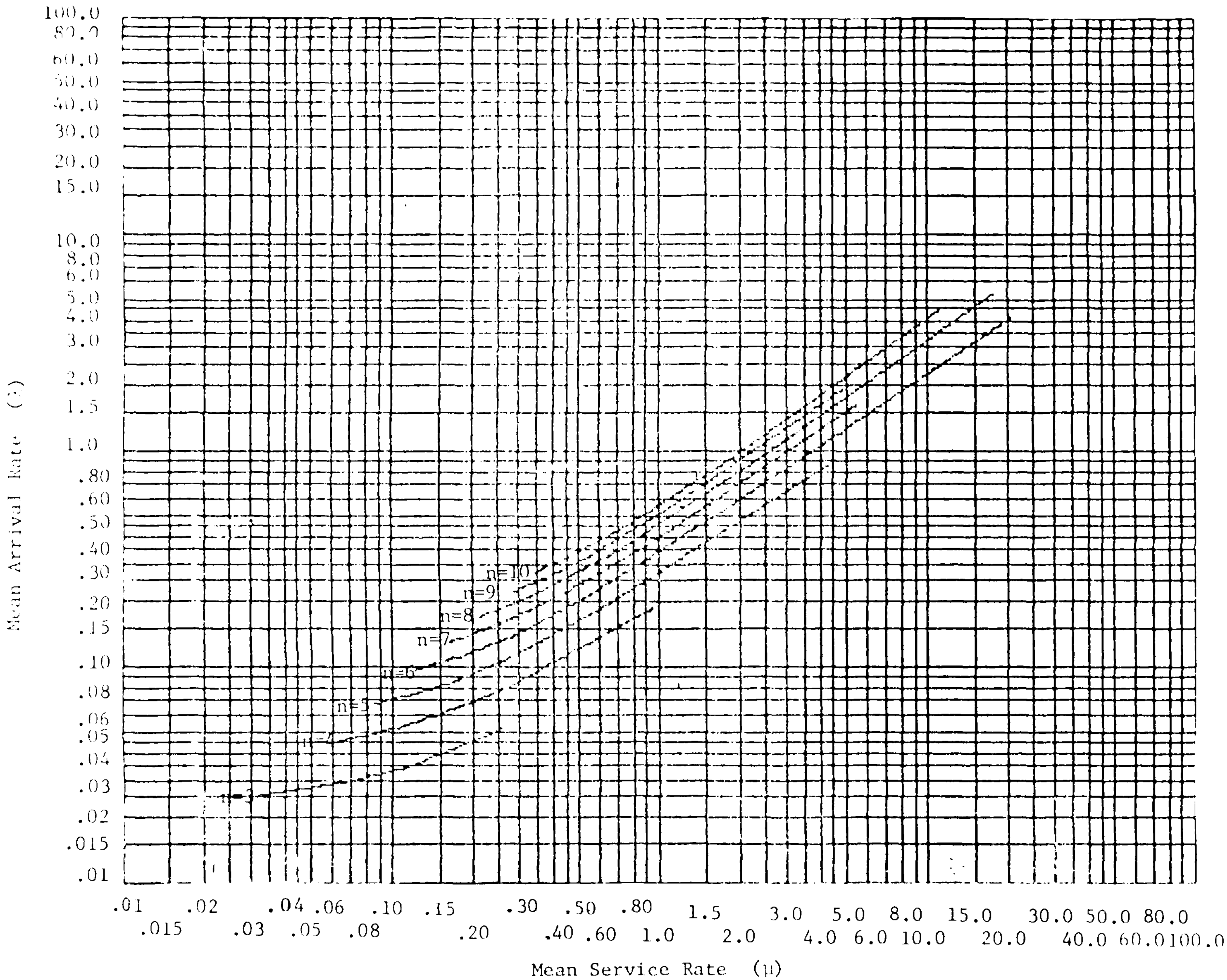


Figure 12. Points on each contour correspond to points for which the probability of exceeding n for an M/M/∞ queue in 30 time units equals .005 .

in Example 1, we came up with a Base 1 repair truncation point, TBR1, of 9. Now, if we still wish to have the system run for 30 time units and the absorbing probability for T_1^{BR} to be $\leq .005$ then, using Figure 12, we show that the arrival and failure rates that will suffice are those below the contour for $n = 9$. So according to this figure, our rates of "λ" = $\alpha_1\lambda_1$MS1 = 1.8 and "μ" = $c_1\mu_1 = 4.5$ will do, because this point is not above this contour. However, if, for example, $\alpha_1\lambda_1$MS1 = 8 and $c_1\mu_1 = 15$, then TBR1 = 9 will not do because the point made up of these rates lies above the $n = 9$ contour.

This figure can also be used for mission times other than 30 time units by scaling the rates appropriately. For example, if we are interested in a mission time equal to 15 time units with $\alpha_1\lambda_1$MS1 = 1.8 and $c_1\mu_1 = 4.5$ and we desire to know whether TBR1 = 7 might be a good truncation point (so as to allow the probability at absorption to be less than .005) then, knowing that this is equivalent to a system with mission time equal to 30, and $\alpha_1\lambda_1$MS1 = 0.9, $c_1\mu_1 = 2.25$ and using Figure 12, we see that we are above the contour for $n = 7$, and as a consequence it is not an appropriate truncation point. However, TBR1 = 9 would certainly work here since the point is below that contour. In fact, if a truncation point is satisfactory for a mission time t, it will certainly be satisfactory for mission times less than t. The opposite is not necessarily true.

Thus, if one has an idea of the total number of states that can be handled on a particular computer, using the state space size formulas [for example, equation (7)] can give rough ideas of what the truncation points must be for the computer to be able to handle the problem. Using Figure 12 in the manner illustrated above will indicate whether these truncation points are adequate for the rates of the particular system under study.

The more complex models reflected in state space sizes given in Tables 7, 8 and 9 can tax even the truncation technique. Truncation might be feasible for the pipeline models as the state space does not blow up too rapidly (see Table 7). However, the multi-machine and intermediate repair types of models, except for perhaps relatively small systems, generally have state spaces which are too large for even truncation to handle, and which would have to be analyzed by other techniques, such as simulation or approximation with simpler systems. Nevertheless, this truncation technique can handle systems large enough to model many realistic size problems of the real world.

ACKNOWLEDGMENTS

This work was partially supported by the Office of Naval Research under contract N00014-83-K-0217 and by the Army Research Office, contract DAAG29-84-K-0160.

REFERENCES

[1] Çinlar, E. (1975). *Introduction to Stochastic Processes.* Prentice-Hall, Englewood Cliffs, New Jersey.

[2] Grassmann, W. (1977). "Transient solutions in Markovian queueing systems," *Computers & Operations Research,* Vol. 4, 47-56.

[3] Grassmann, W. (1984). Personal communication.

[4] Gross, D. and D. R. Miller (1984). "The randomization technique as a modeling tool and solution procedure for transient Markov processes," *Operations Research,* Vol. 32, 343-361.

[5] Gross, D. and D. R. Miller (1984). "Multi-echelon repairable item provisioning in a time-varying environment using the randomization technique," *Naval Research Logistics Quarterly,* Vol. 31, 347-361.

[6] Gross, D., D. R. Miller, and C. G. Plastiras (1984). "Simulation Methodologies for Transient Markov Processes: A Comparative Study Based on Multi-echelon Repairable Item Inventory Systems," *Proceedings of the 1984 Summer Computer Simulation Conference*, Boston, MA, July 1984.

[7] Heyman, D. P. and M. J. Sobel (1982). *Stochastic Models in Operations Research, Volume 1.* McGraw-Hill, New York.

[8] Hillestad, R. J. and M. J. Carrillo (1980). "Models and techniques for recoverable item stockage when demand and the repair process are nonstationary--Part I: Performance measurement," N-1482-AF, Rand, Santa Monica, California.

[9] Hillestad, R. J. (1981). "Dyna-METRIC: Dynamic multi-echelon technique for recoverable item control," WD-911-AF, Rand, Santa Monica, California.

[10] Keilson, J. (1974). "Markov chain models -- rarity and exponentiality," ORC 74-32, Operations Research Center, University of California, Berkeley.

[11] Melamed, B. and M. Yadin (1980). "Randomization procedures in the computation of cumulative time distributions over discrete state Markov processes," Bell Laboratories, Holmdel, NJ.

[12] Miller, D. R. (1983). "Reliability calculation using randomization for Markovian fault-tolerant computing systems," *13th Annual International Symposium on Fault-tolerant Computing, Digest of Papers,* IEEE Computer Society, New York, 284-289.

[13] Muckstadt, J. A. (1973). "A model for a multi-item, multi-echelon, multi-indenture inventory system," *Management Science,* Vol. 20, 472-481.

[14] Ross, S. M. (1980). *Introduction to Probability Models,* 2 Ed. Academic Press, New York.

[15] Sherbrooke, C. C. (1968). "METRIC: A multi-echelon technique for recoverable item control," *Operations Research,* Vol. 16, 122-141.

