ROBUST INITIALIZATION FOR STEADY-STATE SIMULATION

W. David Kelton
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

December 1985
Technical Report 85-35

Robust Initialization for Steady-State Simulation

W. David Kelton
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109–2117

Abstract

When simulating a stochastic model to estimate steady-state parameters of processes of interest, the initial conditions used to start the simulation often bias the estimators, sometimes severely. A common method for ameliorating this bias is to delete, or truncate, an initial portion of the run. If the initial conditions are far from steady state, a large amount of data may have to be discarded from the beginning of each simulation run, increasing costs. We investigate feasible methods for initializing such simulations that lead to lower estimator bias or, alternatively, less requisite deletion. Deterministic and stochastic initialization rules are compared, with appropriately chosen stochastic rules' being preferable. Forms for the initial distribution are suggested by the maximum entropy principle, and the parameters of these distributions may be specified from short pilot runs of the model. The goal throughout was to develop initialization methods that adapt to the particular model being simulated, yet are robust to variation in and misspecification of run parameters and conditions on the part of the analyst. These initialization rules are then applied to queueing models with both steady-state and transient properties known, to a computer model with known steady-state mean but unknown transient, and to a model of an actual manufacturing system with neither steady-state nor transient characteristics known.

Subject classification: 767 initialization methods, 769 initial transient reduction.

This research was supported by a contract from the Decision Technologies Division of Electronic Data Systems Corporation to The University of Michigan.

Simulations of dynamic, stochastic systems may usually be classified as either *terminating* or *steady-state*. Terminating simulations aim to estimate a parameter of the process defined relative to specific starting and stopping conditions, which entail a natural beginning and end of a simulation run. Steady-state simulations, however, have no natural starting or stopping rules, since the parameter to be estimated is defined as a limit as simulated time becomes infinite, and is defined independently of the initial conditions. Thus, the actual starting and stopping of a steady-state simulation becomes a difficult problem. Law (1980) contrasts these two types of simulations in several examples, and discusses the appropriate framework (*replication* of the entire simulation) for statistical analysis of the terminating type.

The framework for statistical analysis of steady-state simulation is far less clear, and has been the subject of a considerable amount of research over the past decade. Although it is hazardous to attempt any comprehensive classification of methodologies, most of these could be probably be put under one of the following six categories: Replication, batch means, time series modeling, spectral analysis, regenerative methods, and standardized time series. We surveyed, in Law and Kelton (1984, 1982), the first five of these as they existed then; additional work includes Schmeiser (1982) and Meketon and Schmeiser (1984) on batch means, Schriber and Andrews (1984) on time series modeling, and Heidelberger and Welch (1981) on spectral analysis. The standardized time series method is developed in Schruben (1983) and Goldsman and Schruben (1984). One important distinguishing characteristic among these methods is whether a single "long" run of the simulation is used, or multiple "short" runs. Batch means, time series modeling, spectral analysis, and regenerative methods work from a single long run, replication uses multiple shorter runs, and standardized time series may be applied to either a single long run or multiple short runs.

The motivation for this work was in the method of replication. Here, several independent and identically operated simulations are made in order to produce the independent and identically distributed (i.i.d.) observations $X_1, X_2, \ldots$, where X_j is the output measure of interest from the jth *entire* simulation run. For example, X_j could be the average over the jth run of the times in system of parts in a manufacturing simulation. In outline, this method is identical to that used for terminating simulations. Replication has two principal advantages:

(1) It is a very simple method to apply, by far the simplest of the six techniques mentioned above. Most simulation languages have some facility for specifying that a model be replicated some number of times.

(2) It produces the i.i.d. sequence $X_1, X_2, \ldots$ of observations on the system, to which the methods

of classical statistics may be *directly* applied. This includes not only the familiar confidence interval construction and hypothesis testing, but also methods such as multiple ranking and selection procedures, a very useful class of techniques in simulation since several alternative system designs are frequently of interest. The key ingredient in being able to apply such techniques is that we have i.i.d. data, with expectation equal to the desired performance parameter.

The last ingredient just mentioned, that the data are unbiased for the performance parameter of interest, leads to the principal disadvantage of replication:

(3) The initialization methods used to begin most steady-state simulations are typically far from being representative of actual steady-state conditions, leading to *bias* in the simulation output, at least for some early portion of the run.

This *initial transient problem* has limited the effectiveness of replication for steady-state analysis. The advantages (1) and (2) above, however, are sufficiently attractive to promote attempts to overcome the disadvantage (3). Most research to date on initial transient detection and reduction has focused on *deletion* (or *truncation*) of some initial portion of the run where the transient bias is most severe; see, for example, Gafarian, Ancker, and Morisaku (1978), Adlakha and Fishman (1982), Schruben (1981, 1982), Schruben, Singh, and Tierney (1983), Welch (1983), and Kelton and Law (1981). These methods attempt to identify at what point the output record from within a run appears to have attained "steady state," and data are retained only from this point on. Such identification may be problematic for some models, and, even if the identification is successful, may prescribe that a large amount of initial deletion is necessary to be safe. In multiple replications, deletion would be necessary from *each* run, possibly resulting in large portions of the output records' being discarded.

The assumption behind the deletion approach is that the initial conditions are given, and are not representative of steady state. Our purpose here is to investigate better ways to initialize steady-state simulations that promote more rapid convergence to steady state. Deletion may still be necessary, but the amount of deletion should be reduced. Our goal is to find feasible, model-dependent methods that are robust to variations and imprecisions on the part of the simulator. If the deletion tactic represents a "cure" to an existing initial transient problem, then robust initialization may be thought of as an attempt at preventative medicine.

We limit our consideration here to models that may be initialized by specifying a single, discrete value; extending these ideas to multivariate state spaces is the subject of current research. Thus, our goal here is to find a way to specify "reasonable" values for the initial state of the system. We develop alternative initialization methods in Section 1, which are then investigated in Section 2 on models about which varying amounts are analytically known. Section 3 contains some general conclusions.

1. INITIALIZATION METHODS

In this section we discuss alternatives for initialization. The basic distinction is drawn between *deterministic* and *stochastic* initial states.

1.1. Deterministic Initial States

The most common way of initializing replications is to begin each one in an identical deterministic state. The canonical example is empty and idle initialization of queueing models (usually the default in simulation languages, which may not be conveniently changed). This clearly results in the runs' producing i.i.d. X_j's, provided that they are terminated in the same way. The choice of the initial state may consider convenience or economy in programming.

A better (at least from the statistical standpoint) consideration would be some sense of similarity to steady-state conditions. Wilson and Pritsker (1978) evaluated finite state space Markov processes and found that choosing an initial state near the mode of the steady-state distribution produced favorable results. We found in Kelton and Law (1985), Kelton (1985), and Murray and Kelton (1985) that, for $M/M/s$, $M/E_m/1$, $E_m/M/1$, and $M/E_m/2$ queues, the initialization state (representing the number of customers or Erlang stages present at time 0) can have a dramatic impact on the rate of convergence to steady state, and that initializing in a state at least as congested as the steady-state mean induced comparatively short transients. Optimal initial states were found, but the cost functions (bias and time to near-stationarity) were quite steep around the optima; thus, one would have to be quite precise (or lucky) in choosing the initial state to realize the potential gain. In order to implement such ideas, some *a priori* estimate of the mode or mean of the steady-state distribution would be required, perhaps available from debugging or pilot runs of the model.

