Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/5521--98-8165

Ordinal Optimization of Communication Network Performance: A New Look Based on use of the Connection Machine

JEFFREY E. WIESELTHIER
CRAIG M. BARNHART

Communication Systems Branch
Information Technology Division

ANTHONY EPHREMIDES

Kaman Sciences Corp.
Alexandria, Virginia
and
University of Maryland
College Park, Maryland

August 3, 1998

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)

2. REPORT DATE
August 3, 1998

3. REPORT TYPE AND DATES COVERED
Interim Report 6/94-12/96

4. TITLE AND SUBTITLE
Ordinal Optimization of Communication Network Performance:
A New Look Based on use of the Connection Machine

5. FUNDING NUMBERS
PE - 61153N
PR - RR015-09-41
WU-DN159-036

6. AUTHOR(S)
Jeffrey E. Wieselthier, Craig M. Barnhart,* and Anthony Ephremides**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/MR/5521--98-8165

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
*Formerly at NRL, currently at TRW Data Technologies Division, Aurora, CO
**University of Maryland and Kaman Sciences Corporation

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Ordinal optimization can be an effective technique for efficiently finding nearly optimal solutions to complex problems. The motivation behind this approach is that finding the optimal solution (or control policy) is often too costly or time consuming, although a suboptimal solution may provide sufficiently good performance. In earlier studies on sequential machines, we demonstrated the effectiveness of ordinal optimization based on the Standard Clock (SC) parallel simulation technique. In this report we study the use of SC and ordinal optimization techniques on the massively parallel Connection Machine CM-5E.

The use of the CM-5E has greatly extended the size of problems that can be addressed. For example, whereas our studies on sequential machines were typically limited to wireless networking examples with up to 8 transceivers per node, the use of the CM-5E has permitted the study of examples with up to 4,000 transceivers per node, thus permitting the study of examples with the dimensions of high-speed networks. We address self-regulation and scalability properties of the solutions, as well as the determination of good solutions for large, finely-quantized systems. The CM-5E has enabled us to demonstrate that good solutions can be found quickly, and often without the need for high-performance computer resources.

14. SUBJECT TERMS
Ordinal optimization
Connection Machine
Standard clock
Parallel simulation
Communication network

15. NUMBER OF PAGES
57

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

CONTENTS

ORDINAL OPTIMIZATION OF COMMUNICATION NETWORK PERFORMANCE:
A New Look Based on Use of the Connection Machine

1 INTRODUCTION

Optimization problems in communication networks can be very difficult because accurate analytical models are often either unavailable or too complex to be evaluated within realistic time constraints. Although in some cases models of reasonable complexity may be available to evaluate system performance under a given set of parameters, the optimization process typically involves the evaluation of system performance under many different sets of system parameters (or, equivalently, for different control policies). The goal here is to find the control policy that provides the best performance. Thus the size of problems that can be addressed by analytical and numerical techniques is generally severely limited.

Because of the unavailability of suitable analytical techniques for a wide variety of performance evaluation and optimization problems in communication networks and other examples of discrete-event dynamic systems (DEDS), simulation is the primary method for their solution. In addition, simulation may reveal important conceptual aspects of a problem that cannot be identified a priori. However, although in many cases simulation can provide a good estimate of system performance, it can also be extremely time consuming, and therefore expensive. This is especially true when the optimization process requires the evaluation of system performance for a large number of policies, as we noted above.

Recent research in the area of DEDS has resulted in the development of a number of approaches that greatly improve the efficiency of the simulation process. For example, the Standard Clock (SC) technique[1] [3, 4, 5] "parallelizes" simulation by passing a common event stream to a large number of problem instances, each of which updates its state subject to its unique control policy. To determine the optimal control policy, performance would be evaluated for a (possibly) exhaustive set of control parameters. However, the accurate performance evaluation of a large number of policies remains a daunting task, even with the improvement in simulation efficiency provided by SC techniques, because long simulations are generally needed to provide accurate estimates of system performance.

[1] Although the basic SC technique is applicable only to systems with exponential interevent times, a number of techniques have been developed to incorporate non-exponential techniques as well. For example, in [1, 2] we showed how SC techniques can be applied to integrated voice/data networks with fixed-length data packets.

Manuscript approved April 7, 1998.

Recently, Ho et al. [6] suggested that, in many applications involving DEDS, the goal of finding the optimal solution could be relaxed to that of finding a sufficiently "good" solution. This approach has been termed "ordinal optimization," where *ordinal* refers to the determination of policies that perform relatively well compared to other candidate policies, without necessarily obtaining accurate estimates of the performance values associated with these policies. Examples of ordinal-optimization approaches include the use of short simulation runs, crude analytical models, or simplified simulation models. When carefully formulated, one or more of these approaches may be able to provide good estimates of the ranking of performance under a large number of alternative sets of control parameters, even though the estimates of actual system performance may be highly inaccurate. The improvement in efficiency (reduction in computer time) can be several orders of magnitude if one is satisfied with finding a good solution, rather than the (typically elusive) optimal solution.

In earlier studies [1, 2] we demonstrated the improvement in efficiency that can be achieved by using the SC approach for the simulation of several examples of DEDS, including integrated voice/data wireless networks. Most importantly, we demonstrated the effectiveness of the combined use of SC and ordinal-optimization techniques for the solution of admission-control problems. Although the SC approach is ideally suited to parallel machines, our studies in [1, 2] demonstrated that considerable improvement in efficiency can also be achieved on sequential computers.

In the present report we discuss the use of SC and ordinal-optimization techniques on the Thinking Machines Corp. Connection Machine CM-5E, which is a parallel machine. To make this report self-contained, we provide background discussions of SC simulation, the voice admission-control problem in circuit-switched wireless networks, and ordinal optimization. We demonstrate that the SC simulation technique has excellent scalability properties on the CM-5E, a crucial property that has enabled the evaluation of considerably larger problems than those that could be addressed by conventional sequential machines. The capability to rapidly examine a large number of control policies permits much more-thorough examination of the state space than is possible with sequential machines; our evaluation of admission-control policies in high capacity networks has helped to confirm our conjecture that voice-call blocking probability is a unimodal function of threshold admission-control policies. The examination of larger problems on the CM-5E has also helped refine our conjecture on "network self regulation" [7, 8] and on the capability of simple control policies to perform well [1, 2]. The scaling capability is also essential in extending our work to broadband networks. Moreover, the use of the CM-5E has provided us with substantial insights on network behavior that have influenced the formulation of our algorithms and our methods.

1.1 Outline of the Report

In Section 2 we review the principles of Standard Clock simulation, and discuss its implementation on sequential and parallel machines. In Section 3 we review the admission-control problem in circuit-switched multihop networks, and discuss the product-form solution that characterizes equilibrium system performance. This is the primary problem that we have approached by means of SC and ordinal-optimization techniques. In Section 4 we show how the

SC simulation model can be applied to circuit-switched multihop networks, thereby permitting the simultaneous generation of a large number of sample paths that are driven by the same event sequence.

In Section 5 we discuss the evaluation of circuit-switched wireless networks for an example that is sufficiently small to be evaluated by numerical techniques. We remark that the network appears to be "self-regulating" in the sense that little improvement is achieved by imposing active admission-control, as compared to the uncontrolled system in which all calls are accepted provided that sufficient network resources are available. In Section 6 we address the question of self-regulation in more detail, first for small examples evaluated on sequential computers, and then for considerably larger examples for which the CM-5E was needed.

In Section 7 we review the principles of ordinal optimization, and we present results for relatively small problems (eight transceivers per node) obtained on sequential machines. Then, in Section 8 we address considerably larger problems (up to 4000 transceivers per node) for which the CM-5E was used. Finally, in Section 9 we present a summary and conclusions drawn from this research.

2 THE STANDARD CLOCK APPROACH TO SIMULATION

We begin by reviewing the principles of SC simulation [3, 4, 5], which permits the simultaneous evaluation of system performance under a large number of control policies. In Section 2.1 we discuss SC implementation on sequential machines, and in Section 2.2 we discuss the case of parallel machines.

2.1 Standard Clock Simulation on Sequential Machines

The principles of SC simulation on sequential machines and the performance improvement that can be achieved, as compared to Brute Force simulation, were discussed in [1, 2]. In this section we summarize the major aspects of the SC methodology as implemented on sequential machines.

We use the M/M/1/K queue paradigm to explain the SC technique. An M/M/1/K queue is a single-server queue with finite buffer capacity K (including the packet in service), Poisson arrival process at rate λ, and exponentially distributed service of expected duration $1/\mu$. We use the following notation to describe the state of the queueing system:

x = the number of packets in the system,

$\pi(x)$ = the steady-state probability that there are x packets in the system.