APPENDIX

TRUNCATION POINT GRAPHS

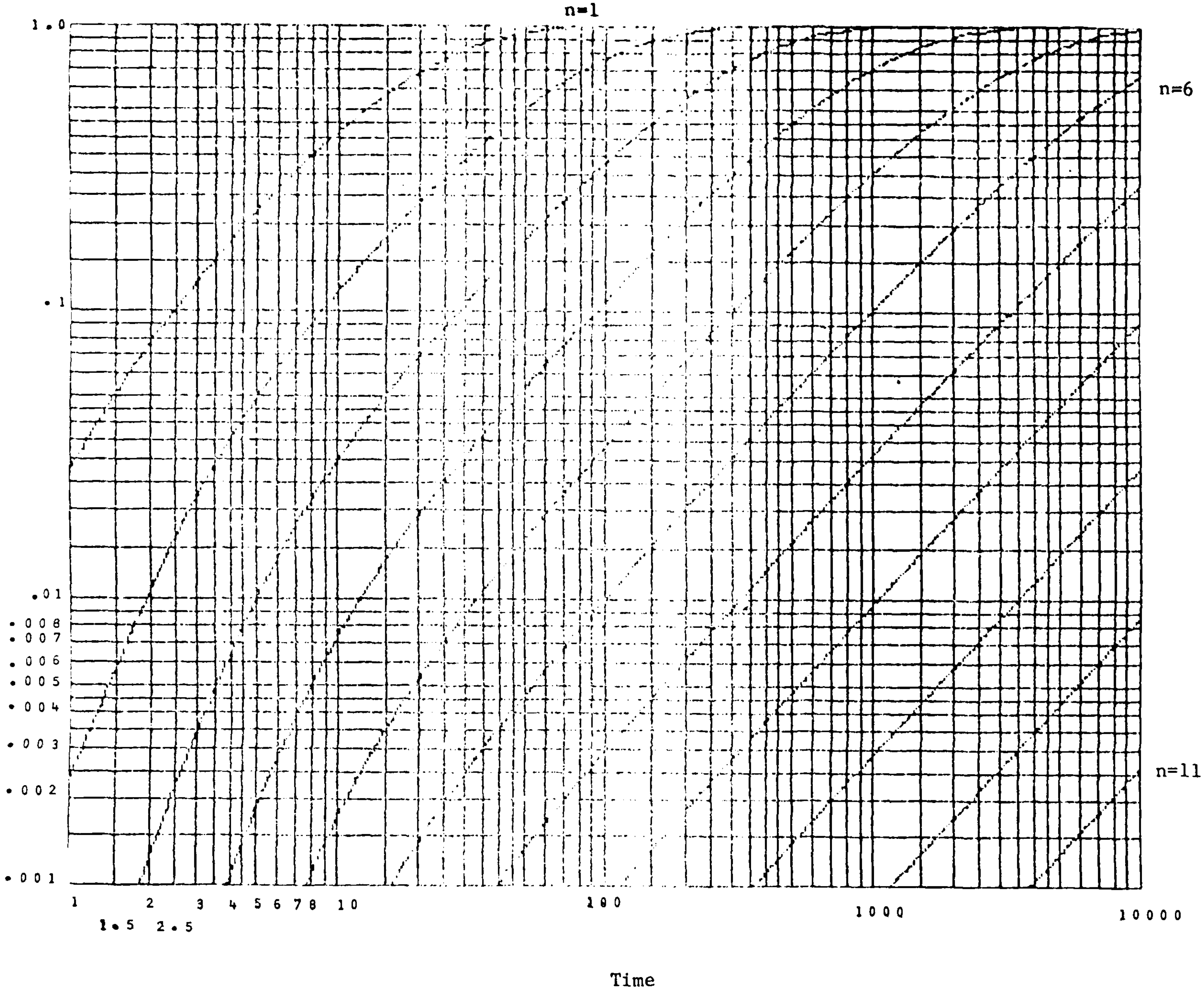


Figure A2. Absorbing Probability vs. Time for $\rho = 0.3$, truncation points of 1 to 11 inclusive.

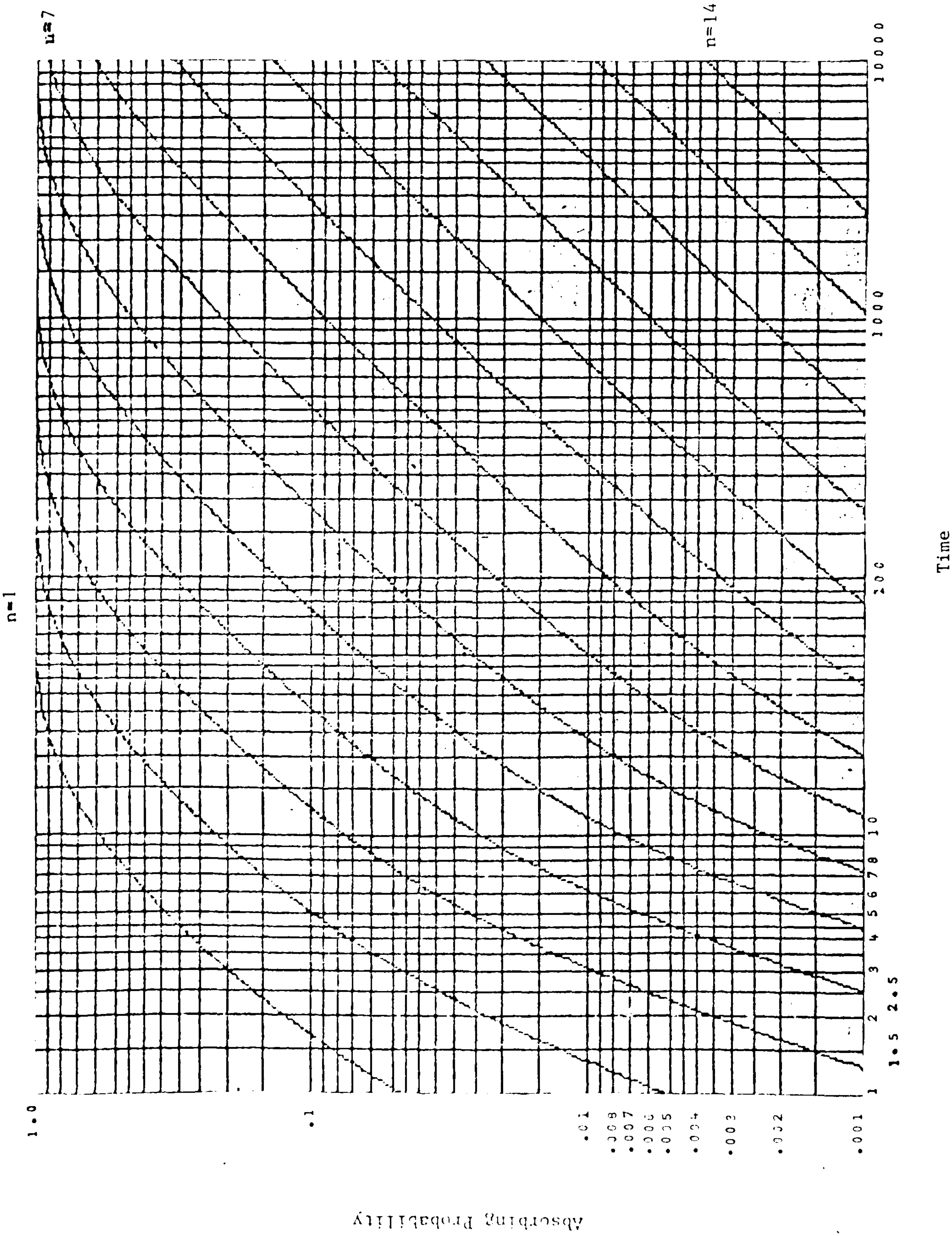


Figure A3. Absorbing Probability vs. Time for ρ = 0.4 , truncation points of 2 to 14 inclusive.

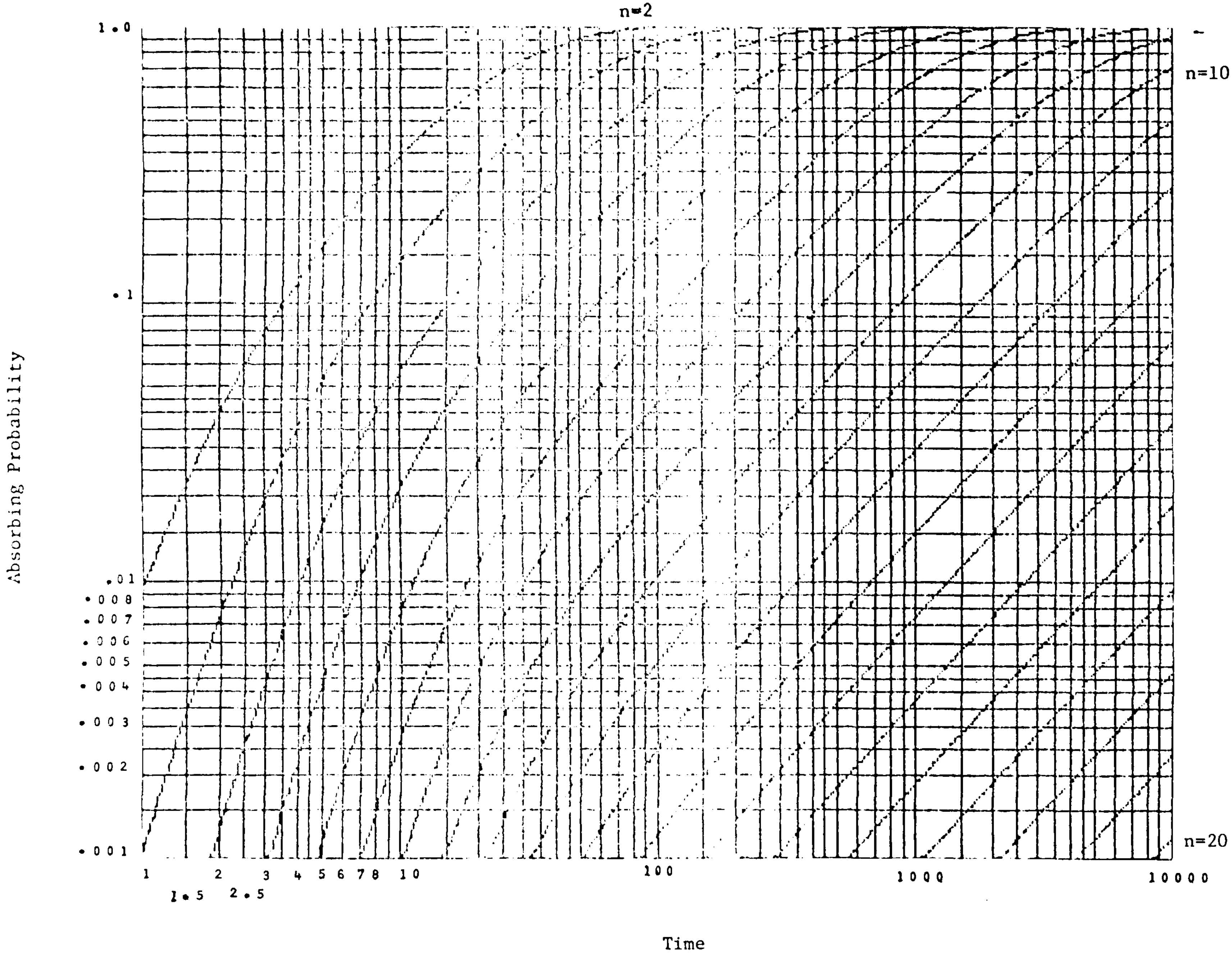


Figure A4. Absorbing Probability vs. Time for $\rho = 0.5$, truncation points of 2 to 20 inclusive.