It is characteristic of deterministic initialization that each replication will pass through a transient phase that is identical across replications, since the initial state is always the same. For example, in a queueing model initialized in an undercongested state, all replications will be downwardly biased. One could purposely vary the initialization state across the replications in a study to produce a range of biases that may offset one another. However, this destroys the identically distributed nature of the replications, rendering subsequent statistical procedures invalid.

1.2. Stochastic Initial States

One way to allow for different realized initialization states across replications, yet retain probabilistic identity of the runs, is to draw the initial state from some distribution. The replications, though actually begun in generally different numerical positions, are still i.i.d. since the *rule* by which the initial states are chosen is always the same. Indeed, the ideal would be to draw the initial state from the steady-state distribution itself, and the simulation would immediately be in steady state. This being clearly unworkable, we need to find appropriate *feasible* ways to specify an initial distribution. Both the *form* and the *parameter(s)* of such a distribution must be determined somehow.

Rarely would there be information available sufficient to specify the form of an initial distribution completely. In the absence of much information about the appropriate initial distribution, one way of specifying the form follows the *principle of maximum entropy*, developed by Jaynes (1957) and going back in essence to Laplace's "Principle of Insufficient Reason" (stating that, in the absence of contrary information, two complementary events should be assigned equal probabilities). Entropy maximization produces a discrete probability mass function (p.m.f.) $\mathbf{p}(\mathbf{x}) = \{p(x_1), \ldots, p(x_k)\}$ on the finite domain $\{x_1, \ldots, x_k\}$ that maximizes the p.m.f's entropy

$$H(\mathbf{p}) = -\sum_{i=1}^{k} p(x_i) \ln p(x_i)$$

subject to the ***expectation constraints***

$$\sum_{i=1}^{k} g_j(x_i) p(x_i) = \nu_j \qquad \text{for} \quad 1 \leq j \leq t \tag{1}$$

where the g_j's are given, real-valued functions. (If we do not wish to impose any expectation constraints, (1) may be omitted by taking $t = 0$.) Formally, the problem is then

$$\max_{\mathbf{p}} H(\mathbf{p}) \qquad \text{subject to} \qquad \sum_{i=1}^{k} p(x_i) = 1 \quad \text{and} \quad (1). \tag{2}$$

Solution of (2) can, in principle, be accomplished by introduction of Lagrange multipliers to account for the constraints. The p.m.f. resulting from entropy maximization is the "maximally noncommittal" distribution that obeys the constraints, in the sense that it agrees with the (observed) frequency distribution obtainable in the largest number of ways from actual data that conform to the constraints (see Jaynes).

In our situation, we considered two possibilities for the expectation constraints (1). First, we may have no information at all from which to specify such constraints, so that (1) is vacuous. In this case, solving (2) leads to a *discrete uniform* distribution on $\{x_1,\ldots,x_k\}$, i.e., $p(x_i)=1/k$ for all i, and we would need only specify a range for the initial state. Second, we may want to impose a mean constraint $\nu_1=\nu$ on the chosen p.m.f., i.e., in (1) we take $t=1$, $g_1(x_i)=x_i$, and $\nu_1=\nu$. Our interest will be in the case $\{x_1,\ldots,x_k\}=\{0,\ldots,m\}$, and with these assumptions the solution to (2) is

$$p(c)=ab^c \qquad \text{for} \quad c=0,1,\ldots,m, \tag{3}$$

for some positive constants a and b chosen to satisfy the constraints. This leads to setting $a=(1-b)/(1-b^{m+1})$, where b is a solution to

$$\frac{-(m+1)b^{m+1}(1-b)+b(1-b^{m+1})}{(1-b)(1-b^{m+1})}=\nu,$$

which would have to be solved numerically. The situation simplifies considerably, however, if we consider the case of large m, i.e., we ignore the truncation. In this case we can reparameterize (3) as the more familiar $p(c)=p(1-p)^c$ with $p=1/(\nu+1)$. Given the uncertainty of the information we are likely to have in choosing an initial distribution, this approximation seems warranted. Thus, in this second approach, we need an *a priori* estimate of the steady-state mean ν of the system state. Finally, it is interesting to note that the geometric distribution for the number of customers in system *is* the correct steady-state form for several queueing models, such as $M/M/1$ and $G/M/1$.

To summarize, the maximum entropy principle suggests that, in the absence of any feeling about the expected system state in steady state, we use a discrete uniform initial distribution, on $\{0,1,\ldots,m\}$ in our applications. Thus, we must specify m somehow. If we have an estimate or guess about the expectation ν of the system state in steady state, we could use a geometric (perhaps truncated at some m) distribution with parameter $p=1/(\nu+1)$. If one has additional information in the form of expectation constraints, they could be built in and the form of p found. We did not consider this explicitly, however, since our attitude was that relatively little will be known *a priori* about the steady-state distribution in most cases.

By using a uniform or geometric initial distribution, we have replaced the problem of specifying a deterministic initial *state* by that of specifying the *parameter(s)* of the initial distribution. It seems that misspecification of such parameters may be less severe than misspecification of a single state in which *all* replications will begin. These parameters could be specified either arbitrarily, or by looking at some short "pilot" runs of the model, which may be available anyway from model debugging and testing. The maximum state observed over such pilot runs would determine m for the uniform initial distribution, while the average observed system state would give a rough value of ν and hence p for the geometric initialization. These pilot runs must also be initialized (and terminated) somehow, so we are still faced with the need to specify something rather arbitrarily. Again, the starting and stopping of these pilot runs is one more level removed from the actual production runs of the model, and their specification should in turn be less important than arbitrary setting of the parameter(s) of the maximum entropy distributions developed above. Presumably one could think of initializing the pilot runs themselves stochastically, based on pre-pilot runs, and so on. We stopped, however, with deterministically initialized pilot runs of somewhat arbitrary length since the final production results appeared to be fairly robust to changes in the conditions of these pilot runs. It will be necessary to specify *some* parameters arbitrarily and deterministically, at some level removed from the production runs; it appeared from our results in Section 2 below that using arbitrarily designed pilot runs to specify the parameters of the initial distribution was sufficient.

1.3. Implementation Issues

In testing the above ideas on various models, we encountered some issues in terms of their implementation. Most importantly, if a queueing model is "loaded up" with some number of customers initially, they must not be counted in the final output statistics. For example, if we are observing the delay in queue process, the delays of those customers artificially placed in the system at time zero cannot be counted in the average, to avoid potential bias. This can be accomplished by flagging the artificial entities, for example by assigning their time-of-arrival attribute to be negative.

2. RESULTS

In order to evaluate the initialization methods discussed in Section 1, we applied them to a variety of models, ranging from simple queues with known steady-state *and* transient expected behavior, to a real simulation of a complex manufacturing system with neither steady-state nor transient properties known.