The discrete parameter K in the M/M/1/K queue makes this system a particularly good candidate for SC simulation, which works equally well for continuous or discrete variables. Using SC techniques, we have simulated system performance for many values of K simultaneously. Typically, in traditional simulations of M/M/1/K queues, an event calendar is used to schedule events (i.e., arrivals and departures) that are generated by using a distinct distribution for each event type. By contrast, in SC simulations a single sequence of random numbers is generated from an exponential distribution with parameter $\Lambda = \lambda + \mu$. Each of the variates in this sequence represents a generic interevent time. The type of event associated with each interevent time is determined by drawing an independent random number U from a uniform distribution on the interval [0,1]; the event is an arrival if $U \leq \lambda/\Lambda$, otherwise it is a departure. Generation of events at this rate Λ, which is referred to as the *maximal rate*, is an example of the well-known technique of *uniformization* [9]. A "ratio yardstick" showing the event-type determination process is shown in Fig. 2.1.[2]

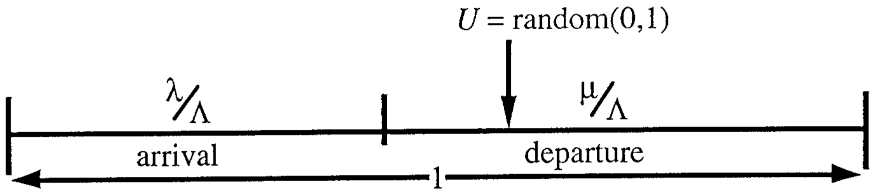


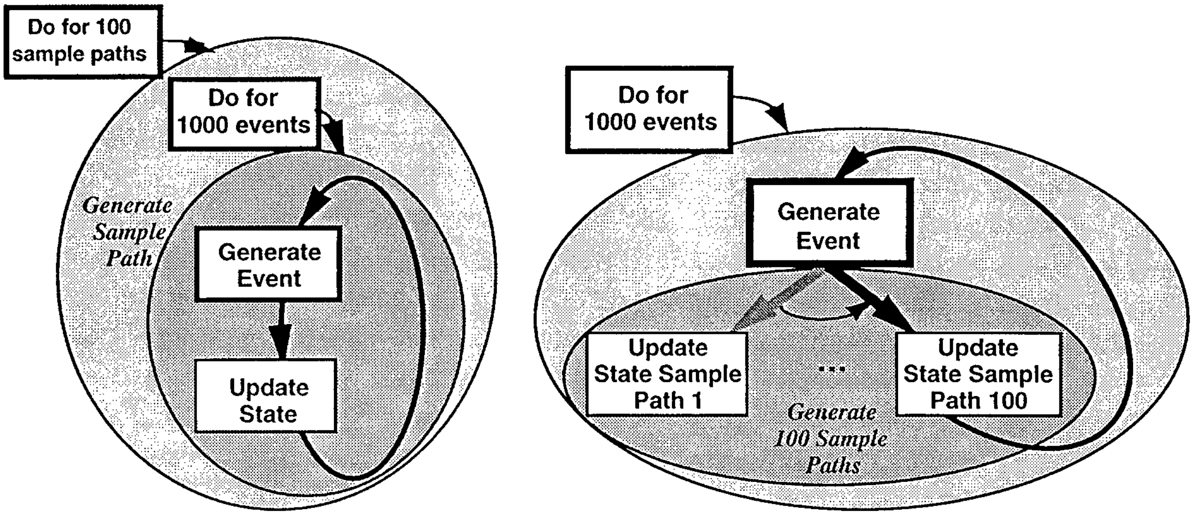
Fig. 2.1 — Ratio yardstick for the M/M/1/K simulations

Thus two random numbers must be generated to produce each event; one to specify the timing of the next event, and the other to specify its type. It is possible that an event determined in this manner turns out to be infeasible (e.g., a departure from an empty system). The interevent time of such a "fictitious" event is used to update the system time as if the event were "real" and did in fact occur, but no state change occurs (the fictitious event is discarded).

The improved efficiency of the SC method is achieved by using the resulting sequence of (interevent time, event type) pairs, known as the *clock sequence*, to simultaneously generate sample paths for a number of structurally similar, but parametrically different, systems. In particular, a single clock sequence can be used to generate N sample paths in parallel for N M/M/1/K queues, each with a different value of K.

Figure 2.2 shows a simplified comparison of SC and conventional, or *Brute-Force* (BF), simulation for the case of 100 sample paths ($N = 100$) and a simulation duration of 1000 events, where a sequential machine is used in both cases. The program structure is shown as nested "Do loops." In BF simulation the "Do 100 sample paths" is the outer loop, and within this loop the "Do 1,000 events" loop actually constructs each of the sample paths. With this loop structure, 100,000 events and 100,000 state updates are required. In contrast, with SC simulation the order of the "Do loops" is reversed. As a consequence of this reordering, the "Generate Event" procedure is outside the inner loop, and each event generated is passed to all 100 sample paths. Thus, with SC simulation only 1,000 events need to be generated, although we still have to perform the 100,000 state updates as in the BF case.

[2] The ratio yardstick is easily extended to complex examples with many different types of events. As in the present example, events are generated at the maximal rate (which is the sum of all exponential interevent rates in the system), and each event corresponds to a region on the yardstick, the width of which is equal to the probability of the corresponding event. The event type is determined by the region into which a random number (drawn from a uniform distribution on [0,1]) falls. The computational effort involved in determining the event type by means of the alias method [10] is independent of the number of event types, thus making this method suitable for complex systems.



(a) BF (b) SC

Fig. 2.2 — A comparison of program structure in BF and SC simulations

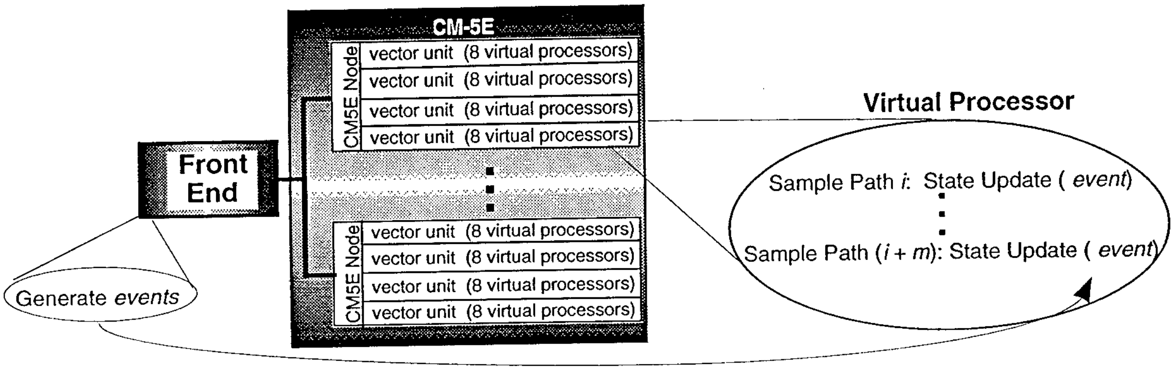
In the M/M/1/K example, arrival events are always feasible for all values of K; however, an arrival when the buffer is full (i.e., when $x = K$) does not lead to a change of state because that arrival is blocked and rejected. Similarly, a fictitious event (e.g., a departure when $x = 0$ in this system) leaves the state unchanged. Thus a different trajectory may be produced in each of these experiments, even though the clock sequence (and thus each event) is the same for all. Since each element of the clock sequence is used N times, the number of events that must be generated is reduced by a factor of N. This reduction has a dramatic effect on the overall simulation time because the generation of events is considerably more time consuming than the consequent updating of system state, as our simulation timing results showed in [1, 2]. Typically, in our studies of the parallel simulation of up to 10,000 sample paths (i.e., K was varied from 10 to 10,000 in steps of 1), simulation was sped up by a factor of between 3 and 5, depending on the machine being used. The effect of this speedup is further enhanced by the ordinal optimization techniques that are discussed in Sections 7 and 8; the correlation introduced into sample paths that are driven by the same event sequence permits a relatively accurate ranking of policies (from best to worst) on the basis of relatively short simulation runs.

Unfortunately, the speedup achieved by using the same event sequence for all N sample paths can be reduced by the occurrence of fictitious events; thus, additional events must be generated to observe the specified number of real events. However, in [1, 2] we demonstrated that significant speedup can be obtained even when the fraction of fictitious events is high. Another limitation of this approach is that it is normally restricted to systems with exponentially distributed interarrival times, although it has recently been shown that some deterministic events [11] or other nonexponentially distributed events [12] can also be incorporated into the model and, in fact, we demonstrated in [1, 2] how fixed-length data packets can be incorporated into SC simulation models.[3]

[3] The data-packet departure events are deterministic, when conditioned on the data-packet arrival times and on the voice state.

2.2 Standard Clock Simulation on Parallel Machines

Although the example presented in Section 2.1 is based on the use of a sequential machine to perform SC simulation, it can readily be seen that the SC technique is extremely well suited for implementation on parallel machines such as the Connection Machine CM-5E. As is shown in Fig. 2.3, the connection machine front end generates the events, each of which is passed to all of the parallel processors, each of which is responsible for the updating of one or more sample paths. Each of the 256 nodes on NRL's CM-5E consists of four vector units, each of which in turn consist of 8 virtual processors. Thus the CM-5E provides 8192 virtual processors, which operate in parallel on the data generated by the front end. Each of the virtual processors can be programmed to generate *m* sample paths, thereby resulting in the parallel generation of 8192 *m* sample paths, each of which is driven by the same sequence of events.


CM-5E
CM5E Node
vector unit (8 virtual processors)
vector unit (8 virtual processors)
vector unit (8 virtual processors)
vector unit (8 virtual processors)
CM5E Node
vector unit (8 virtual processors)
vector unit (8 virtual processors)
vector unit (8 virtual processors)
vector unit (8 virtual processors)
Front End
Generate *events*
Virtual Processor
Sample Path *i*: State Update (*event*)
Sample Path (*i* + *m*): State Update (*event*)

256 nodes ⇒ 1024 vector units ⇒ 8192 virtual processors

⇒ 8192 • *m* sample paths simulated simultaneously

Fig. 2.3 — Standard Clock Simulation on the CM-5E

The CM-5E can be programmed as either a Single-Instruction Multiple Data (SIMD) or a Multiple-Instruction Multiple Data (MIMD) machine. Our approach, as described above, has been to exploit the "data-parallelism" offered by the C* programming language, which amounts to using the machine as a SIMD one. A speedup of approximately 100 has been observed, as compared to a Sparc-10 workstation.

An alternative approach, in which the CM-5E is used as a MIMD machine, would involve "message passing," which essentially uses each of the 256 processors as a standalone machine. The CM-5E provides a very fast communication channel between the machines. Use of this MIMD approach would reduce the dependence on the front end (actually, it probably virtually eliminates the front end, or more precisely, it treats the front end as just another processor) as compared with our SIMD approach. Our impression is that such an approach would fail to exploit the vector units, and would tend to be bound by communication overhead.