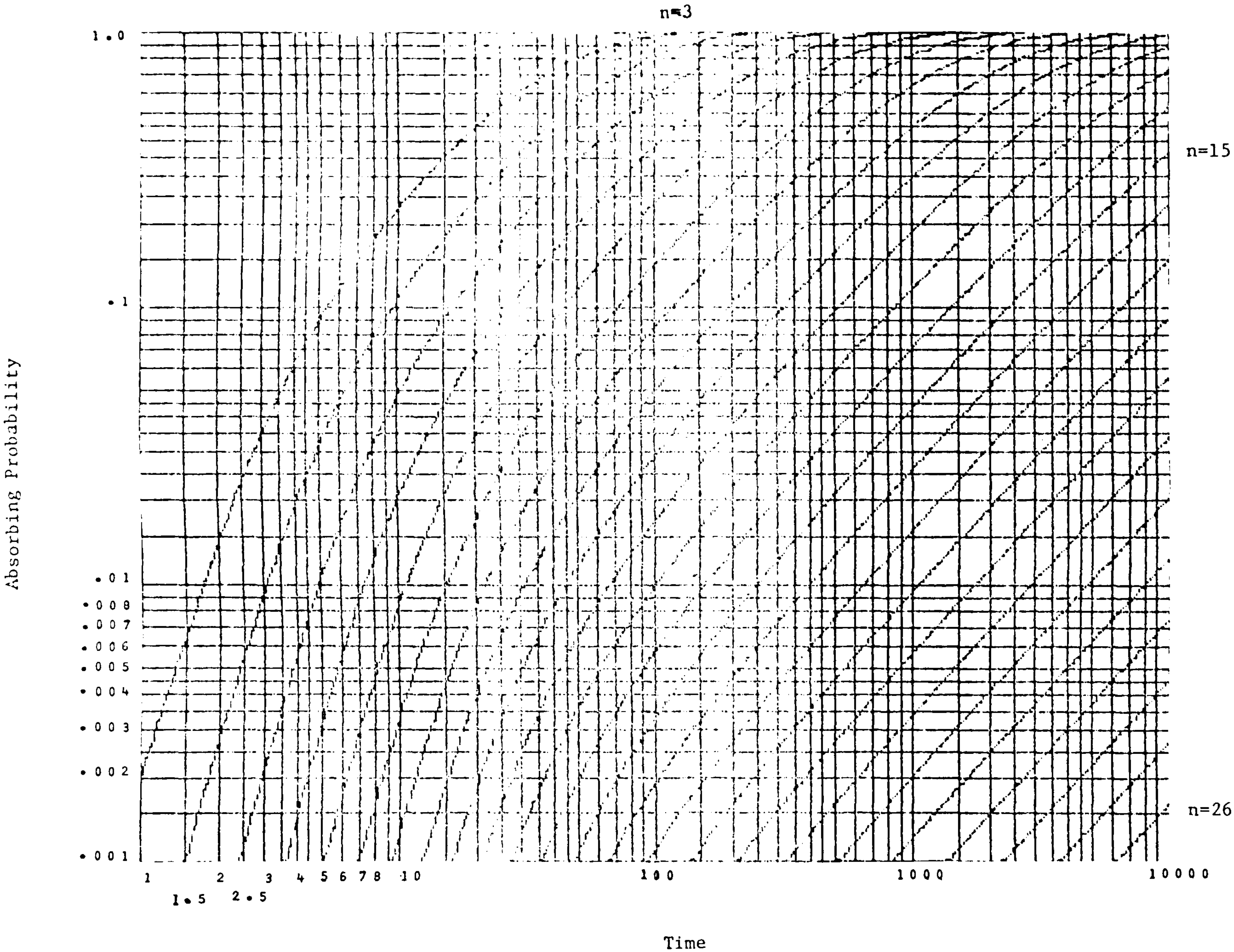


Figure A5. Absorbing Probability vs. Time for ρ = 0.6 , truncation points of 3 to 26 inclusive.

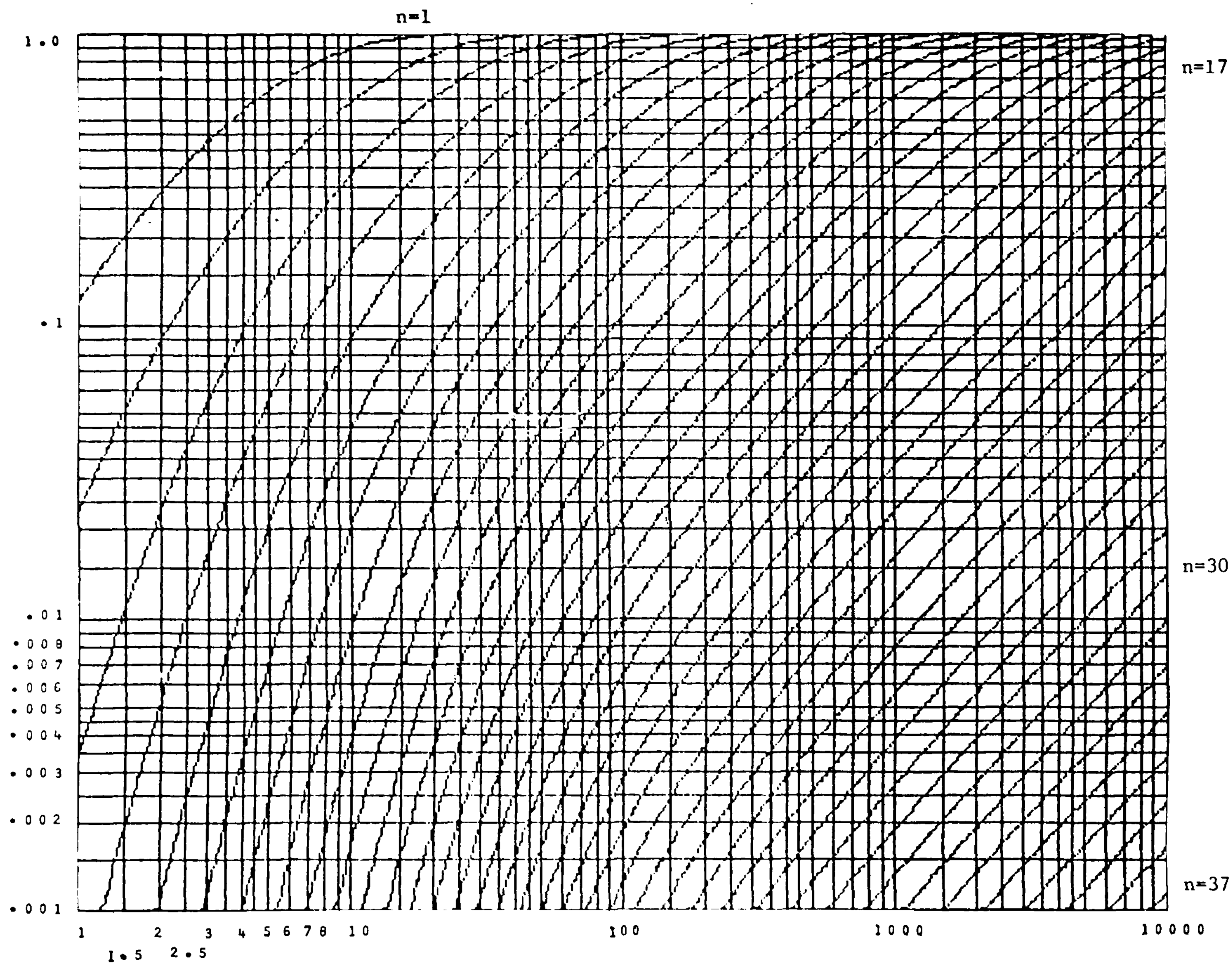


Figure A6. Absorbing Probability vs. Time for $\rho = 0.7$, truncation points of 1 to 37 inclusive.