2.1. Evaluation Criteria

We used three criteria to evaluate the effectiveness of the alternative initialization methods:

(1) Plots of the expected transient response (or an estimate of it) as a function of discrete time (customer or job number). This provides a qualitative yet informative view of the nature and rate of convergence of the model to steady state for initialization alternatives.

(2) The percent bias (with respect to the steady-state mean μ) of the expected average $\mathrm{E}(\overline{Y}(n))$ of the first n points in a run, $100|\overline{Y}(n) - \mu|/|\mu|$. This provides a measure of the bias in a run of length n points without deletion.

(3) The time (in number of simulated points) to attain near steady state (within $100q\%$ of μ),

$$n(C, q) = \min\{n \geq 1 : |\mathrm{E}(Y_i) - \mu| < q|\mu| \ \forall\, i > n\},$$

where Y_i is the ith observation *within* a replication. Here, C represents the initial state, which could be a constant or a random variable. This measure indicates the extent of deletion still needed for a particular initialization rule to converge to within a tolerance of steady state.

The first criterion was applied to all models, but the last two were used only for the queueing models with analytically known transients since only in this case did they exhibit behavior that was stable enough to be meaningful. The transient curves (1) usually provided enough information to judge the effectiveness of the initialization rules.

2.2. Analytical Models

Three different $GI/G/s$ queues were used here, with the observed process $\{Y_i\}$ being the delay in queue of the ith arriving customer, arrival rate 1, and traffic intensity $\rho = 0.9$ in all cases. The three particular cases considered were

(1) The $M/M/3$ queue, with steady-state mean delay in queue μ given in Gross and Harris (1974) and transient expected delay in queue $\mathrm{E}(Y_i)$ obtained from Kelton and Law (1985).

(2) The $M/E_4/1$ queue, where E_4 denotes a 4-Erlang distribution, with μ obtained from tables in Hillier and Yu (1981) and transient $\mathrm{E}(Y_i)$ from Kelton (1985).

(3) The $E_4/M/1$ queue, with μ and $\mathrm{E}(Y_i)$ obtained from the same sources as for the $M/E_4/1$ queue.

These are certainly very simple models, but the full analytic information about them allows us to make a more exact analysis of initialization rules, at a reasonable computing cost.

Figure 1 presents the results for the $M/M/3$ queue. The horizontal axis is the customer number, i, and the vertical axis scales the expected delay in queue $\mathrm{E}(Y_i)$ on various curves defined by different initialization methods. The dashed line is at level μ for all figures. The first panel of the figure, labeled D, traces the transient for deterministic rules ranging from empty and idle ($c = 0$ customers present initially) as the bottom curve, to $c = 30$ customers present initially in the top curve. The importance of the initialization is clear, with some states being far preferable to others. The V-shaped curves in Figure 2 show the percent bias (top panel) and warmup period for a 5% tolerance (bottom panel), as functions of the initial number c of customers present, indicating that $c = 14$ is the best deterministic initial state, but that missing this value by very much entails heavy penalties. Panel U of Figure 1 gives the result of applying a discrete uniform initialization for the initial (now stochastic) number of customers present, C. The range for C we took to be $\{0, 1, \ldots, m\}$, and let $m = 10, 20$, and 30. The points on the curve were obtained by conditioning on $C = c$ and then weighting the curves from panel D of the figure by the appropriate uniform p.m.f. $p(c) = 1/(m+1)$, plotting $\sum_{c=0}^{m} \mathrm{E}_c(Y_i)p(c)$ as a function of i, where E_c indicates that the expectation curve from panel D with initial state c is being used. In this case, $m = 20$ appears to be the best choice, but in any case these stochastically initialized runs (with parameters of the initial distribution being arbitrarily specified) appear safer than most of the deterministic initializations. Panel G of Figure 1 is similar to Panel U except that truncated (at $c = 30$) geometric distributions with arbitrarily specified p are used as the initial distribution; we let p take on the values $0.05, 0.10, 0.20, \ldots, 0.90, 0.95$ with a lower value of p producing a higher curve. Again, this stochastic initialization seems more stable than deterministic rules, but the parameter choice appears to have some effect. The final panel of Figure 1, labeled P, indicates the result of using a pilot run (initialized empty and idle, and lasting for 100 delays in queue) to specify m for the discrete uniform initial distribution (top curve) and p for the geometric initialization (bottom curve). These results are partly empirical, in that our procedure was to make a pilot run, observe the maximum number of customers in the system (to specify m) and the time-average number of customers present (specifying ν and thus p), use these initial distributions as in panels U and G to weight the curves from panel D, then repeat this whole process 100 times. The curves in panel P are the result of averaging the resultant 100 transient curves; this procedure mimics the pilot run strategy as it would be applied in practice.

While there is still a transient portion present, its magnitude and duration are considerably milder than most of the deterministic initializations, and it appears somewhat better to let a pilot run determine the initial distributions' parameters than to specify them arbitrarily. The horizontal lines in Figure 2, labeled U and G, represent the percent bias (top panel) and transient duration (bottom panel) when using these pilot run-determined initial distributions; while not as good as the optimal deterministic initialization, these rules appear to provide a safer initialization policy than most deterministic rules. By using a longer pilot run or one initialized in something other than the empty and idle state, the results would be improved.

Figures 3 and 4 consider the $M/E_4/1$ queue with $\rho = 0.9$, and parallel the conditions of Figures 1 and 2 exactly. The same general comments would apply; in fact, the stochastic initialization based on a pilot run appears to be more effective here than for the $M/M/3$ model. Figures 5 and 6 give the results for the $E_4/M/1$ model, again under the same conditions. In this case, the pilot-run scheme appears to have nearly eliminated the initial transient; in terms of the length of the transient (Figure 6, bottom panel), these rules are actually better than the *best* deterministic rule.

These models, with their full analytic information, allow a relatively intensive examination of initialization options. In the following two subsections, we consider two models with less analytic tractability, but which are considerably more complex and realistic.

2.3. An Analytical/Empirical Model

It is generally easier to obtain steady-state means than transient means, and we now consider a closed queueing system in this class. The model is of a time-shared computer system consisting of a single CPU and 20 terminal users, sending jobs for processing by the CPU. The queue discipline for the CPU is round robin, i.e., there is a maximum amount of time (0.1 time unit) that a job is allowed to occupy the CPU continuously, after which it is requeued at the end. When finished, a job returns to its terminal, and the elapsed time from the sending of a job from its terminal to its return is called a *response time*. This terminal then enters a "think state," which has an exponential duration with mean 25. Additionally, there is a swap time of 0.015 required each time a job is removed from the queue and placed in the CPU. The process observed here was Y_i = the response time of the ith job returning to its terminal. The steady-state mean response time μ for this system was derived in Adiri and Avi-Itzhak (1969), to which the reader is referred for more detail. The scalar initial state here is C = the number of jobs present in the computer (CPU plus queue) at time zero, with the assumption that each such job has not visited the CPU at all yet. While μ is known for this system, $\mathrm{E}(Y_i)$ as a function of i is not, so we estimated this transient by averaging across 1500 independent replications of the model for each of the 21 possible deterministic initialization states $c \in \{0, 1, \ldots, 20\}$. Panel D of Figure 7 displays some of these curves, and we again see the consequences of over- or under-initialization on the transient approach to steady state. Panel U displays the result of a uniform initialization between 0 and m for $m = 5$, 10, 15, and 20, and Panel G does likewise for geometric initialization for $p = 1/3$, 1/6, 1/11, and 1/16; these arbitrary stochastic initializations again appear safer then an arbitrary deterministic initialization. Panel P illustrates the use of a pilot run (of length 40 response times, in each case) to determine a uniform or geometric initial state. The "0" indicates that the pilot runs were initialized with no jobs in the computer, and the "H" denotes a pilot run initialized with half (i.e., 10) jobs present in the computer initially. There appears to be much to gain by using such pilot-run-determined initialization distributions, and slightly better results are obtained if these pilot runs themselves are begun in something other than the empty and idle state. We repeated the experiments producing Panel P of Figure 7, except using pilot runs of length 80 response times, twice as long. The results were curves nearly identical to those presented, indicating robustness of the results to the choice of this characteristic of the pilot runs.