2.2.1 Algorithmic Insights Gained from use of the CM-5E

In [1, 2] we discussed the application of SC and ordinal-optimization techniques to integrated voice/data networks. In our integrated network model, voice traffic performance is independent of data traffic because data uses the time-varying residual capacity not being used by voice. The objective is to choose the voice admission-control policy that minimizes data-packet delay, subject to a constraint on voice-traffic blocking probability. Most of the work discussed in [1, 2] was based on the use of sequential machines, although we have done some studies of integrated networks on the CM-5E as well. These studies revealed a bottleneck that resulted from the processing of deterministic events, as described below.

In an integrated voice/data network with fixed length data packets, voice-call arrivals and departures and data-packet arrivals are stochastic exponentially distributed events, which are easily incorporated in the Standard Clock paradigm. However, because the data packets are of fixed length, the time required to transmit a packet is fixed; hence, the data-packet departure events are deterministic (given the stochastic events in the system). In the "Direct-Processing" approach we developed for handling the deterministic data-packet departures [1, 2], before processing the next stochastic event, each vector unit (where, loosely speaking, a vector unit corresponds to a particular voice admission-control policy) must have completely processed all of the data-packet departure events in its deterministic event queue. Thus, all processors experience the delay associated with the processing time of the vector unit with the longest deterministic event queue. In the "Improved-Processing" approach, we take advantage of the constant data-service rate between voice events, and the PASTA (Poisson Arrivals See Time Averages) theorem [13], [14] to periodically reconstruct the data departure sequence at convenient times in the simulation (see [1, 2] for details). We found that this algorithmic change gave about an order of magnitude speedup on both the CM-5E and on sequential machines.

3 ADMISSION CONTROL IN CIRCUIT-SWITCHED MULTIHOP NETWORKS

In this section we describe the problem of admission control in circuit-switched multihop communication networks. The use of standard clock simulation and ordinal optimization have been powerful tools for the solution of this problem, and the use of the CM-5E has significantly enhanced the size of problems that can be addressed, while providing added insight into the properties of good solutions.

The requirement of low-variance, short delay for voice service in communication networks motivates the customary approach of establishing circuit-switched paths (i.e., either true circuits or virtual circuits) between communicating nodes for the duration of each voice call. Because the source and the destination nodes in a wireless network are not generally within direct communication range, relaying over multihop paths may be required. We assume that unless voice calls are accepted for immediate transmission (in practical terms, this may mean within several hundred ms of their arrival), they are "blocked" and lost from the system, a mode of operation generally referred to as "blocked calls cleared." Appropriate performance measures for this mode of operation include blocking probability and throughput.

Network performance can be improved by administering controls in the form of call-admission policies, as well as by routing and by link-activation scheduling [15, 16]. These are complex, interdependent problems that must be addressed jointly to determine the form of overall "optimal" network operation. Although we recognize the interdependence of these problems, our focus in this report is on the voice-call admission-control problem. We eliminate the need to solve the routing problem by assuming, as is customarily done, that fixed multihop paths between source-destination pairs are used. Similarly, we eliminate the need to address the link-activation scheduling problem by assuming the use of channelization in the frequency domain (by means of frequency-division multiple access—FDMA).

Although a great deal of attention has been paid to the modeling and performance evaluation of circuit-switched voice in communication networks [17, 18], relatively little has been done until recently in the area of optimal voice-call admission control in wireless circuit-switched networks. An uncontrolled mode of operation is typically assumed, in which all voice calls are admitted as long as sufficient network resources (e.g., link bandwidth) are available. In a controlled system, it may be advantageous to block a call even though resources are currently available because the acceptance of a particular call now may result in the blockage of several other future calls (or perhaps a call of higher precedence) that could otherwise have been accepted. Thus it may be possible to improve performance by administering "active" admission control for voice traffic.

In [7, 8] we showed how the recently introduced methodology of multiple-service, multiple-resource (MSMR) modeling, in conjunction with the use of "coordinate-convex" policies [19, 20], can be used to study voice admission control in radio environments. Search techniques were developed to speed up the determination of optimal admission-control policies. Although these approaches are considerably more efficient than exhaustive search, they are computationally intensive, thus limiting the size of problems that can be studied. In [1, 2] we developed ordinal-optimization techniques, used in conjunction with Standard Clock simulation techniques. Using this approach, nearly optimal solutions can be determined rapidly, thus greatly extending the size of the admission-control problems that can be studied. In the present report we demonstrate how the size of problems that can be studied can be extended considerably further by using a parallel computer, namely the Thinking Machines Corp. Connection Machine CM-5E.

3.1 The Circuit-Switched Model

We consider a wireless network in which FDMA is used to provide contention-free channel access to a multihop circuit over a predetermined path between the source and destination nodes throughout the duration of each accepted voice call. Network topology can be described in terms of the communication resources available at each node and the connectivities between nodes. To illustrate our problem formulation, we consider the simple seven-node star-network example shown in Fig. 3.1. Nodes 2 – 7 are each connected to only node 1, thus necessitating the use of node 1 as a relay in all multihop circuits. We consider three source-destination pairs [(2,5), (3,6), and (4,7)], which correspond to circuits 1, 2, and 3, respectively.

The state of the system is defined to be $\boldsymbol{x} = \{x_1, x_2, x_3\}$, where x_j is the number of calls currently active over circuit j.

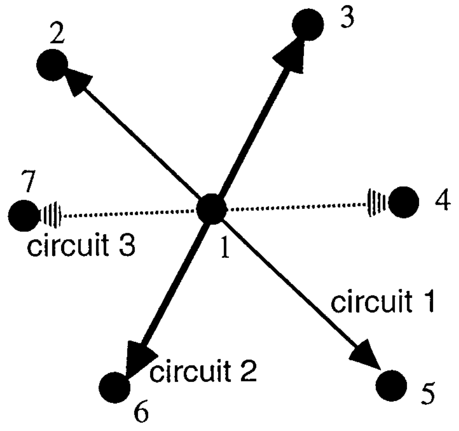


Fig. 3.1 — A simple 3-circuit network

An uncontrolled system operates in a mode known as "complete sharing," in which a voice call is accepted as long as there are sufficient resources at all nodes along the multihop path; if node i has T_i transceivers, it can support up to T_i simultaneous calls.[4] We refer to the limits imposed by the values of T_i as the "capacity constraints." For example, if each of the nodes in the network of Fig. 3.1 has five transceivers, the resulting uncontrolled system state space Ω is as shown in Fig. 3.2. Equivalently, the capacity constraints expressed as inequalities describe the system state space:

$x_1 \leq 5;\ x_2 \leq 5;\ x_3 \leq 5;$ (i.e., no more than 5 calls may be accepted on any circuit)

$x_1 + x_2 + x_3 \leq 5.$ (the hub, node 1, can handle no more than 5 calls)

In general, a vector description of circuit j in terms of the nodes it traverses is given by

$$c_j = \{c_{j1}, c_{j2}, c_{j3}, \ldots, c_{jN}\}, \tag{3.1}$$

where

$$c_{ji} = \begin{cases} 1, & \text{if circuit } j \text{ traverses node } i \\ 0, & \text{otherwise} \end{cases}, \tag{3.2}$$

and N is the number of nodes in the network. For example, circuit 1 in Fig. 3.1 is represented as

$$c_1 = \{1, 1, 0, 0, 1, 0, 0\}.$$

[4] Other models are certainly possible. For example, if traffic is one way, rather than interactive, the source node would not have to dedicate a receiver to support the call. Similarly, receive-only nodes would not need transmitters. The approach presented here can be modified straightforwardly to accommodate variations such as these. The implicit assumption that the number of transmitters and receivers at a node are equal, which lets us describe the node in terms of a single parameter T_i, is used here because of convenience of notation, and can easily be relaxed.

Now we can express the capacity constraints as

$$\sum_{j=1}^{J} x_j c_{ji} \leq T_i, \qquad i = 1, \cdots, N \tag{3.3}$$

where J is the number of circuits in the network.

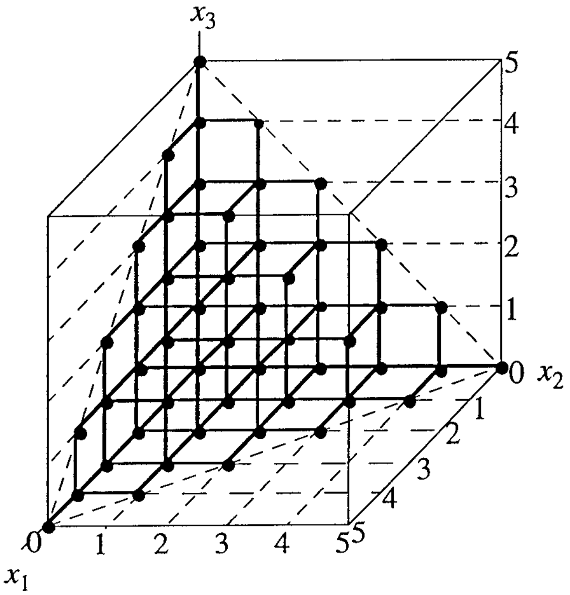


Fig. 3.2 — The admissible state space Ω for the network of Fig. 3.1

A similar equation can be written for wired networks, in which case c_{ji} would take on the value of 1 if and only if circuit j traverses link (rather than node) i. The capacity constraints would be again written in terms of T_i, which would be now interpreted as the number of calls that can be supported by link i. Some of the differences between wireless and wired networks, and their impact on network operation and performance evaluation, are discussed in [7, 8] and [21, 22, 23].

In this report we do not address the protocol issues that are associated with call setup, such as the control messages that must be exchanged to disseminate the network state to all nodes. Our focus is on the development of a mathematical system model that demonstrates the performance improvement that can be achieved through the use of admission control.