Fig 1

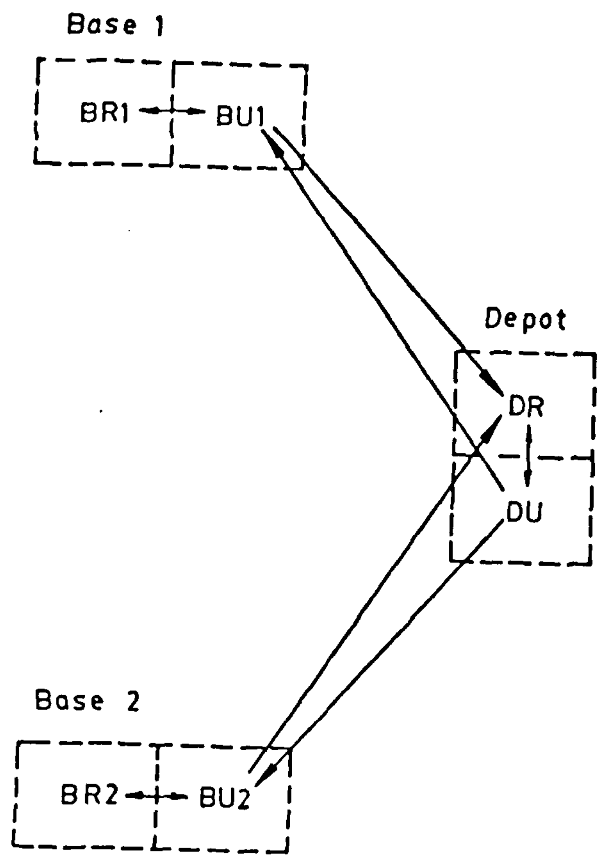


Fig 2

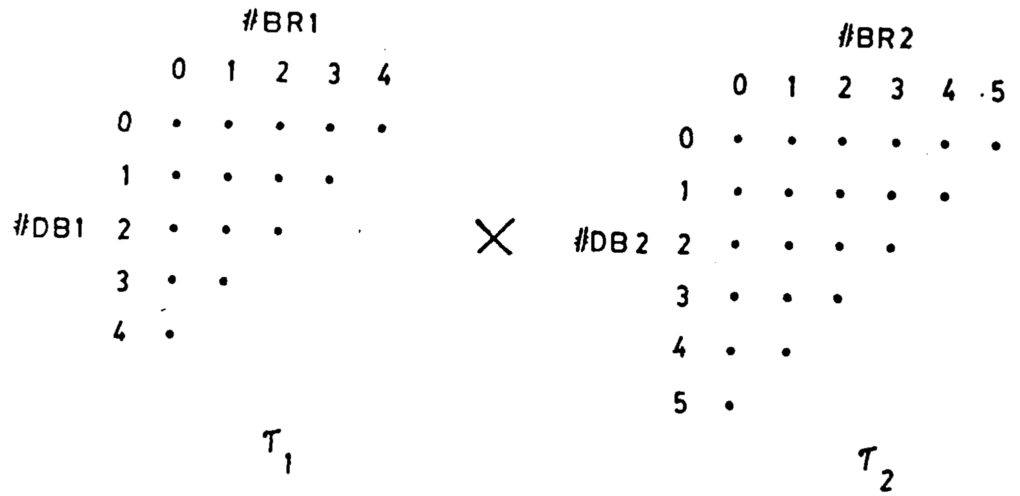


Fig. 3

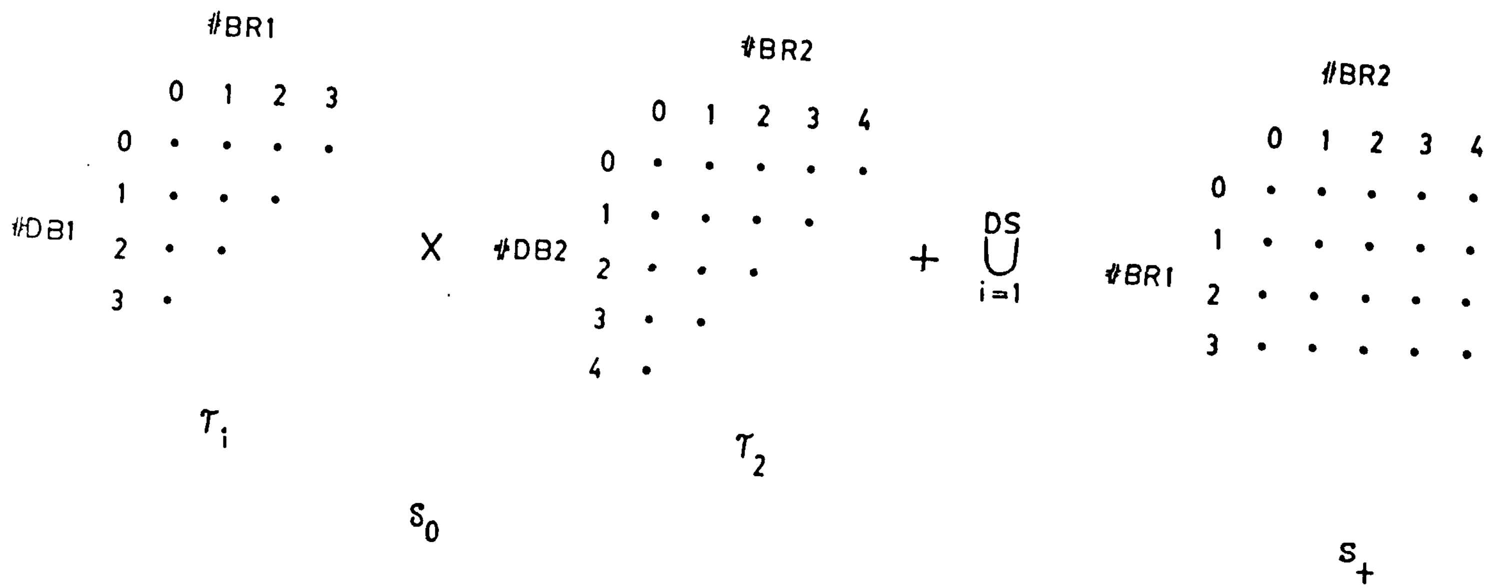
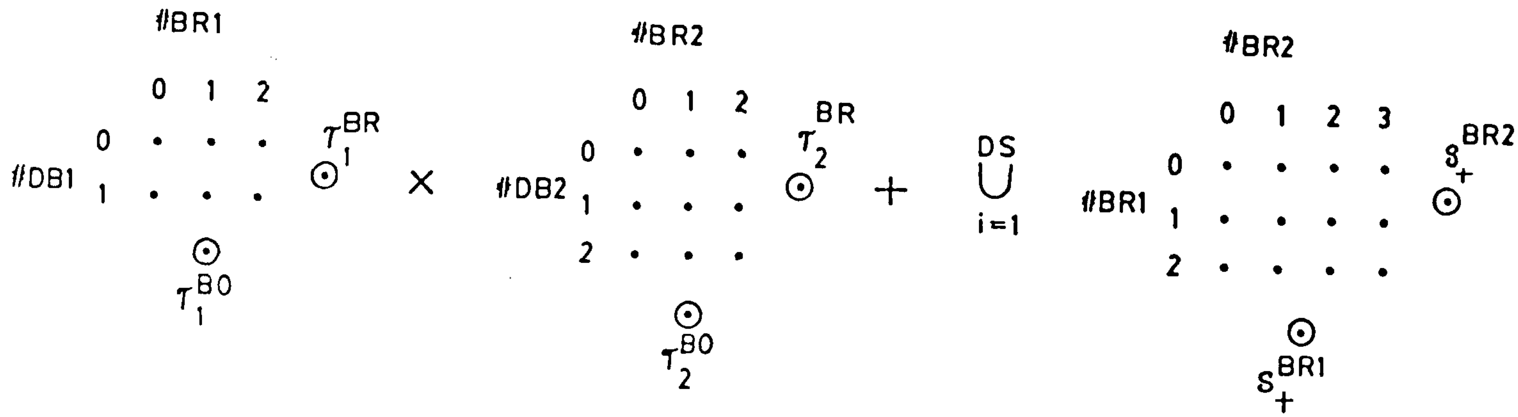
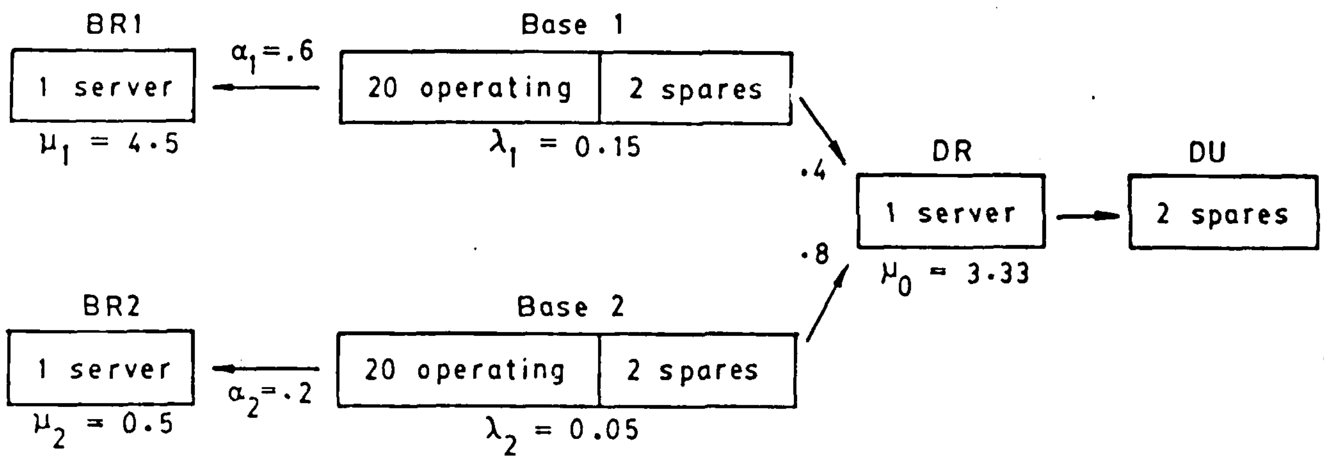

#BR1
0 1 2 3
#DB1
0 1 2 3
τ1
X
S0
#BR2
0 1 2 3 4
#DB2
0 1 2 3 4
τ2
+
DS
i=1
#BR2
0 1 2 3 4
#BR1
0 1 2 3
S+

Fig. 4.



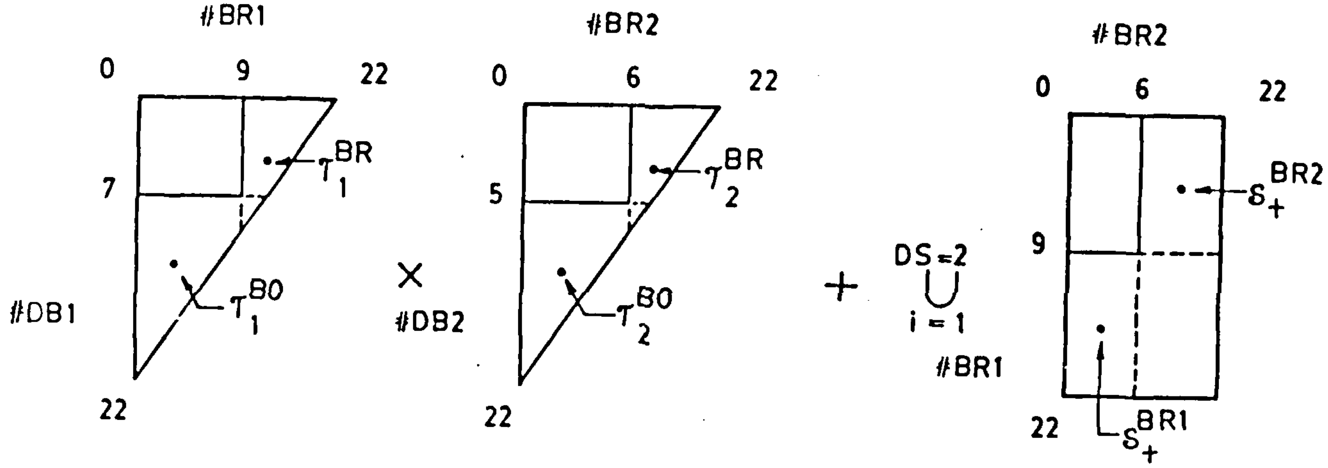
#BR1
0 1 2
#DB1
0
1
τ BR 1
τ BO 1
×
#BR2
0 1 2
#DB2
0
1
2
τ BR 2
τ BO 2
+
DS
i=1
#BR2
0 1 2 3
#BR1
0
1
2
S BR2 +
S BR1 +