2.4. An Actual Manufacturing Model

In order to examine these methods' performance in more realistic settings, and also to see whether practical implementation of the techniques is workable, we examined a complex model of a manufacturing system producing parts supplying the automotive industry. The system consisted of seven operations on some 22 distinct machines in series/parallel configurations, with conveyors of limited capacity providing material handling functions between the operations areas. Incoming parts followed a prescribed route through the system, and the process observed was $Y_i =$ the total time in system of the ith part produced. Neither the steady-state mean nor the transient mean function of this system is known.

In order to establish a benchmark, we made one long run of 1000 observations, and grouped the output into five batches of size 200 each; the resulting batch means were computed, and it appeared from these data as well as from plots of the individual times in system that deletion of the first batch (200 observations) was adequate to remove initial bias. Thus, we retained the last four batches, and used their average as the benchmark for the steady-state mean. For convenience, we scaled this value to 1; with this scaling, the final four retained batch means were 1.07, 0.96, 1.07, and 0.91, indicating that steady-state seems to have been reached.

To see what would happen with the usual empty and idle initialization, we next made 10 independent replications of length 200 parts each, and observed the average time in system, resulting in (after scaling to unity of steady state) ten values between 0.53 and 0.74, with an average of 0.61. This represents a 39% bias with respect to the targeted steady-state mean, reflecting a fairly severe initialization effect.

To implement the pilot-run-determined initialization, using uniform distributions, we observed during the first of the above ten empty and idle initialized runs the maximum number of parts present at several points of congestion in the system, and used these maxima to specify upper bounds for uniform initializations at these points. Note that we are applying the univariate initialization technique repeatedly in the same model, to different components. Ten replications, again of length 200 parts each, were then made, drawing the initial state from these uniform distributions, and the resulting number of parts were placed around the system at time zero. This time the ten replication averages fell between 0.82 and 1.47, with a mean of 1.15. Thus, in comparison with the usual empty and idle initialization, this technique cut the bias by more than half.

3. CONCLUSIONS

It appears that there is much to be gained by investigating the choice of initial conditions for steady-state simulation, and that using stochastic initialization, probably with parameters specified by short pilot runs, can greatly reduce the severity and duration of the biased transient portion of the output. We have demonstrated on a variety of models that the techniques yield desirable performance, and can be used on actual simulations of complex systems. While not eliminating the transient completely, these methods produce results that are much more reliable and stable, and are less sensitive to errors in judgment on the part of the analyst. It is recommended that, particularly for relatively short runs in the context of the method of replication, simulations be initialized stochastically rather than deterministically.

The results in this paper have been limited to models that can be initialized by specification of a scalar. In the manufacturing model of Section 2.4 we simply applied this technique repeatedly (and independently) to the several coordinates of a vector needed to initialize the system state. We are currently investigating ways to account for dependencies between the coordinates of the initial state vector in more complex models.

ACKNOWLEDGMENTS

This research was supported by a contract from the Decision Technologies Division of Electronic Data Systems to The University of Michigan. The input of Ziv Barlach, Sam MacMillan, and Michael Moore, all of DTD/EDS was particularly helpful. The author wishes to thank Joseph Murray of The University of Michigan for his help in the analysis, programming, and writing. Conversations with Emily Roth of GTE Laboratories and Alan Rutan of Raytheon (who suggested the medical analogy in the introduction) were also quite helpful.

REFERENCES

Adiri, I., and B. Avi-Itzhak. 1969. A Time-Sharing Queue with a Finite Number of Customers. *J. Assoc. Comput. Mach.* **16,** 315–323.

Adlakha, V.G., and G.S. Fishman. 1982. Starting and Stopping Rules for Simulations Using a priori Information. *Eur. J. Opnl. Res.* **10,** 379–394.

Gafarian, A.V., C.J. Ancker, Jr., and T. Morisaku. 1978. Evaluation of Commonly Used Rules for Detecting "Steady State" in Computer Simulation. *Naval Res. Logist. Quart.* **25,** 511-529.

Goldsman, D., and L.W. Schruben. 1984. Asymptotic Properties of some Confidence Interval Estimators for Simulation Output. *Mgmt. Sci.* **30,** 1217-1225.

Gross, D., and C.M. Harris. 1974. *Fundamentals of Queueing Theory.* John Wiley & Sons, New York.

Heidelberger, P., and P.D. Welch. 1981. Adaptive Spectral Methods for Simulation Output Analysis. *IBM J. Res. Develop.* **25,** 860–876.

Hillier, F.S., and O.S. Yu. 1981. *Queueing Tables and Graphs.* North Holland, New York.

Jaynes, E.T. 1957. Information Theory and Statistical Mechanics. *Physical Review* **106,** 620–630.

Kelton, W.D. 1985. Transient Exponential-Erlang Queues and Steady-State Simulation. *Comm. Assoc. Comput. Mach.* **28,** 741–749.

Kelton, W.D., and A.M. Law. 1981. A New Approach for Dealing with the Startup Problem in Discrete Event Simulation. *Naval Res. Logist. Quart.* **30,** 641–658.

Kelton, W.D., and A.M. Law. 1985. The Transient Behavior of the $M/M/s$ Queue, with Implications for Steady-State Simulation. *Opns. Res.* **33,** 378–396.

Law, A.M. 1980. Statistical Analysis of the Output Data from Terminating Simulations, *Naval Res. Logist. Quart.* **27,** 131-143.

Law, A.M., and W.D. Kelton. 1982. Confidence Intervals for Steady-State Simulations, II: A Survey of Sequential Procedures. *Mgmt. Sci.* **28,** 550–562.

Law, A.M., and W.D. Kelton. 1984. Confidence Intervals for Steady-State Simulations: I. A Survey of Fixed Sample Size Procedures. *Opns. Res.* **32,** 1221–1239.

Meketon, M.S., and B.W. Schmeiser. 1984. Overlapping Batch Means: Something for Nothing? *Proceedings of the 1984 Winter Simulation Conference,* 227–230.