3.2 Control Policy

Our ultimate goal is to achieve optimal network performance by exercising an admission-control policy on call requests. In our studies of voice-only networks we use the criterion of blocking probability; for integrated networks it is customary to use the weighted sum of blocking probability and expected data-packet delay, or simply data-packet delay subject to a maximum allowable blocking probability. In practice, the true optimal solution is usually elusive, and we must settle for a good suboptimal solution.

We assume that a central controller makes the decisions on whether or not to accept calls based on perfect knowledge of the number of calls of each type that are currently in progress (i.e., the system state $\boldsymbol{x}$), and hence on the set of resources that are available for new calls. The transceivers needed to establish a circuit are acquired simultaneously when the call arrives, and are released simultaneously when the call is completed. Calls are blocked when one or more nodes along the path do not have a transceiver available, or when a decision is made not to accept a call despite the availability of transceivers. Blocked calls are lost from the system. These assumptions, coupled with use of the class of coordinate-convex admission-control policies discussed below, lead to a mathematically tractable description of system performance.

In [20], admission-control policies are divided into five classes: complete sharing, complete partitioning, trunk reservations, coordinate convex, and dynamic programming. Complete sharing is the simplest form of admission control because no active control is used, i.e., voice calls are admitted as long as resources are available at all nodes along the path. Complete partitioning, under which fixed amounts of resources are allocated a priori exclusively to particular call types, is the most restrictive form of control; it may provide the least efficient use of network resources, but it easily provides a guaranteed level of service to each call class. Trunk reservations provide a compromise between complete sharing and complete partitioning; all primary routed calls are accepted (provided that resources are available), but alternate-routed calls are accepted only if some threshold level of resources are available. Coordinate-convex and dynamic-programming control policies, which are discussed below, both use a modified form of complete sharing within a state space that is a restriction (a subspace) of the uncontrolled state space. In this report, we focus on the class of coordinate-convex control policies because they provide a form of intelligent resource sharing without the complexity of dynamic programming.

The capacity constraints limit the state space Ω in which $\boldsymbol{x}$ is allowed to take values. We assume that the state space is coordinate convex [24], i.e., in a system with J circuits, if $\boldsymbol{x}$ is an admissible state ($\boldsymbol{x} \in \Omega$) and $x_j \geq 1$, then $\boldsymbol{x}' = (x_1, x_2, \ldots, x_j-1, \ldots, x_J)$ must also be an admissible state ($\boldsymbol{x}' \in \Omega$). This condition implies the very reasonable property that at the completion of a call, the resources it used are immediately available to serve other calls, and that call durations are independent of the system state. We consider policies that retain the coordinate convexity of the state space. Under such policies, a new call is admitted with probability 1 if the state to be entered is in the admissible region; otherwise, it is blocked. The objective is to determine the coordinate-convex set Ω^* that provides the optimal value of the specified performance index. Thus, a coordinate-convex policy is specified in terms of the set of admissible states in a discrete state space. The control policy is effectively a further restriction of the (already coordinate-convex) admissible state space defined by the capacity constraints (Eq. (1)).

To obtain a truly optimal admission-control policy, it would be necessary to use dynamic-programming techniques. Whereas a coordinate-convex policy is fully specified by the set of admissible states, dynamic programming requires not only the specification of the set of admissible states but also the specification of the transitions between states that are permitted. In dynamic programming solutions, some transitions are not permitted (e.g., some calls are not accepted) even though permitting them would bring the system to an admissible state (a state that can be reached by means of a different path). The product-form solution does not apply when the state space is not coordinate convex or when not all transitions to admissible states are permitted. Thus, Markovian decision techniques are needed to evaluate system performance under a large number of alternative dynamic-programming policies. Nevertheless, although the optimal policy is not necessarily a coordinate-convex one, Jordan and Varaiya have provided examples in which the best coordinate-convex policy performs almost as well as dynamic-programming solutions [20]. A crucial advantage of coordinate-convex policies is that they are easy to implement. Furthermore, because their product-form solution is easier to evaluate than a dynamic-programming solution, it is possible to solve much larger problems if we consider only coordinate-convex policies.

For notational purposes, we subdivide the control policy into a set of "threshold controls," and a set of "linear-combination controls." Thresholds restrict the number of calls that will be admitted to the individual circuits, and can be expressed as

$$x_j \leq X_j = \text{ threshold on circuit } j, \qquad 1 \leq j \leq J. \tag{3.4}$$

The linear-combination controls are restrictions on the sums of the number of calls of various types, i.e.,

$$\sum_{j \in S_l} \alpha_{lj} x_j \leq Y_l = \text{ threshold on the weighted total number of calls of types } \in S_l, \tag{3.5}$$

for suitable subsets S_l of the set of all call types, as we discuss later in this section. The coefficients α_{lj} are included in Eq. (3) to permit different weights for different call types. Although not all coordinate-convex policies can be described by Eqs. (2) and (3), this description is sufficiently rich since empirical evidence has shown that even the class of threshold policies alone provides solutions that are very nearly optimal, if not exactly optimal, in the class of coordinate-convex policies.

Although it is generally necessary to evaluate all coordinate-convex policies—that is, all combinations of threshold values, all S_l's that represent linear combinations of the circuits with all values of Y_l, and all appropriate values of α_{lj} —to discover and verify the optimal one, we use $\alpha_{lj} = 1$ throughout this report. This simplification is required because of the complexity of the problem for even small networks. Since the parameters we wish to optimize are the boundaries of the state space (which, in fact, correspond to the chosen control policy), the dimensionality of the problem is equal to the number of control policy constraint equations. Consideration of values of α_{lj} other than 1 would cause the dimensionality to explode. The use of $\alpha_{lj} = 1$ is also partially justified by the theoretical results of Foschini and Gopinath [25], and Jordan and Varaiya [20], and by our empirical observations that indicate that control policies using only $\alpha_{lj} = 1$ are very nearly, if not actually, optimal, as mentioned above.

An important question that arises in performance evaluation is how to organize the description of the different coordinate-convex regions. The individual thresholds must be examined, and the sets S_l must be determined. We find the sets S_l by examining, for each node, subsets of the set of circuits that intersect that node. For example, if three call types (a, b, and c)

pass through a node, it is sometimes beneficial to limit the pairwise sums $x_a + x_b$, $x_a + x_c$, and/or $x_b + x_c$. To limit the size of the state space that must be searched, we do not consider linear-combination controls that lower the bounds imposed by the capacity constraints of Eq. (1). For example, in the network of Fig. 3.1, the only location that circuits intersect is at node 1 where all three circuits cross. The corresponding capacity constraint is $x_1 + x_2 + x_3 \leq T_1$. We do not consider linear-combination controls that impose a value less than T_1 on the sum of all three call types. Doing so would be equivalent to removing transceivers from the node (without relocating them to other nodes). Therefore, in our example, in addition to the capacity constraints of Eq. (1) and the threshold constraints of Eq. (2), we consider the following three linear-combination controls:

$$x_1 + x_2 \leq Y_1, \qquad x_1 + x_3 \leq Y_2, \qquad x_2 + x_3 \leq Y_3.$$

3.3 The Solution and Performance Measures

We assume that the generation process for a call of type j (i.e., a call that uses circuit j) is Poisson with rate λ_j, and that its duration is exponentially distributed with mean $1/\mu_j$; the corresponding offered load is $\rho_j = \lambda_j/\mu_j$. Furthermore, control is centralized, and the resources needed to support a circuit are acquired simultaneously when the call arrives and are released simultaneously when the call is completed. Calls are blocked when one or more nodes along the path do not have a transceiver available or when a decision is made not to accept a call, as is discussed below. Blocked calls are assumed to be lost from the system.

Under these conditions, in conjunction with the use of coordinate-convex policies, it has been shown [19, 20] that the Markov chain describing the system state is time-reversible and has the product-form stationary distribution

$$\pi_\Omega(x) = \pi_\Omega(0)\prod_{j=1}^{J}\frac{\rho_j^{x_j}}{x_j!}, \tag{3.6}$$

where $\pi_\Omega(0)$ is the normalization constant given by

$$\pi_\Omega(0) = \left\{\sum_{x\in\Omega}\prod_{j=1}^{J}\frac{\rho_j^{x_j}}{x_j!}\right\}^{-1}. \tag{3.7}$$

Although the assumption of exponential call durations is not necessary for the product-form solution to apply (a Poisson arrival process and general service time distribution is sufficient [26]), we prefer to make this assumption in order to maintain a Markovian state space structure that can be used for dynamic analysis [21, 22, 23], and which permits the use of Standard Clock simulation techniques. It is convenient to define an index quantity associated with any given subset Ω' of the state space by

$$G(\Omega') = \sum_{x \in \Omega'} \prod_{j=1}^{J} \frac{\rho_j^{x_j}}{x_j!}. \tag{3.8}$$

Then the connection between the admissible state space Ω and the crucial quantity $\pi_\Omega(0)$ is provided by the relation $G(\Omega) = \{\pi_\Omega(0)\}^{-1}$.

The control policy is defined by the specification of the admissible state space Ω. For any such state space, it is straightforward (though time consuming) to evaluate $\pi_\Omega(0)$, which in turn permits the evaluation of performance measures such as throughput and blocking probability. In state $\boldsymbol{x}$ the total number of active calls is

$$\gamma(\boldsymbol{x}) = \sum_{j=1}^{J} x_j. \tag{3.9}$$

We define throughput $\Gamma(\Omega)$ to be simply the expected number of active calls averaged over the system state:

$$\Gamma(\Omega) = \sum_{\boldsymbol{x} \in \Omega} \{\gamma(\boldsymbol{x})\pi_\Omega(\boldsymbol{x})\}. \tag{3.10}$$

Our throughput metric is somewhat different from the usually defined one, which is the number of *completed* calls per unit time. We have found our metric to be useful because it is closely related to the notion of "residual capacity" [1], which is a measure of the resources available for data in integrated voice/data networks after voice traffic has claimed its required resources. If all call types have the same expected length, or alternatively if the throughputs of each call type are weighted in proportion to their expected length, our definition simply "scales" uniformly the usual metric.