Fig 5



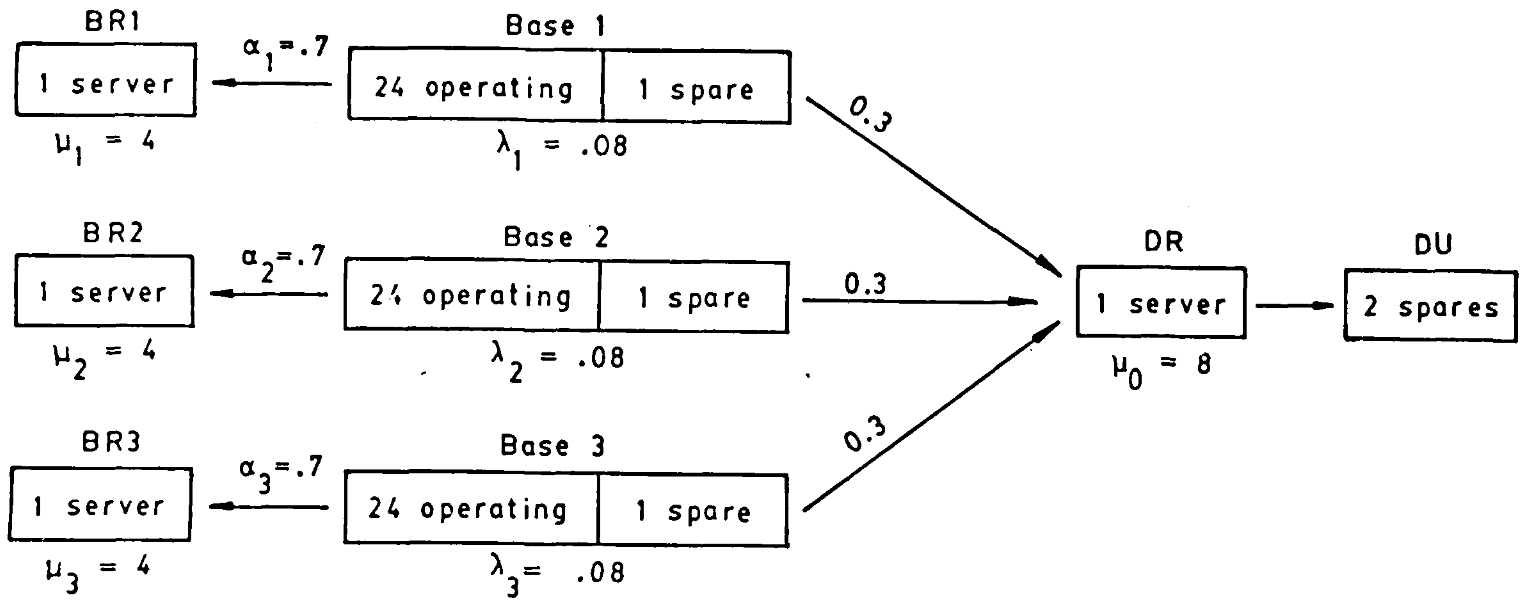
BR1
1 server
μ1 = 4.5
α1 = .6
Base 1
20 operating
2 spares
λ1 = 0.15
.4
.8
DR
1 server
μ0 = 3.33
DU
2 spares
BR2
1 server
μ2 = 0.5
α2 = .2
Base 2
20 operating
2 spares
λ2 = 0.05

Fig 6.



#BR1
0
9
22
7
22
#DB1
τ_1^{BR}
τ_1^{BO}
×
#BR2
0
6
22
5
22
#DB2
τ_2^{BR}
τ_2^{BO}
+
DS=2
⋃
i=1
#BR1
#BR2
0
6
22
9
22
δ_+^{BR2}
δ_+^{BR1}

Fig 7



BR1
1 server
μ1 = 4
α1 = .7
Base 1
24 operating
1 spare
λ1 = .08
BR2
1 server
μ2 = 4
α2 = .7
Base 2
24 operating
1 spare
λ2 = .08
BR3
1 server
μ3 = 4
α3 = .7
Base 3
24 operating
1 spare
λ3 = .08
0.3
0.3
0.3
DR
1 server
μ0 = 8
DU
2 spares

Fig 8.