Murray, J.R., and W.D. Kelton. 1985. The Transient Response of the $M/E_k/2$ Queue and Steady-State Simulation. Technical Report 85-29, Department of Industrial and Operations Engineering, The University of Michigan, 1985.

Schmeiser, B.W. 1982. Batch Size Effects in the Analysis of Simulation Output. *Opns. Res.* **30,** 556–568.

Schriber, T.J., and R.W. Andrews. 1984. An ARMA-Based Confidence Interval for the Analysis of Simulation Output. *Amer. J. Math. and Mgmt. Sci.* **4,** 345–373.

Schruben, L.W. 1981. Control of Initialization Bias in Multivariate Simulation Response. *Comm. Assoc. Comput. Mach.* **24,** 246–252.

Schruben, L.W. 1982. Detecting Initialization Bias in Simulation Output. *Opns. Res.* **30,** 569–590.

Schruben, L.W. 1983. Confidence Interval Estimation Using Standardized Time Series. *Opns. Res.* **31,** 1090–1108.

Schruben, L.W., H. Singh, and L. Tierney. 1983. Optimal Tests for Initialization Bias in Simulation Output. *Opns. Res.* **31,** 1167–1178.

Welch, P.D. 1983. The Statistical Analysis of Simulation Results. In *The Computer Performance Modeling Handbook*, pp. 268–328, S.S. Lavenberg (ed.). Academic Press, New York.

Wilson, J.R., and A.A.B. Pritsker. 1978. Evaluation of Startup Policies in Simulation Experiments. *Simulation* **31,** 79–89.

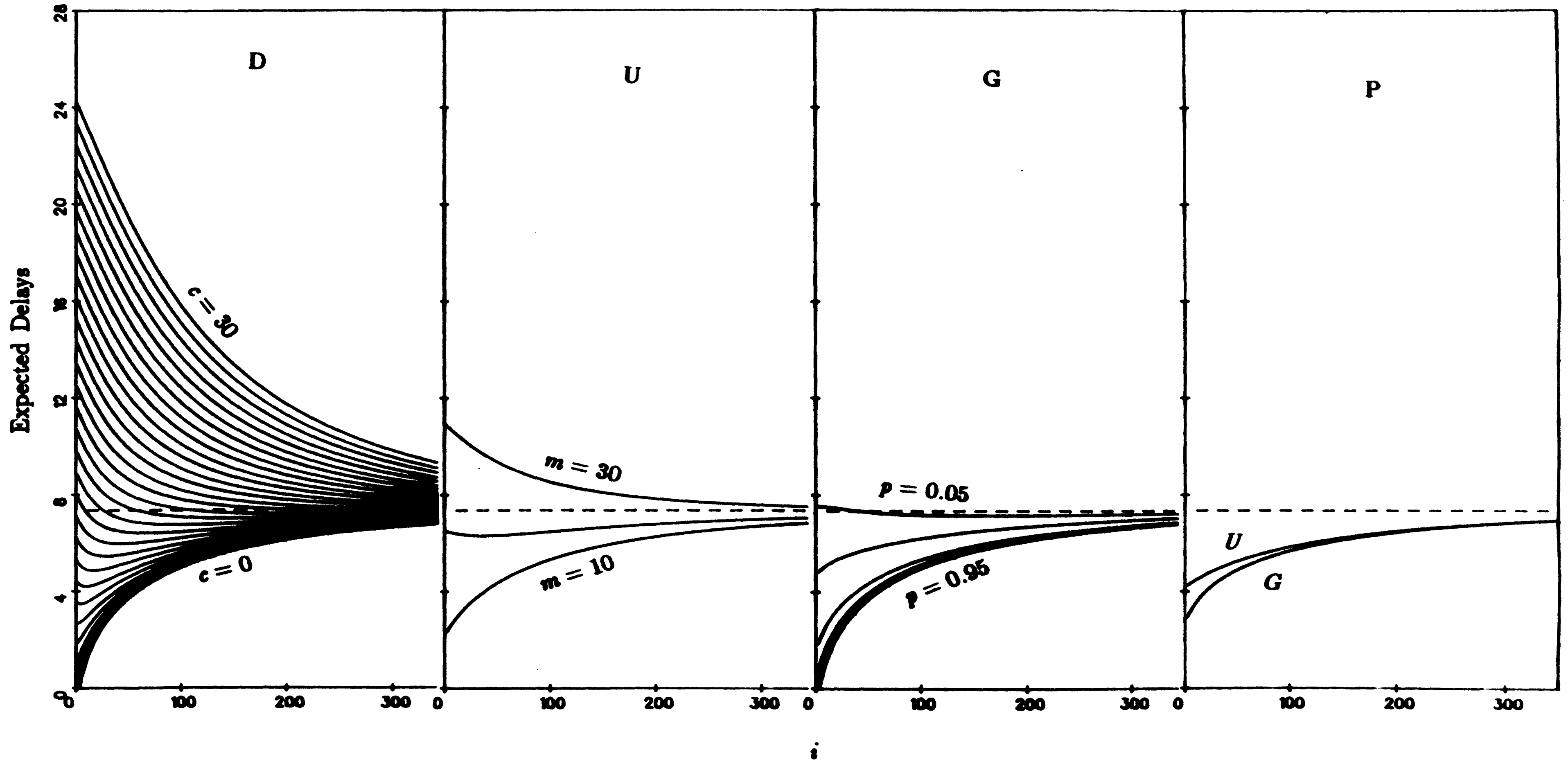

Figure 1. Expected Delays in Queue for the $M/M/3$ Queue, $\rho = 0.9$.