Blocking probability $P_b(\Omega)$ is the ratio of the expected number of blocked calls per unit time to the expected total number of call arrivals per unit time:

$$P_b(\Omega) = \frac{\sum_{j=1}^{J} \lambda_j P_{bj}(\Omega)}{\sum_{j=1}^{J} \lambda_j} = \frac{\sum_{x \in \Omega} \sum_{j=1}^{J} 1\left(\left(x_j + 1\right) \notin \Omega\right) \lambda_j \pi_\Omega(x)}{\sum_{j=1}^{J} \lambda_j}, \tag{3.11}$$

where $P_{bj}(\Omega)$ is the fraction of type-j calls that are blocked, and $1(\cdot)$ is the indicator function, which is 1 if the argument is true and 0 otherwise.

3.4 Computational Issues in Evaluating the Solution

Knowledge that the equilibrium distribution $\pi_\Omega(\boldsymbol{x})$ satisfies the product-form distribution greatly simplifies the evaluation of system performance. Since the distribution is known to within the normalization constant $\pi_\Omega(0)$, there is no need to solve the balance equations associated with the Markov chain that describes our system. However, the evaluation of $\pi_\Omega(0)$ is computationally intensive because it requires the evaluation and summation of a large number of terms of the form $\prod \rho_j^{x_j}/x_j!$. (Computational issues associated with loss networks are addressed in [27]). Considerable effort has been exerted in developing efficient procedures for calculating the normalization constant (see e.g., [28]), but such methods are generally problem specific and of little help to our problem.

Our goal is to restrict the admissible state space to a coordinate-convex region such that the desired performance measure is optimized. The direct approach is to compute $\pi_\Omega(0)$ and the performance index for all possible coordinate-convex regions. We have developed a recursive procedure to accelerate the evaluation of a large number of different admission-control policies, and a descent-search method to minimize the number of policies that must be evaluated in searching for the optimal one; both of these, which are discussed in [7, 8], are directly applicable to either cost criterion.

We have not quantified the computation time associated with these procedures, and we acknowledge that even with these enhancements the problem quickly becomes intractable. Our methods simply extend somewhat the range of problems that can be evaluated. By contrast, the ordinal optimization approach [1, 6], which is discussed in Sections 7 and 8, offers a method to extend significantly the size of problem that can be evaluated. The idea behind ordinal optimization, and a principle that underlies many optimization problems, is that a good suboptimal solution frequently provides a level of performance acceptably close to the true optimum. Therefore, it may be possible to reduce the complexity by making simplifying assumptions, or by solving a less complex but "ordinally related" problem.

3.4.1 Use of the CM-5E to Obtain the Product-Form Solution

The difficulty in determining the optimal admission-control policy stems both from the number of policies that have to be evaluated and compared, as well as from the complexity of the performance evaluation under any individual policy. The complexity of both aspects of this problem increases rapidly with increasing problem size (i.e., number of circuits and number of transceivers per node).

The need to compute a large number of normalization constants simultaneously (i.e., one for each control policy that is being evaluated), each of which requires a calculation of considerable complexity, suggests the potential applicability of the CM-5E. To exploit the massively parallel nature of this machine, we distributed the policies over the virtual processors (VP's). For example, we can assign the computation of one normalization constant to each VP (easily extended to several policies/normalization constants per VP). Under this method, the front end sequentially calculates $\prod \rho_j^{x_j}/x_j!$ for every state $\boldsymbol{x}$ in the uncontrolled system, and distributes each of the results to all of the policies (which correspond to VP's). Each VP then sums the terms associated with the states in the admissible region associated with its admission-control policy. Ideally this approach would allow N policies to be evaluated in the time it takes to calculate the product-form solution once, where N is the number of VP's. We don't have precise timing estimates, but we seem to be getting near that performance. However, even calculating the product-form solution once can become infeasible because of the explosion of the state space. For the network example to be shown in Fig. 5.2, the largest number of transceivers per node that can reasonably be handled by the CM-5E is 40.

Therefore we have tried an alternative approach in which the evaluation of an individual normalization constant is distributed over the VP's. Under this method, each VP calculates $\prod \rho_j^{x_j} / x_j!$ for one state $\boldsymbol{x}$ (or, more generally, for several states). To determine the normalization constant (and hence the desired equilibrium distribution) for the given policy, the VP's return their un-normalized computations of the state probability (i.e., $\prod \rho_j^{x_j} / x_j!$) to the front end, which sums appropriate sets of these values and calculates appropriate ratios. Thus, the inherent parallelism in the most complex part of the computation, namely the evaluation of $\prod \rho_j^{x_j} / x_j!$ for each state, is exploited by distributing the calculations over the set of CM-5E processors. This approach works, and is very fast. However, the requirement to store the value of $\prod \rho_j^{x_j} / x_j!$ for every state $\boldsymbol{x}$ requires excessive memory for large problems (even on NRL's CM-5E, which has (or had at the time these computations were performed) the largest memory in the world). By contrast, the computation of the complete normalization constant on a VP does not require much storage because the individual terms in the summation are simply summed and written over. Thus, in our original method the efficiency is limited by the need to recalculate the quantity $\prod \rho_j^{x_j} / x_j!$ for many states $\boldsymbol{x} = \{x_1, ..., x_J\}$, whereas in the alternative method memory constraints are the dominant consideration.

One possible approach for the approximation of normalization constants in large product-form network problems is the use of Monte-Carlo summation with importance sampling [29]. We have not used this approach yet, but we expect to do so in the future. Instead, we have used SC simulation techniques to evaluate a large number of admission-control policies in parallel. In the next section we discuss our SC simulation model of circuit-switched networks.

4 SC SIMULATION OF CIRCUIT-SWITCHED VOICE-ONLY NETWORKS

We have used SC techniques to evaluate the performance of a circuit-switched network under a number of different admission-control policies. The availability of the exact solutions obtained in [8, 30] has allowed us to validate the results of our simulations. The discrete parameters of interest in this case are the circuit thresholds X_j and the linear-combination controls Y_l. The events that must be generated are arrivals and departures. Arrivals to circuit j are denoted by a_j, and occur at rate λ_j^V. The departure rate for each active call on circuit j is μ_j^V; thus, if there are x_j active calls on circuit j, the departure rate for calls of type j is $x_j \mu_j^V$. This situation is translated to events in the simulation as follows. We define X_{jmax} to be the maximum value X_j can have over the entire set of policies. Clearly, X_j cannot be greater than the number of transceivers at any node along the path defined by circuit j; thus

$$X_{jmax} = \min_{i \ni c_{ji}=1} (T_i). \tag{4.1}$$

Because there may be up to X_{jmax} active calls on circuit j, we must consider X_{jmax} different departure events d_{jn} ($n = 1, \ldots, X_{jmax}$) for circuit j, namely

$d_{jn} \Rightarrow$ feasible departure from circuit j if $x_j \geq n$, otherwise fictitious event. (4.2)

Following the usual technique used in uniformization [9], the *maximal rate* of this system is

$$\Lambda = \sum_{j=1}^{J}\left(\lambda_j + X_{jmax}\mu_j\right). \tag{4.3}$$

Figure 4.1 shows the ratio yardstick for a network with five circuit types, where the maximum threshold on each circuit is three. This ratio yardstick corresponds to operation of the network at the maximal rate, and it can generate all events that are needed for the simulation of systems with control policies in which the circuit thresholds do not exceed 3. In our example $\lambda_j^v = \mu_j^v = 1$, which implies that all events are equally likely. All arrival events in this system are real, and the corresponding calls are accepted as long as their acceptance does not violate any of the capacity constraints or any of the threshold or linear-combination control values. However, departure events are fictitious if an insufficient number of calls of that type are currently active, as discussed above. For example, referring again to Fig. 4.1, an event of type d_{13} will be fictitious (and hence ignored, although time will be updated) if there are less than three calls of type 1 currently active.

In [1, 2] we discussed the possibility of using "custom" ratio yardsticks, which are constructed to conform with the particular parameters of each sample path, resulting in the generation of fewer fictitious events. Despite their potential advantages, we have found in our SC simulations that the added burden of maintaining and accessing separate custom yardsticks for every sample path outweighs the reduction in the number of fictitious events provided by their use [31]. Furthermore, it was demonstrated in [32] and [33] that the use of custom yardsticks decreases the correlation among the sample paths, thus necessitating considerably longer simulation runs to achieve comparable ordinal comparisons.

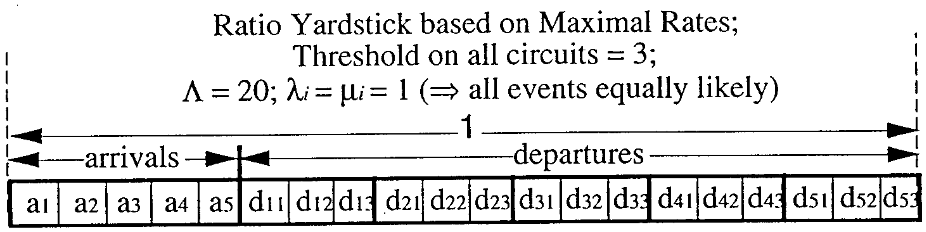

Ratio Yardstick based on Maximal Rates;
Threshold on all circuits = 3;
Λ = 20; λi = μi = 1 (⇒ all events equally likely)
1
arrivals
departures
a1
a2
a3
a4
a5
d11
d12
d13
d21
d22
d23
d31
d32
d33
d41
d42
d43
d51
d52
d53

Fig. 4.1 — Ratio yardsticks showing event sets for the circuit-switched network

4.1 Sequential Machine Simulation Timing Results

We have performed both Brute Force (BF) and SC simulations of this network model to obtain timing results. Performance results presented in [34] and [31] show that when more than 2,000 sample paths are simulated, the SC method is approximately 2.5 times faster than the BF approach on a sequential machine. Despite the 17.5-fold increase in the size of the event set for the circuit-switched voice model as compared to the M/M/1/K queueing model (there are 35 events in the voice model as opposed to 2 in the M/M/1/K model), the time per call for the event-generation and state-updating functions is approximately the same as that in the M/M/1/K simulations.[5] Based on results presented in [1, 2], we have found that the time per call to the state-update and the event-generation functions is relatively insensitive to the size and complexity of the problem. Thus, the SC approach should scale reasonably well, provided that sufficient memory is available.