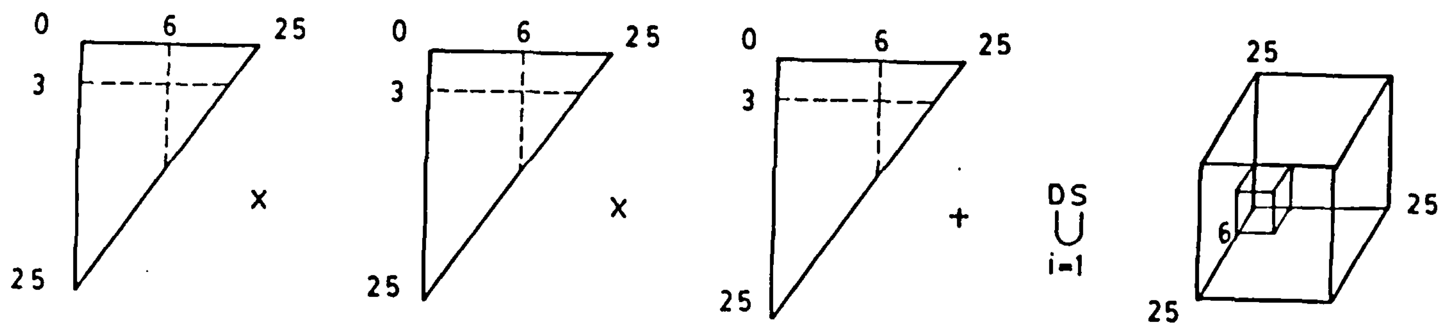


Fig 9

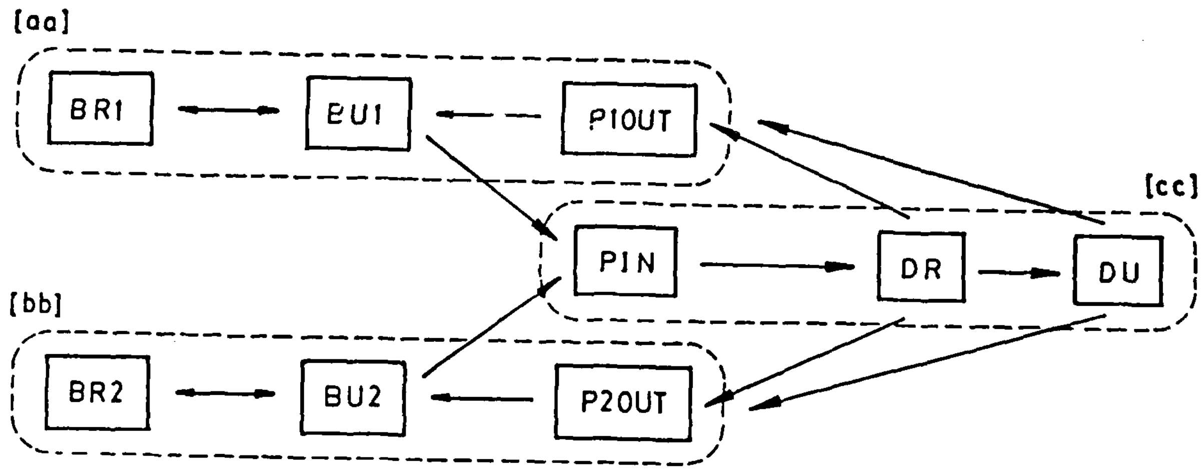


Fig 10

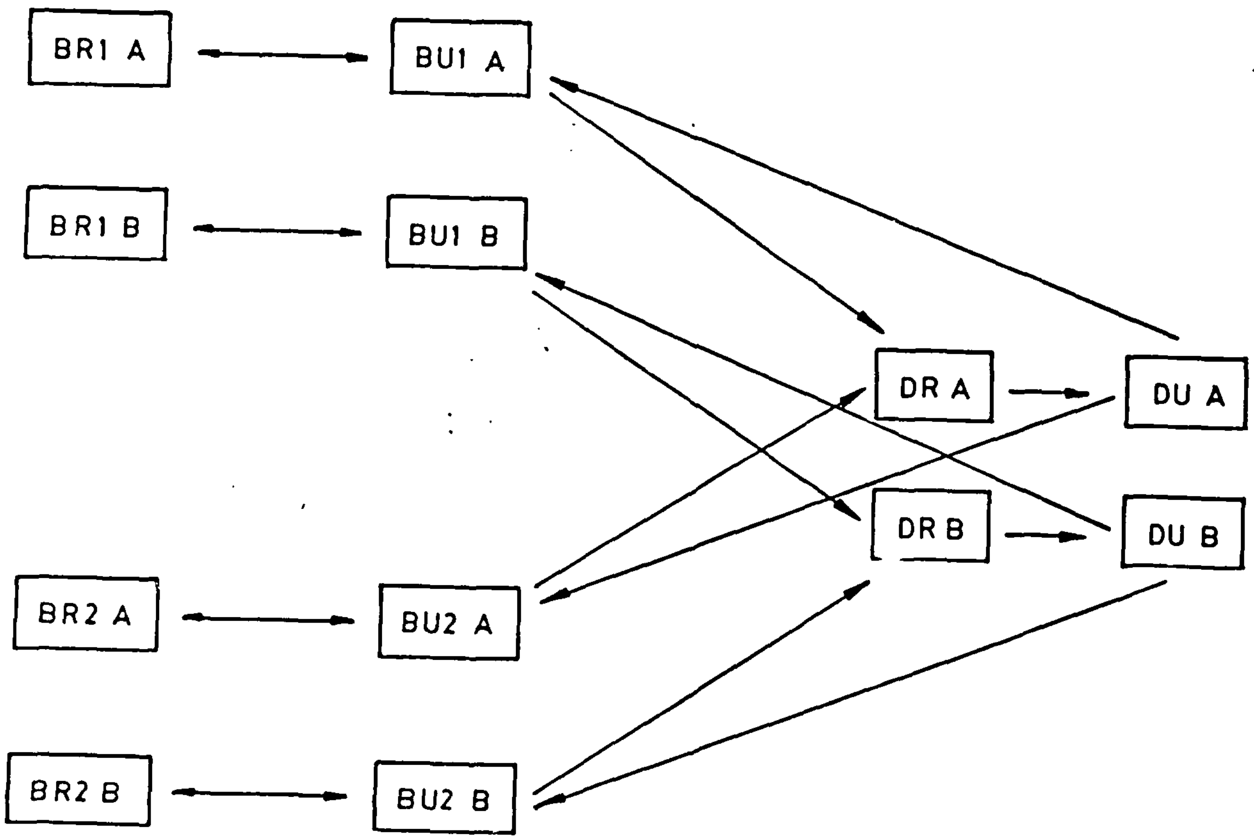


Fig 11

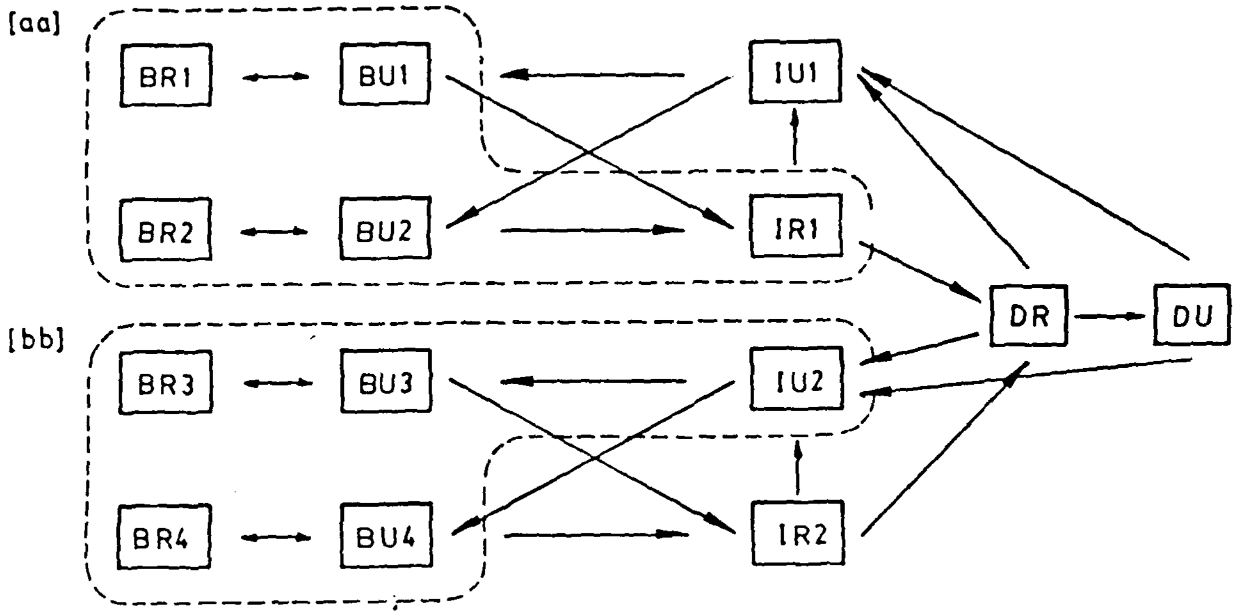
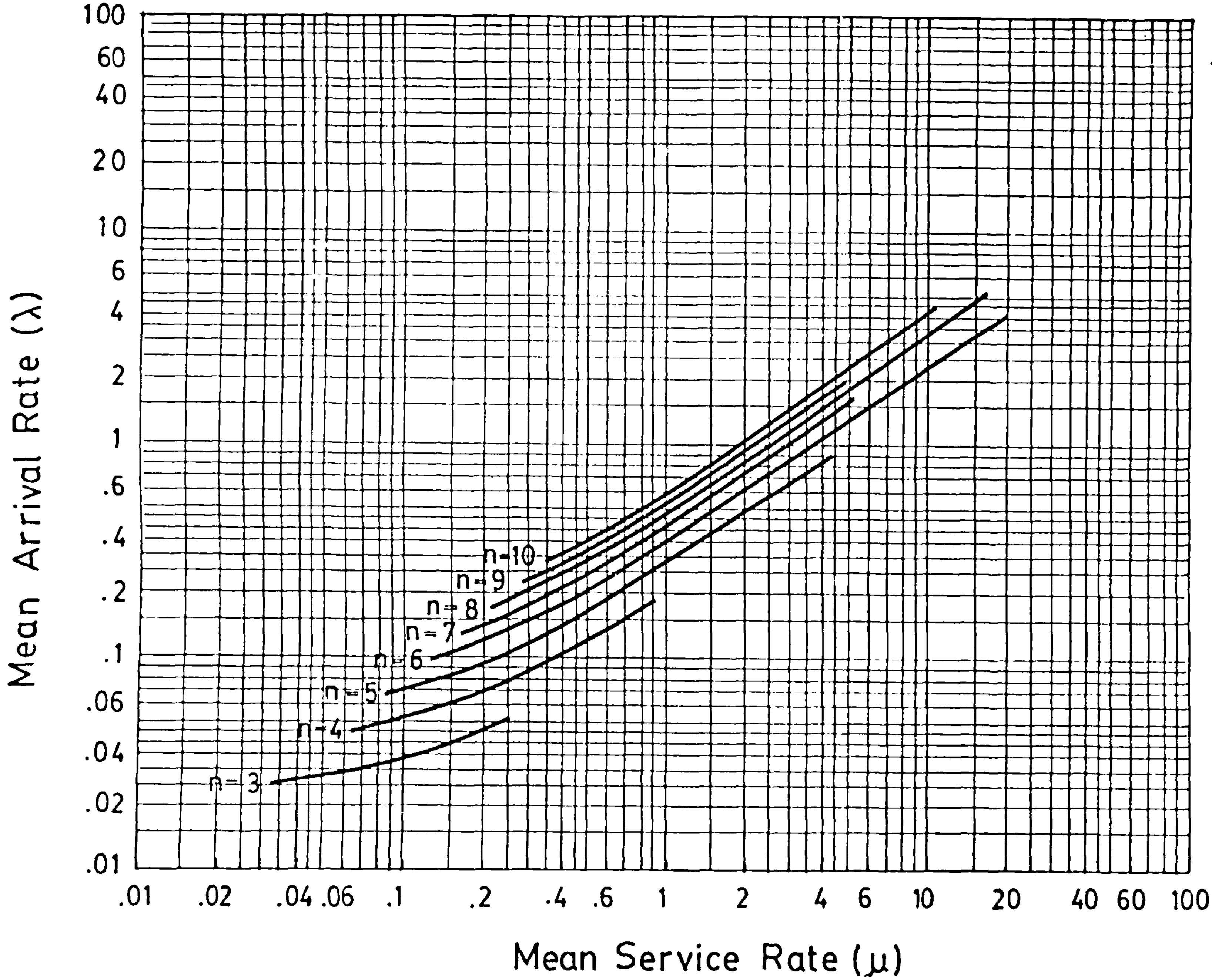


Fig 12



100
60
40
20
10
6
4
2
1
.6
.4
.2
.1
.06
.04
.02
.01
Mean Arrival Rate (λ)
n=10
n=9
n=8
n=7
n=6
n=5
n=4
n=3
.01
.02
.04 .06
.1
.2
.4
.6
1
2
4
6
10
20
40 60
100
Mean Service Rate (μ)