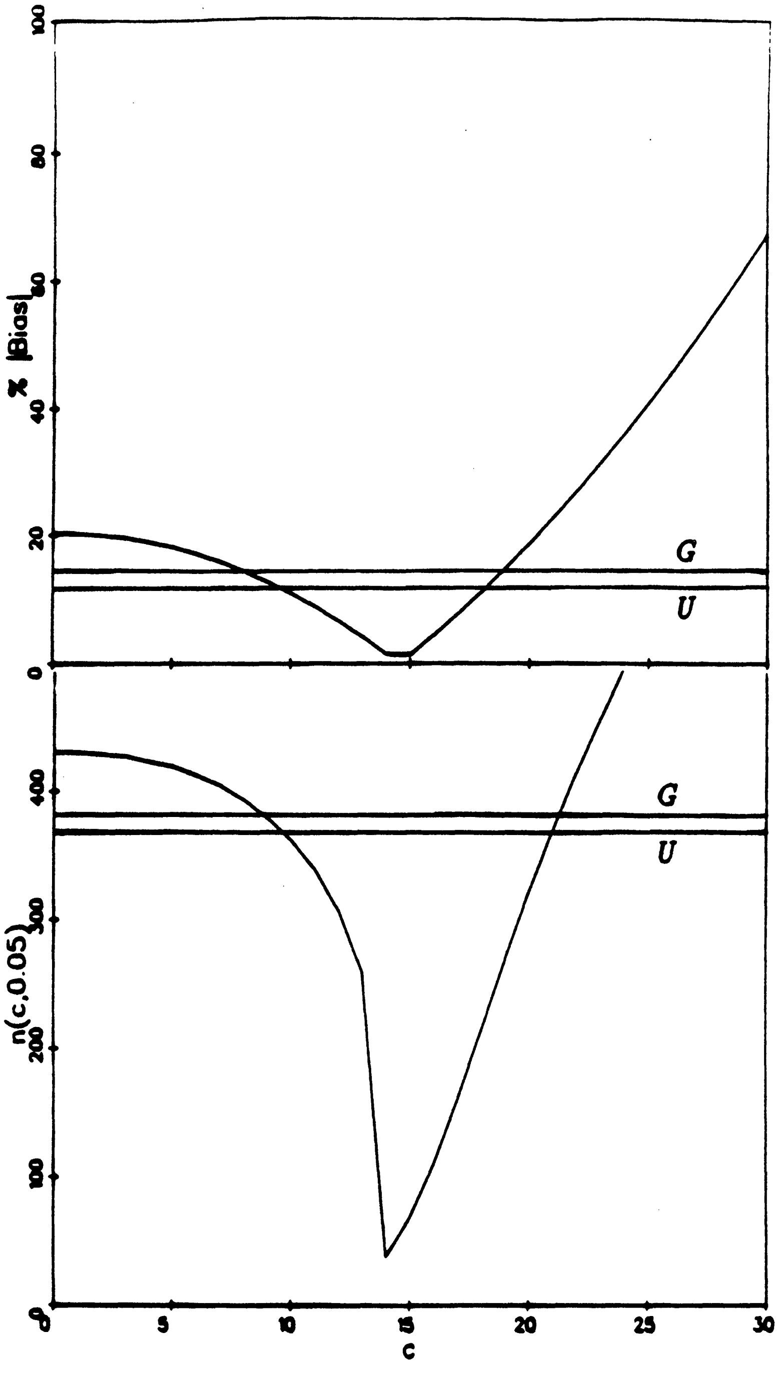

Figure 2. Bias and Transient Length for the $M/M/3$ Queue, $\rho = 0.9$.

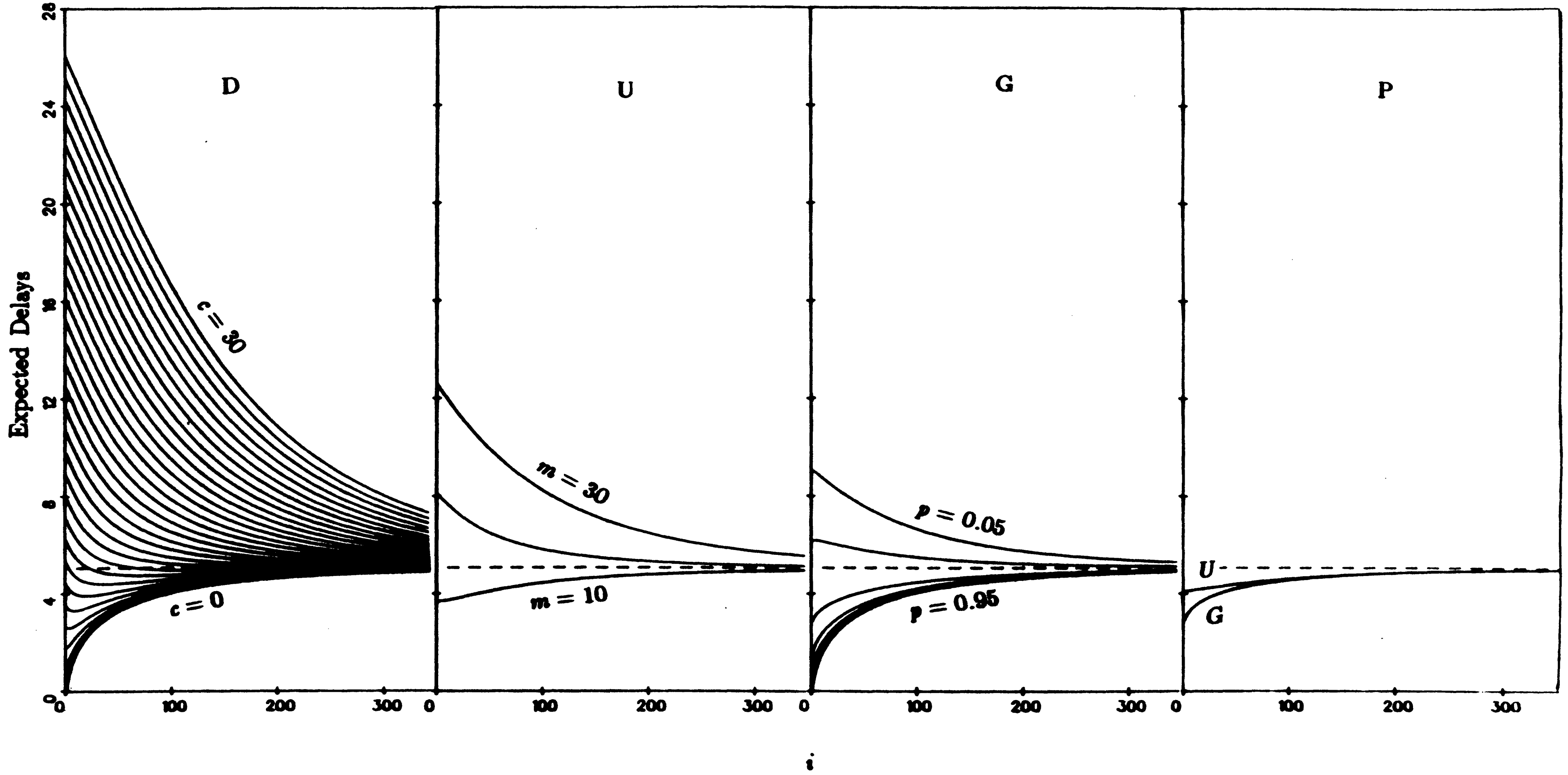

Figure 3. Expected Delays in Queue for the $M/E_4/1$ Queue, $\rho = 0.9$.