Our timing results for simulations of an integrated voice/data network, which were also presented in [1, 2], suggest that the SC method scales well with increasing problem complexity. The integrated network is considerably more complex than the voice-only network. In addition to the 35 voice-event types, there are ten types of data-arrival events and their corresponding deterministic data-departure events (with data events occurring at a considerably higher rate than voice events). Despite this added complexity, the time required to simulate the integrated network is only slightly larger than that for the voice-only network. This further confirms the above observations on scalability.

4.2 CM-5E Simulation Timing Results

We have not performed the extensive timing tests on the CM-5E that we did on sequential platforms. However, we have observed that the time required to perform SC simulations on the CM-5E is virtually independent of problem size for up to two policies per virtual processor (i.e., for up to 16,384 policies). For problem sizes of about 2,000 policies, we have seen on the CM-5E two orders of magnitude speedup over the time required on sequential machines.

Two measures that have been proposed to characterize the "speedup" achieved by parallel computing are Amdahl's Law and Gustafson's scaled speedup [35]. Both measures are functions of a quantity denoted by β, which is the ratio of the time required for the serial (front end) portion of the computation to the total time (serial + parallel) required for the computation. Amdahl's Law assumes that both the serial and the parallel portions of the computation scale up with the number of processors, whereas Gustafson's scaled speedup assumes that the time required for the serial portion is fixed and scaling only effects the parallel portion. The Gustafson model best fits our SC simulations of M events, because the serial portion of the computation is the time required to generate M events, and is independent of the number of sample paths. The Gustafson scaled speedup measure is given by

$$S = \beta + n\,(1 - \beta) \qquad (4.4)$$

where n is the number of processors.

Using timing results from our Brute Force simulations of circuit-switched voice networks [[31], Table 3], we obtain an estimate of $\beta \approx 0.6958$ (= (`get_event` + `random` + `log`)/ (`get_event` + `random` + `log` + `update`) = 13.3/19.1143). Using this quantity in the Gustafson scaled speedup measure with $n = 256$ processors on the CM-5E provides a predicted speedup of 78.567, which is commensurate with our casual timing observations. The use of $n =$

[5] This is not surprising, because a well-known property of the alias method is that the time needed to generate an event is independent of the number of event types.

8192 in the above expression, corresponding to the total number of virtual processors on the CM-5E, results in a predicted speedup of 2492. We have not observed speedup values in this range, but neither have we fully exploited the capabilities of the CM-5E by loading the VP's sufficiently to cause the parallel computation time to dominate total simulation time. As mentioned above, we have found that total simulation time is virtually independent of the number of sample paths for up to 16,384 sample paths. This suggests that greater speedup is possible when simulating larger numbers of sample paths.

5 NUMERICAL EVALUATION OF NETWORK PERFORMANCE

Most of the numerical examples presented in this report are based on the sample network shown in Fig. 5.1, in which links are shown connecting all nodes that are within communication range of each other. Figure 5.2 shows five circuits superimposed on this network graph; a call on one of these circuits requires the use of one transceiver at every node along the corresponding path.

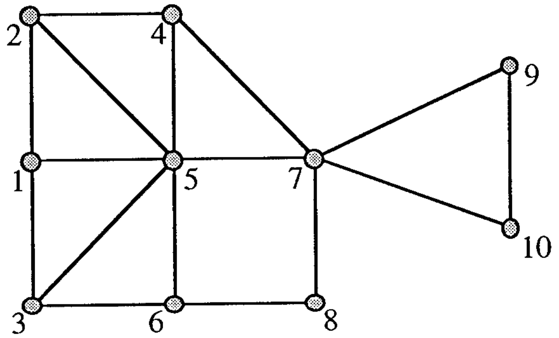


Fig. 5.1 — An example ten-node multihop network

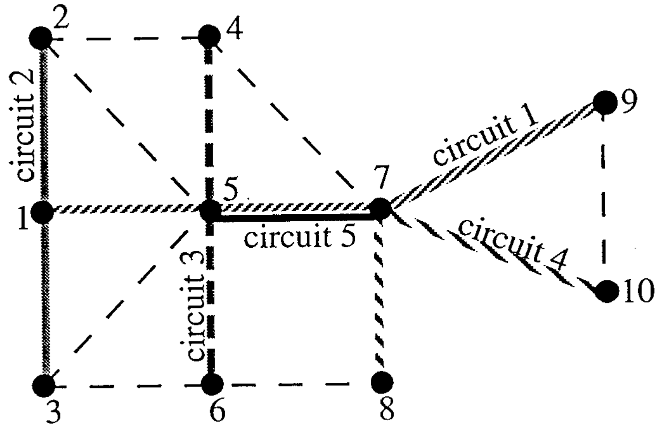


Fig. 5.2 — Five circuits superimposed on the network of Fig. 5.1

By examining Fig. 5.2, we can expand the capacity constraints Eq. (1) as follows:

$$x_j \leq \min_{i \ni \text{ node } i \in \text{ circuit } j} \{T_i\}, \quad j = 1,\cdots,5, \qquad i = 1,\cdots,10$$

$x_1 + x_2 \leq T_1$ (capacity constraint from node 1)

$$x_1 + x_3 + x_5 \leq T_5 \qquad \text{(capacity constraint from node 5)} \qquad (5.1)$$
$$x_1 + x_4 + x_5 \leq T_7 \qquad \text{(capacity constraint from node 7)}$$

The control policy, which is expressed in terms of the variables X_j and Y_l whose values are to be optimized, is obtained from the capacity constraints and is written as

$$x_j \leq X_j \qquad \text{(thresholds)}$$
$$x_1 + x_3 \leq Y_1 \quad x_1 + x_5 \leq Y_3 \quad x_3 + x_5 \leq Y_4 \qquad \text{(linear-combination controls from node 5)} \qquad (5.2)$$
$$x_1 + x_4 \leq Y_2 \quad x_4 + x_5 \leq Y_5 \qquad \text{(linear-combination controls from node 7)}$$

We consider an example in which the loads of all circuits are equal (i.e., $\rho_j = \rho, j = 1, \ldots, 5$), and eight transceivers are allocated to each node (i.e., $T_i = 8$, $i = 1, \ldots, 10$). The maximum number of calls permitted on any one circuit is arbitrarily restricted to six ($X_j \leq 6$).[6] We introduce the following notation:

Ω_X = the best control policy for the system controlled by adjusting only the thresholds,
Ω^* = the "optimal"[7] policy (using both thresholds and linear-combination controls).
Γ_Ω = the throughput of the uncontrolled system,
Γ_X = the throughput of the system under policy Ω_X,
Γ^* = the throughput of the fully controlled (using both thresholds and linear-combination controls) system under policy Ω^*,
$P_b(\Omega)$ = the blocking probability of the uncontrolled system,
$P_b(\Omega_X)$ = the blocking probability of the system controlled by adjusting only the thresholds,
$P_b(\Omega^*)$ = the blocking probability of the fully controlled system,

The threshold policy Ω_X is simply the set of threshold values $\{X_1, \ldots, X_5\}$ that produces the maximum value of throughput among the class of policies in which only the thresholds can be controlled. The "optimal" policy Ω^* is the set of X_j's and Y_l's $\{X_1, \ldots, X_5, Y_1, \ldots, Y_5\}$ that produce the maximum value of throughput among the class of policies in which both thresholds and linear-combination controls can be adjusted. In Table 5.1 we list all of the above quantities as a function of the offered load ρ, along with the percentage improvement in performance that is observed as compared with the performance of the uncontrolled system. The optimal policies listed in the table were obtained by performing an exhaustive search over all subspaces defined by the thresholds and linear-combination controls of Eq. (7). This search was facilitated by our recursive technique for computing the normalization constants and the corresponding blocking probabilities associated with each candidate subspace [7, 8]. Elements of Ω^* that are different from those corresponding elements of Ω_X are shown in bold.[8]

[6] Setting $X_j < T_i$ prevents circuit j from acquiring all of the resources at node i. In addition to providing a degree of fairness, doing so also reduces the complexity of the problem by reducing the size of the admissible state space.

[7] These policies may not be the true optimal coordinate-convex policies, since our search is limited to the class of policies defined by thresholds and linear-combination controls, which is a subset of all coordinate-convex policies (see Section 2.1). Therefore, we cannot be certain that any solution we find is truly optimal in this class. However, we conjecture that these solutions are close to the true optimal because the use of a descent search has not found any better solutions. This constitutes sufficient evidence provided that blocking probability is actually a unimodal function of the boundaries of the state space as discussed in Section 3.2.

[8] Let us clarify the use of bold entries for the values of Y_l. For all values of ρ, $Y_1 = Y_2 = Y_4 = Y_5 = 8$. Thus these four linear-combination control values are at the "uncontrolled" value, which is equal to the number of transceivers at

Table 5.1 — Optimal control policies for the network of Fig. 5.2 with $T_i = 8$ and $X_j \leq 6$.