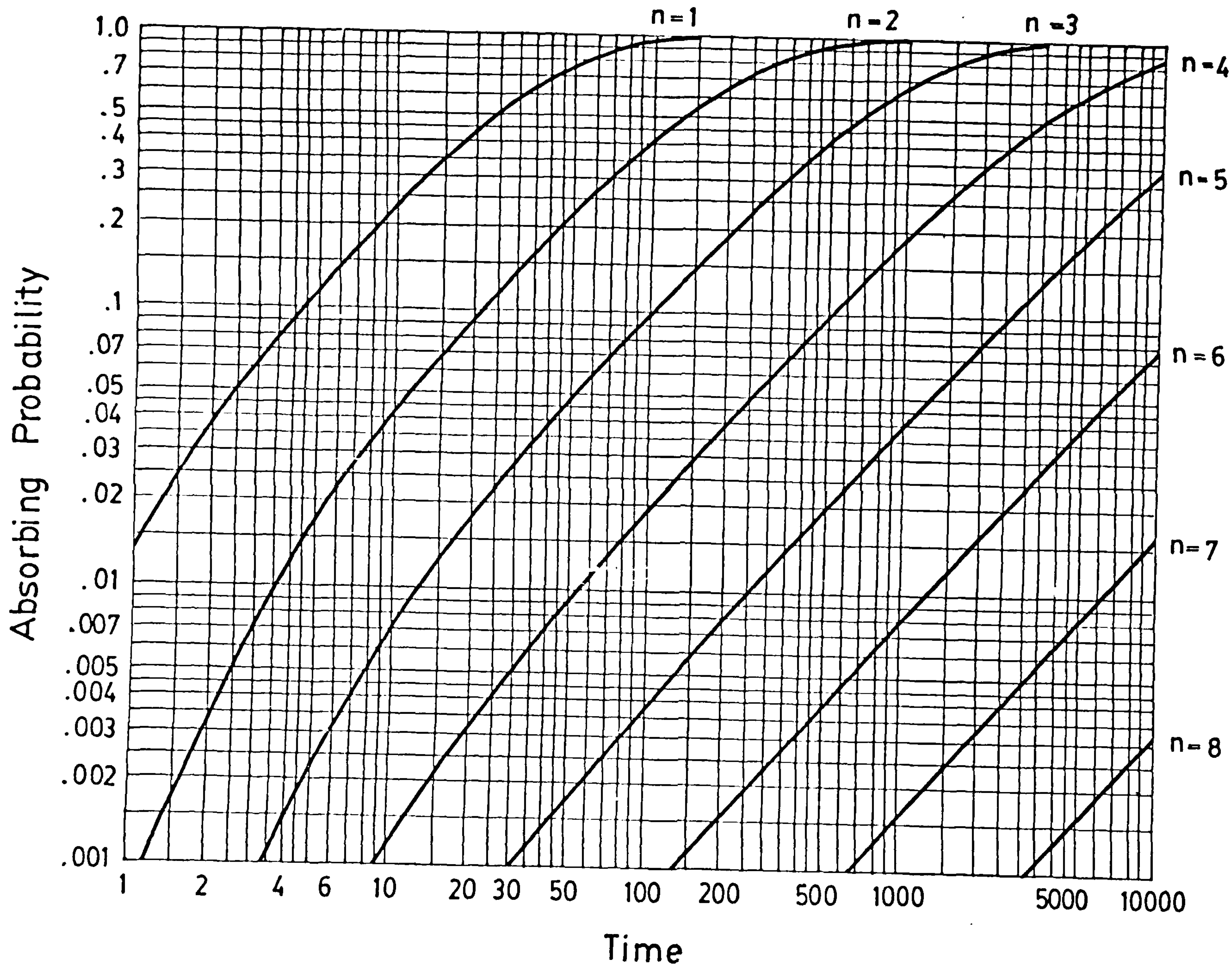
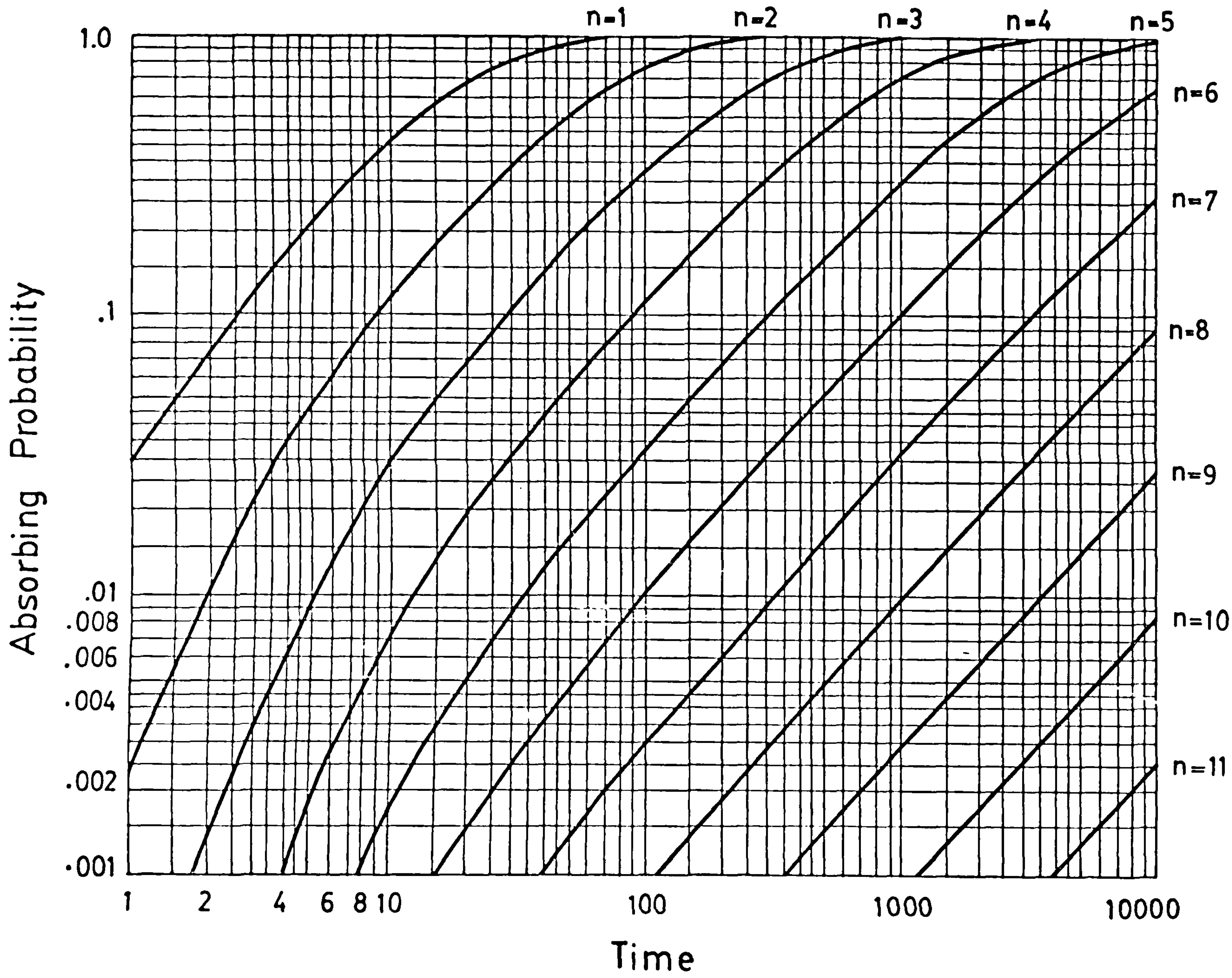

n=1
n=2
n=3
n=4
n=5
n=6
n=7
n=8
1.0
.7
.5
.4
.3
.2
.1
.07
.05
.04
.03
.02
.01
.007
.005
.004
.003
.002
.001
1
2
4
6
10
20
30
50
100
200
500
1000
5000
10000
Absorbing Probability
Time

Fig 11



n=1
n=2
n=3
n=4
n=5
n=6
n=7
n=8
n=9
n=10
n=11
1.0
.1
.01
.008
.006
.004
.002
.001
1
2
4
6
8
10
100
1000
10000
Time
Absorbing Probability

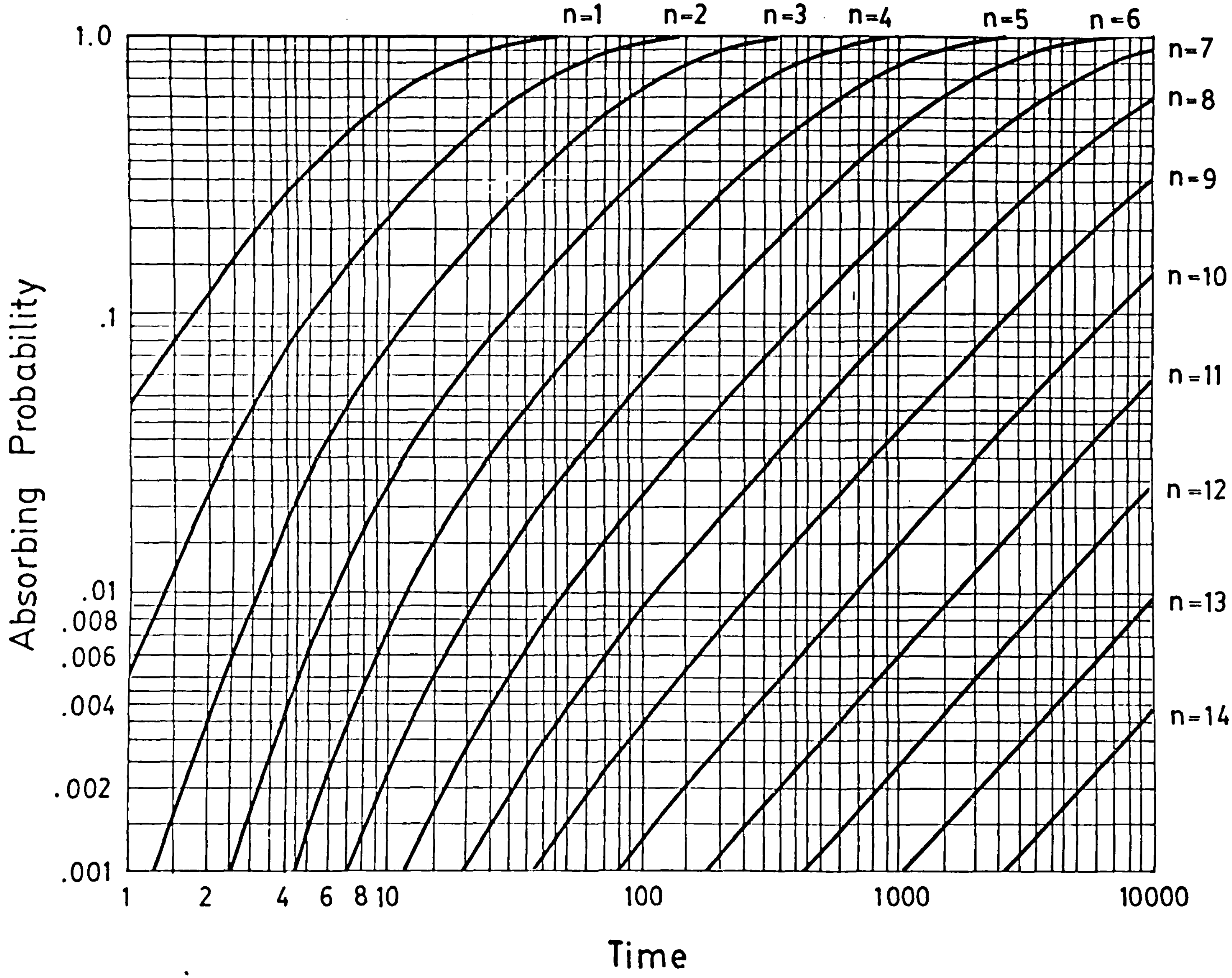
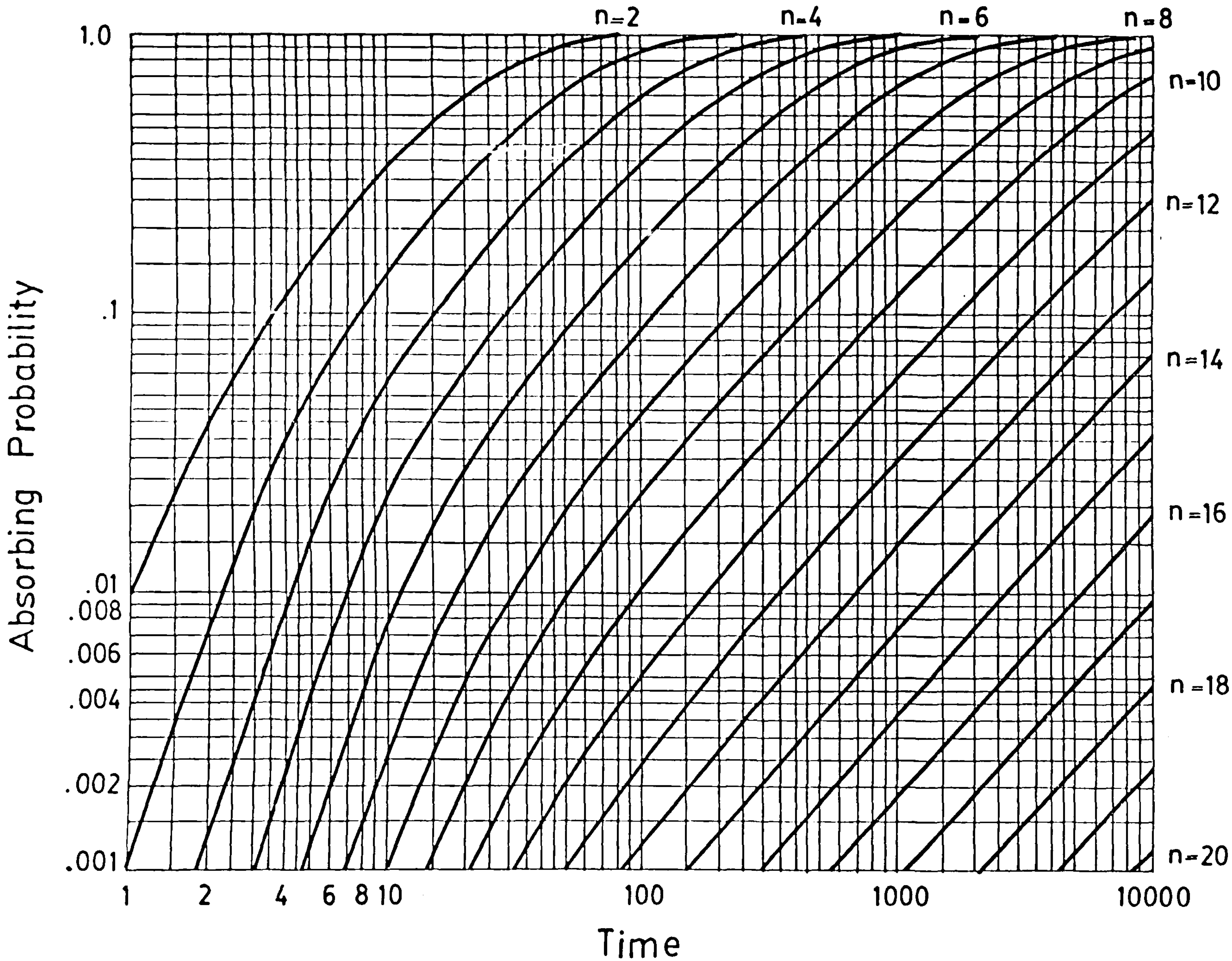
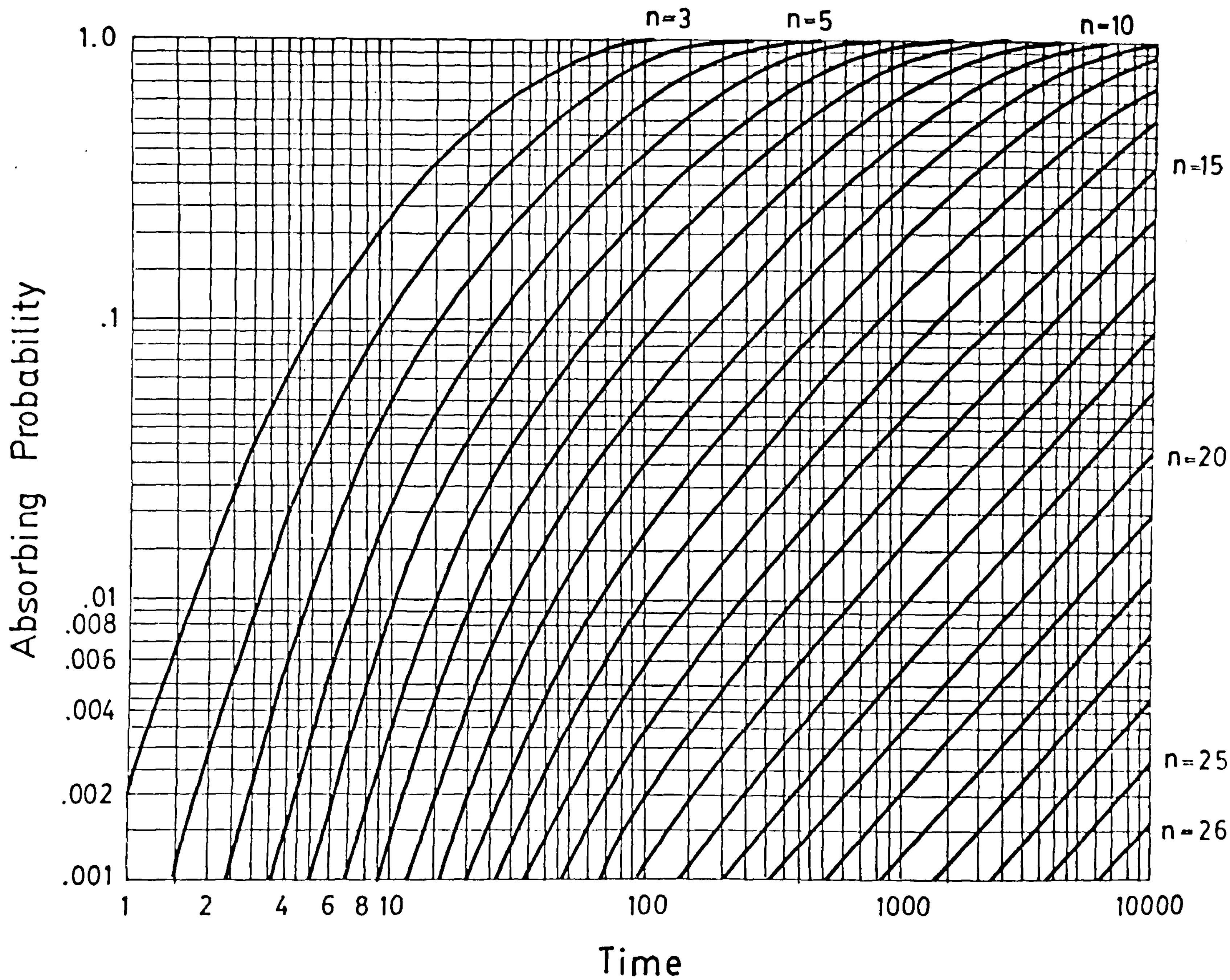