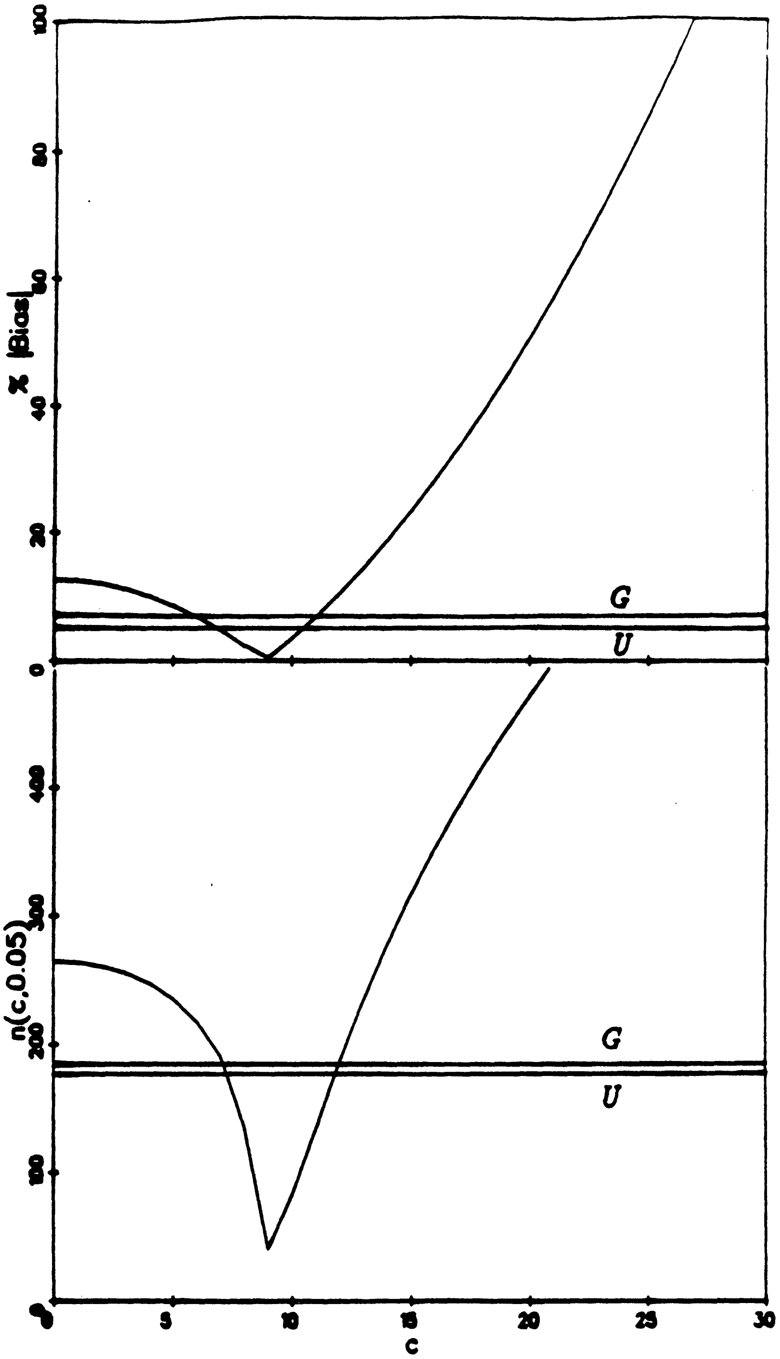

Figure 4. Bias and Transient Length for the $M/E_4/1$ Queue, $\rho = 0.9$.

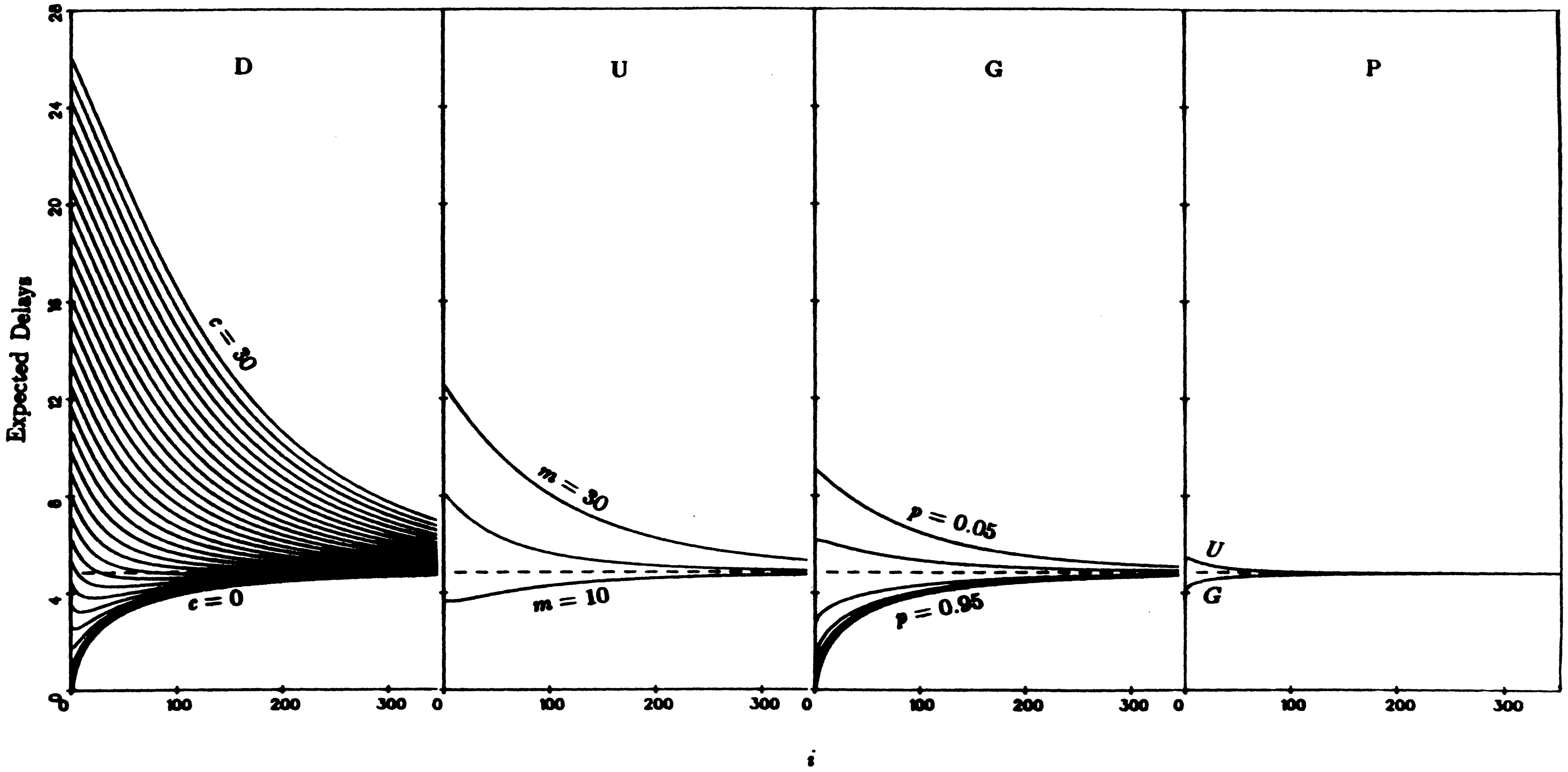

Figure 5. Expected Delays in Queue for the $E_4/M/1$ Queue, $\rho = 0.9$.