ρ	Γ_Ω	Γ_X {%gain}†	Γ^* {%gain}†	$P_b(\Omega)$	$P_b(\Omega_X)$ {%gain}†	$P_b(\Omega^*)$ {%gain}†	Ω_X	Ω^*
2.5	10.1807	10.1811 {0.0040}	10.1811 {0.0042}	0.18555	0.185512 {0.0175}	0.185511 {0.0183}	{5,6,6,6,6}	{5,6,6,6,6, 8,8,**7**,8,8}
3.5	12.0336	12.07499 {0.3438}	12.07667 {0.3578}	0.31237	0.310001 {0.7568}	0.309905 {0.7876}	{2,6,6,6,5}	{**3**,6,6,6,5, 8,8,**5**,8,8}
4.5	13.2986	13.48778 {1.4222}	13.50657 {1.5635}	0.40895	0.400543 {2.0555}	0.399708 {2.2598}	{2,6,6,6,3}	{2,6,6,6,**4**, 8,8,**4**,8,8}
5.5	14.2267	14.57842 {2.4721}	14.62406 {2.7929}	0.48267	0.469876 {2.6497}	0.468216 {2.9935}	{1,6,6,6,2}	{**2**,6,6,6,**3**, 8,8,3,8,8}
6.5	14.9364	15.45569 {3.4768}	15.51528 {3.8757}	0.54042	0.524440 {2.9568}	0.522607 {3.2960}	{1,6,6,6,2}	{**2**,6,6,6,2, 8,8,**2**,8,8}
7.5	15.4950	16.13004 {4.0985}	16.23304 {4.7632}	0.58680	0.569865 {2.8860}	0.567119 {3.3541}	{1,6,6,6,1}	{**2**,6,6,6,**2**, 8,8,2,8,8}
8.5	15.9449	16.68042 {4.6127}	16.77403 {5.1997}	0.62483	0.607519 {2.7697}	0.605317 {3.1222}	{1,6,6,6,1}	{**2**,6,6,6,**2**, 8,8,2,8,8}
10	16.4753	17.28274 {4.9006}	17.36501 {5.4000}	0.67049	0.654345 {2.4084}	0.652700 {2.6537}	{1,6,6,6,1}	{**2**,6,6,6,**2**, 8,8,2,8,8}
15	17.5489	18.34001 {4.5078}	18.39848 {4.8420}	0.76601	0.755467 {1.3770}	0.754687 {1.4787}	{1,6,6,6,1}	{**2**,6,6,6,**2**, 8,8,2,8,8}

† All gains are relative to the uncontrolled system.

The results show that at low network loads (for ρ less than 2.5 in this network), it is best to accept all calls as long as transceivers are available, i.e., the optimal policy is to administer no control. As ρ is increased, calls of type 1 are the first to have their threshold reduced. This is intuitively satisfying because circuit 1 shares at least one node with each of the other circuits. Calls of type 5 are the next to have their threshold reduced. This is because they interfere with calls of type 3 and 4. Calls of type 2, 3, and 4 never have their thresholds reduced because (after type 1 and type 5 calls have been eliminated) they do not share resources with any other type of call. We see that incorporation of the linear-combination controls permits the increase of some of the threshold values. For example, with ρ = 3.5 the threshold X_1 can be increased from 2 to 3 by administering the policy Ω^*, in which the linear-combination control $(x_1 + x_5 \leq) \ Y_3$ is set at 5. Thus the maximum number of type-1 calls is increased, but the sum of the number of type-1 and type-5 calls that can be simultaneously active is reduced from 7 to 5 under Ω^*. In this example, use of Ω^* resulted in a slight improvement over the performance obtained by using Ω_X.

Although operation at blocking levels as high as those considered here may be unrealistic, our results are informative because they demonstrate a range of loading where the use of admission control can provide improved performance. For example, in the network

each node; hence these entries are not bold since they do not differ from those of the threshold-only policy Ω_X. On the other hand, Y_3 is always less than the number of transceivers at node 5, but this entry is bold only when it is different from the summation $(X_1 + X_5)$ in the Ω_X solution.

studied here, the only way to reduce the blocking probability to a value less than 0.18 (which would typically be an unacceptably high value) is to lower the offered load to a value of ρ that is less than 2.5. However, when $\rho \leq 2.5$ the optimal admission-control policy is the uncontrolled policy. Thus, the network appears to be "self-regulating" in the sense that the uncontrolled policy provides optimal performance throughout a wide range of desirable performance values. Furthermore, at higher offered loads the degree of improvement that can be obtained by means of admission control is quite small; in terms of throughput it is at most 5.4%, and in terms of blocking probability it is at most 3.35%. We discuss this self-regulation phenomenon further in Section 6, where we explain this inability to obtain significant performance gains.

Our observations raise a fundamental question: Is the improvement obtained by administering control worth the effort invested in performing a search over many candidate policies, each of which requires the evaluation of the normalization constant associated with the product-form solution? This question cannot be answered on the basis of a single isolated example. Many of the examples presented in this report were motivated by the desire to find a case where the use of an admission-control policy has a substantial impact on the network performance.

5.1 Variations on the Admission Control Problem

By using overall blocking probability as our performance measure, we implicitly assume that all call types have equal importance. In some applications, it may be appropriate to associate with each call type a "weight," which reflects the revenue returned for providing the service, or the cost (e.g., lost revenue) of failing to provide the service. For example, certain circuits may be weighted more heavily if they carry traffic of higher priority. In [7, 8] we discuss the performance improvement that can be achieved by means of admission control when unequal weights are applied to the circuits.

In [7, 8] we observed that the choice of routes has a significant impact on system performance. In fact, the choice of path sets to reduce congestion can be more effective than admission control in reducing blocking probability. However, we also observed that the application of admission control does improve further the performance obtained by choice of good paths alone.

6 NETWORK SELF-REGULATION

In [7, 8] we examined "multi-cross networks," a class of networks that we believe provide insight into the reasons we have not observed significant performance improvement through the use of admission-control policies. In this section we first review our observations on network self-regulation, originally made in [7, 8] for the network example discussed in Section 5. We then discuss network performance as the number of transceivers per node is increased to values as large as 4000. Analytical results were obtained on the Connection Machine CM-5E for as many as 40 transceivers per node. Standard Clock simulation on the CM-5E was used for examples with a larger number of transceivers.

6.1 Multi-Cross Networks and Network Self-Regulation

In an attempt to determine the reason that such a small degree of performance improvement was achieved through the use of admission control, we have examined the network configuration of Fig. 6.1, which we refer to as a "multi-cross network." The horizontal circuit c_0 intersects each of the remaining N circuits, $c_1, \ldots, c_N$. Circuits c_1 through c_N are mutually disjoint and share resources only with circuit c_0. The network of Fig. 5.2, which has served as the primary testing ground for our admission-control studies, is similar to the multi-cross network since circuits c_2, c_3, and c_4 are mutually disjoint and share resources only with circuit c_1 at one node. However, it differs in that circuit c_5 intersects with circuit c_1 at two nodes, and with circuits c_3 and c_4 at one node each. Nevertheless, it is reasonable to expect that an examination of multi-cross networks should be able to provide some degree of insight into the operation of more-complex networks.

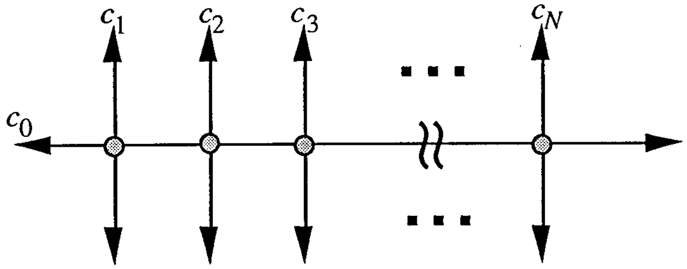


Fig. 6.1 — An example of a multi-cross network

Intuitively, it can be seen that at high offered loads or for high values of N (the number of mutually disjoint circuits intersecting circuit c_0), optimal admission control will restrict the number of calls allowed on circuit c_0, ultimately not admitting any calls on this circuit. Intuition might also suggest that the performance gain obtained by administering such control would continue to grow with N. Here, as throughout this report, we do not address the issue of fairness; the performance measure is defined simply in terms of overall average blocking probability.

We have examined these hypotheses by applying our descent-search method to a series of multi-cross networks with increasing N and fixed utilization rates and capacities. In these studies, we have set the node capacity T_i = circuit capacity = 6, $\rho_0 = 14$, and ρ_1 through $\rho_N = 7$. These parameters correspond to a very heavily loaded network. We shall see that although the optimal admission-control policy is quite different from the uncontrolled policy, there is little difference in performance at heavy loads. At low levels of offered load, the optimal policy is the uncontrolled policy, in which all calls are admitted as long as resources are available to support them; thus the network is naturally self-regulating.

Since at most only two circuits intersect at any node, there are no linear-combination controls to adjust. Thus, the descent search attempts to find the optimal value of the $N + 1$ circuit thresholds. The results for $N = 1, \ldots, 9$ are shown in terms of blocking probability in Fig. 6.2 and in terms of performance gain in Fig. 6.3. In the example of Section 5, when equally weighted services are competing for a single set of resources (e.g., in a multi-cross network with $N = 1$) the best strategy is to administer no control—all calls are accepted as long as resources are available to serve them. Thus, in Figs. 6.2 and 6.3 the optimal policy and the uncontrolled policy are the same when $N = 1$. However, at the high utilization rates assumed in this example, the optimal policy for $N > 1$ is $\Omega^* = \{0,6,6,\ldots\}$. Despite the fact that the optimal policy results in the rejection of all calls on circuit 0, the performance difference between the optimal and uncontrolled systems is relatively small, and decreases as N increases (for $N > 3$).