n=1
n=2
n=3
n=4
n=5
n=6
n=7
n=8
n=9
n=10
n=11
n=12
n=13
n=14
1.0
.1
.01
.008
.006
.004
.002
.001
1
2
4
6
8
10
100
1000
10000
Absorbing Probability
Time

Fig A3

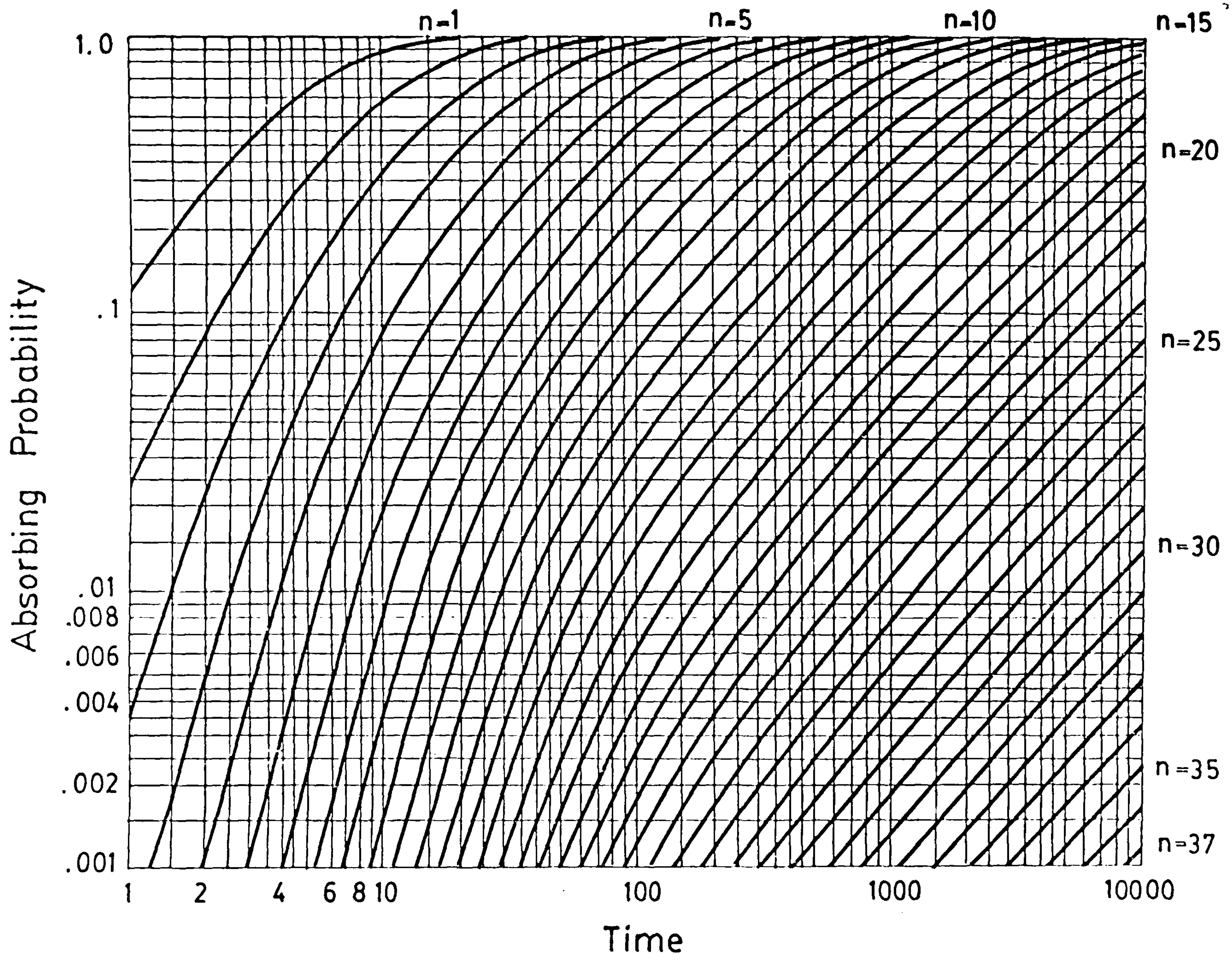


Absorbing Probability
1.0
.1
.01
.008
.006
.004
.002
.001
Time
1
2
4
6
8
10
100
1000
10000
n=2
n=4
n=6
n=8
n=10
n=12
n=14
n=16
n=18
n=20

Fig. A4



Absorbing Probability
1.0
.1
.01
.008
.006
.004
.002
.001
n=3
n=5
n=10
n=15
n=20
n=25
n=26
1
2
4
6
8
10
100
1000
10000
Time



n=1
n=5
n=10
n=15
n=20
n=25
n=30
n=35
n=37
1.0
.1
.01
.008
.006
.004
.002
.001
Absorbing Probability
1
2
4
6
8
10
100
1000
10000
Time

Fig A6

END

2-87

DTIC