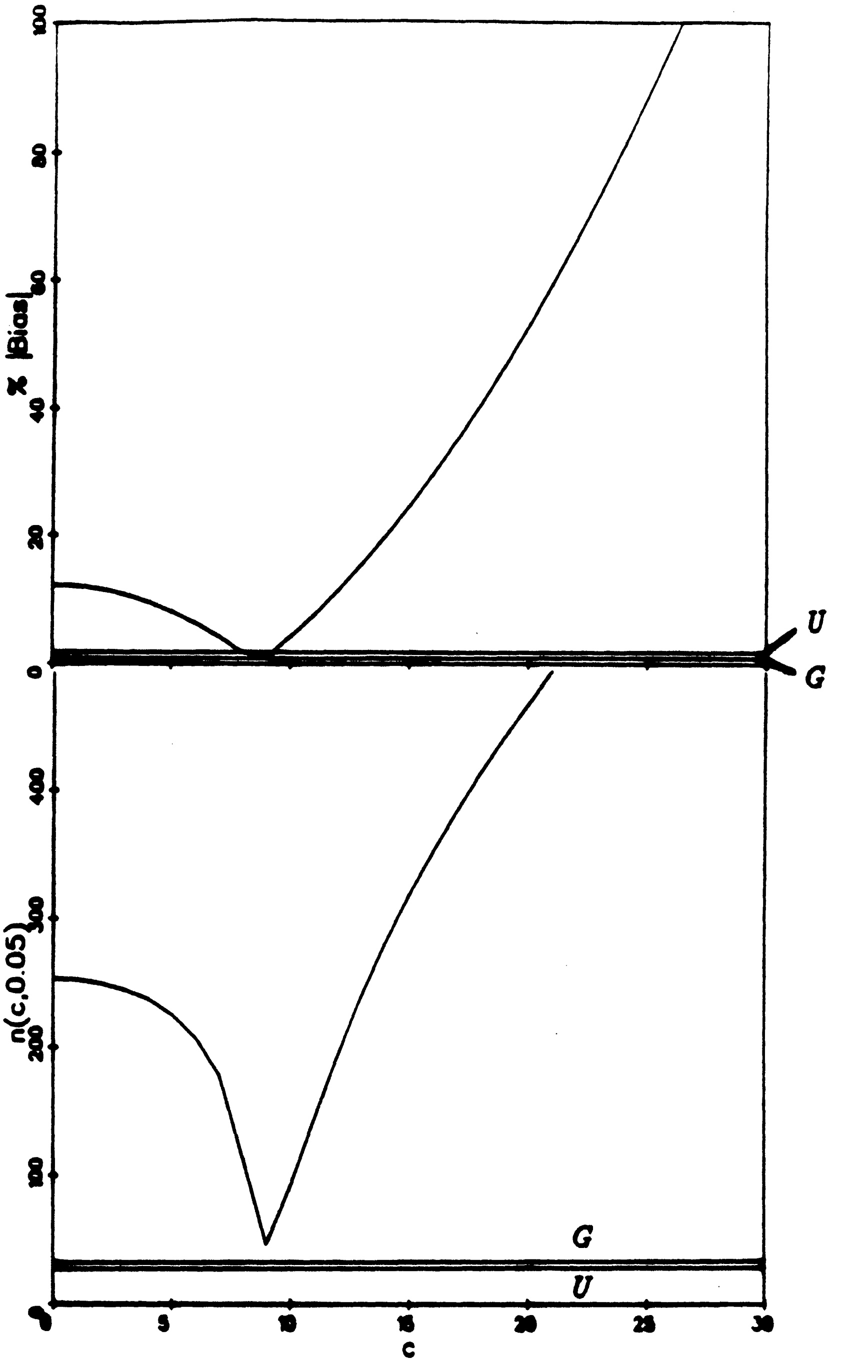

Figure 6. Bias and Transient Length for the $E_4/M/1$ Queue, $\rho = 0.9$.

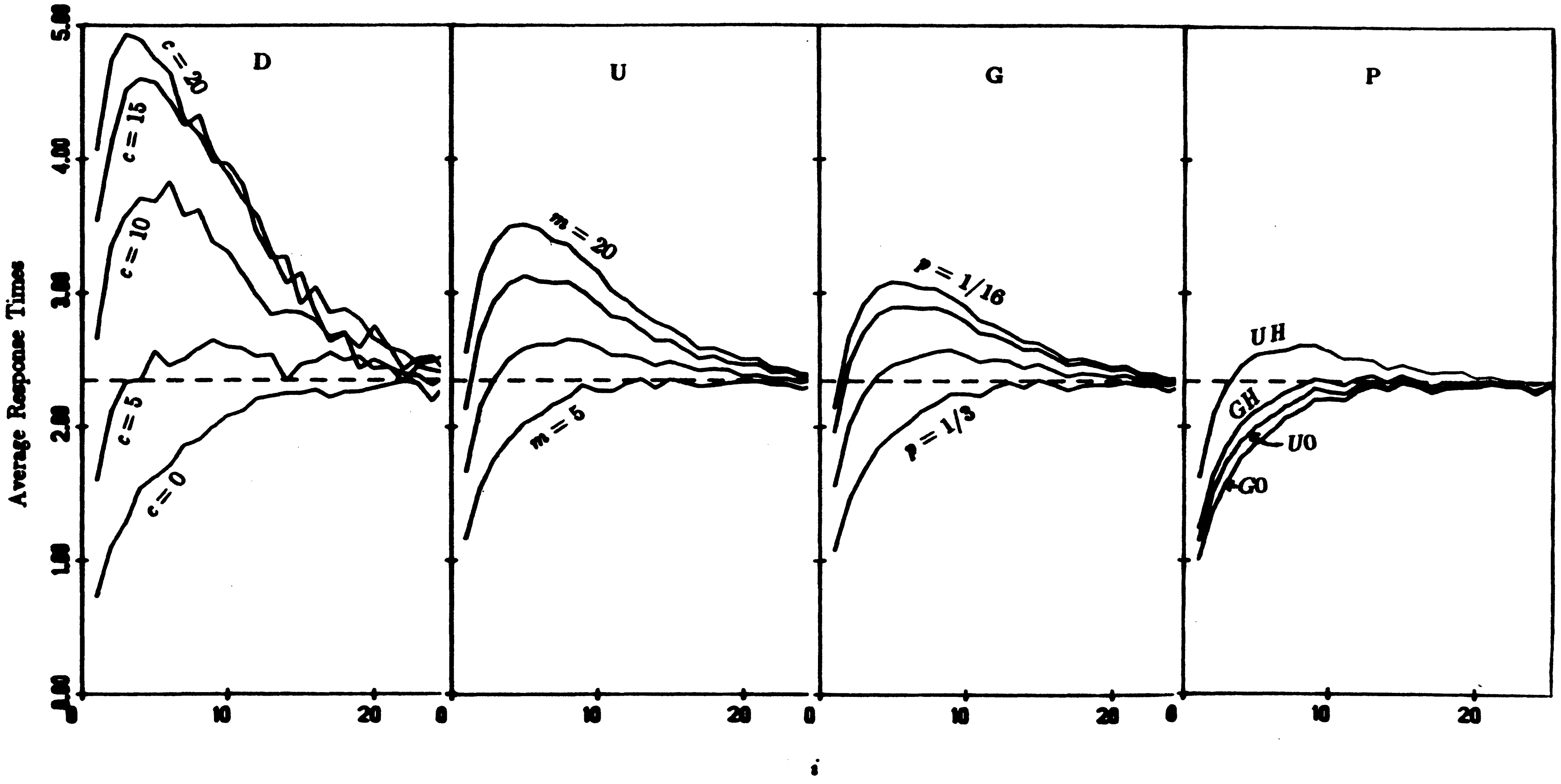

Figure 7. Average Response Times for the Time-Shared Computer Model.