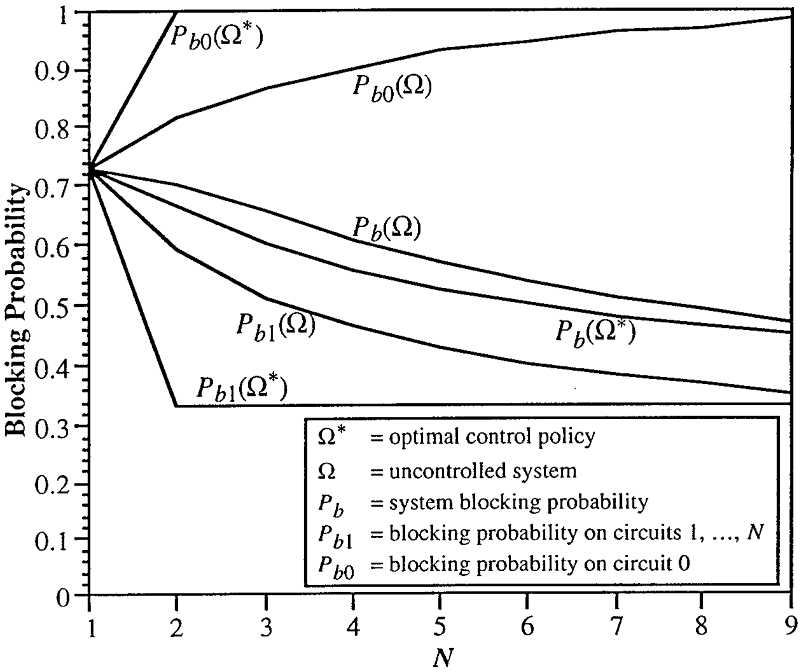


Fig. 6.2 — Blocking probability in multi-cross network vs N

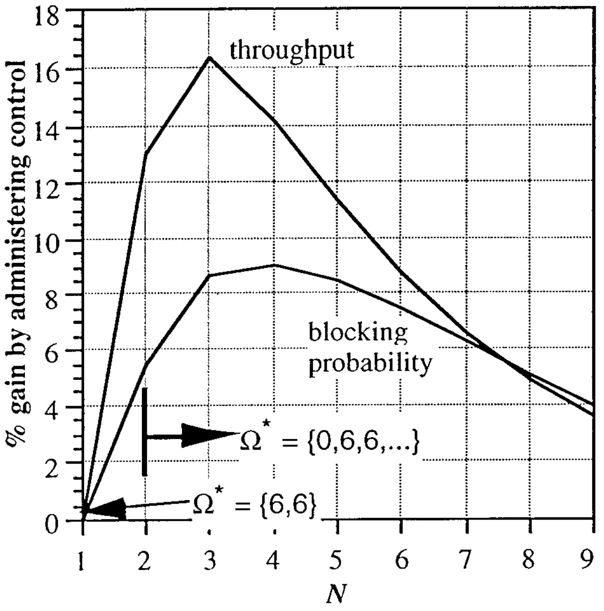


Fig. 6.3 — Throughput and blocking probability gains vs N

At lower network loads the optimal policy would admit some calls on circuit c_0 at lower values of N, but for sufficiently large N the optimal policy will block all calls on c_0. (For example, with $\rho_0 = 5$ and $\rho_1, \ldots, \rho_N = 1$, the optimal policy is the uncontrolled one when $N \leq 3$; when $N = 5$, $\Omega^* = \{5, 6, 6, 6, 6\}$; when $N = 7$, $\Omega^* = \{4, 6, 6, \ldots\}$; when $N = 11$, $\Omega^* = \{3, 6, 6, \ldots\}$.)

In the uncontrolled network, the circuit blocking probability on c_0, $P_{b0}(\Omega)$, increases with N as shown in Fig. 6.2; as it approaches 1.0, the circuit blocking probability on circuits $c_1, \ldots, c_N$, denoted by $P_{b1}(\Omega)$, approaches the minimum blocking probability $P_{b1}(\Omega^*)$, which is achieved by using the optimal control policy Ω^*. The uncontrolled overall blocking probability $P_b(\Omega)$ approaches $P_b(\Omega^*)$, the value obtained by applying the optimal policy. Thus, in the uncontrolled network as N increases there is increased likelihood that the resources needed to complete a call on circuit c_0 will be seized by one of the N multi-cross circuits, and calls on c_0 will have reduced access to the network while calls on the crossing circuits ($c_1, \ldots, c_N$) will see improved access. In a general network setting with equally weighted calls, this self-regulation phenomenon will tend to admit more calls on the circuits that compete with the fewest other circuits, and reject more calls on the circuits that must compete with many other circuits for resources. Our studies have shown that the best form of admission control is to restrict the access of those circuits that interfere with many other circuits. Thus the application of admission control and the result of self-regulation achieve nearly the same network performance. We therefore conclude that network performance generally can be improved by administering control, but the gain relative to the uncontrolled system is limited by the network's natural self-regulation. These results apply to the case in which all call types are of equal importance or generate equal revenues. In [7, 8] we demonstrated that considerable improvement in system performance can be achieved in systems different weights are applied to calls of different types.

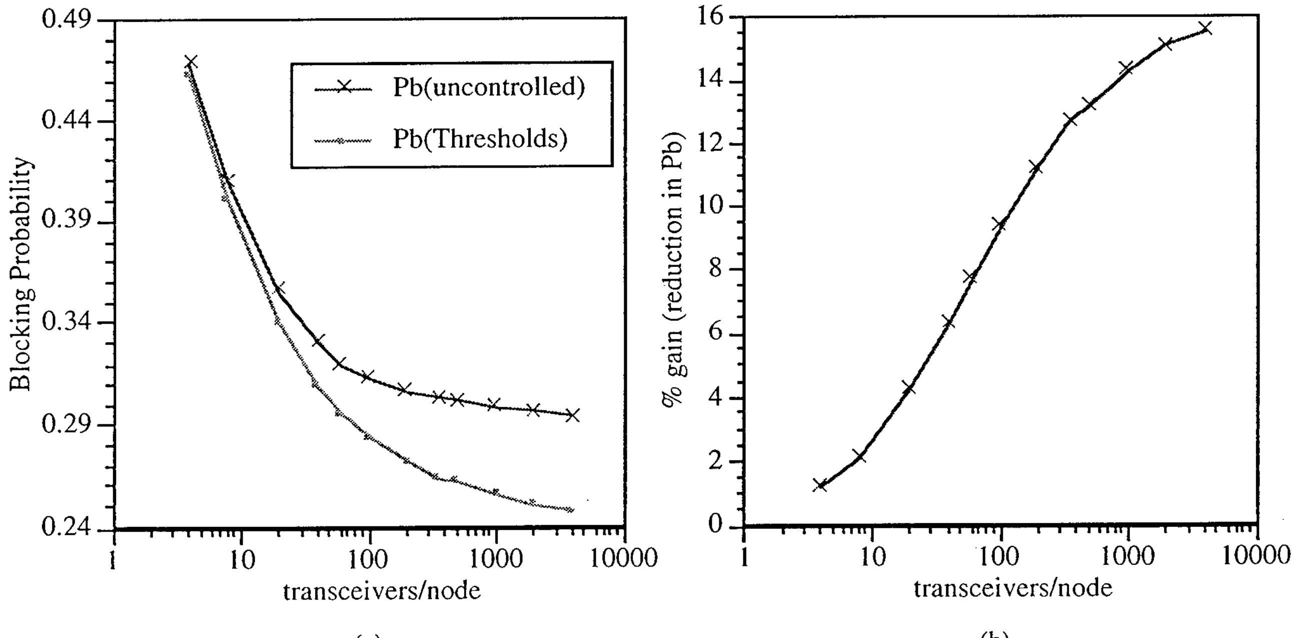
Although this example corresponds to a case of extremely heavy loading, the conclusions we have drawn on self-regulation apply more generally to wireless fixed-route circuit-switched networks as well. At sufficiently low traffic levels, the optimal policy admits some calls of type 0 if the value of N is also sufficiently small. However, as in the heavy-traffic example, the optimal threshold value for c_0 is reduced to 0 for sufficiently large N, and the performance of the uncontrolled system again approaches that of the optimally controlled one as N approaches ∞.

Our observations on self-regulation apply to wireless circuit-switched fixed-route networks with coordinate-convex admission-control policies, and it is unclear whether they can be generalized to other classes of networks. In particular, a crucial aspect of wireless networks is that the transceivers are not a priori allocated to particular links, thus permitting the almost complete sharing of transceivers (to the extent permitted by the coordinate-convex policy) among call types. In wired networks, considerable sharing is also possible, although the fact that links are permanently established may have an impact on the degree of improvement that can be achieved by admission control. Also, the use of alternate routing schemes may have significant impact in either wireless or wired applications. Further study is needed to determine whether, and to what degree, self-regulation is a widely shared property in such systems.

6.2 Self regulation in high-capacity networks

The availability of the Connection Machine CM-5E has provided us the capability to examine considerably larger examples than the one discussed in Section 5. Our focus has again been on the network of Fig. 5.2, where we have investigated performance as the communication capacity (transceivers per node) is increased from 4 to values as high as 4000. In all cases, we set the maximum number of calls per circuit at $N_c = 3/4\ N_t$, (where each node has N_t transceivers), and for values of N_t as large as 500 we performed an exhaustive search of the threshold policies $\{X_1, N_c, N_c, N_c, X_5\}$; $X_1, X_5 = 0, \ldots, N_c$ (i.e., we have varied the threshold on circuits 1 and 5 while holding the thresholds on circuits 2-4 constant at their maximum value).[9] Thus the exhaustive search requires the examination of $(N_c + 1)^2$ policies in our examples.

Where feasible (i.e., $N_t \leq 40$) the exact blocking probability was calculated via the product-form solution. The performance under one control policy was computed on each virtual processor, as discussed in Section 3.4.1. For $N_t > 40$, the product-form solution is prohibitively complex, even on a high performance computer such as the CM-5E. Therefore, SC simulations on the CM-5E were used to estimate the blocking probability. The baseline simulations were 10^6 events (i.e., total events including fictitious ones) in duration, which corresponds to approximately 85,000 voice arrivals per circuit.



(a)

(b)

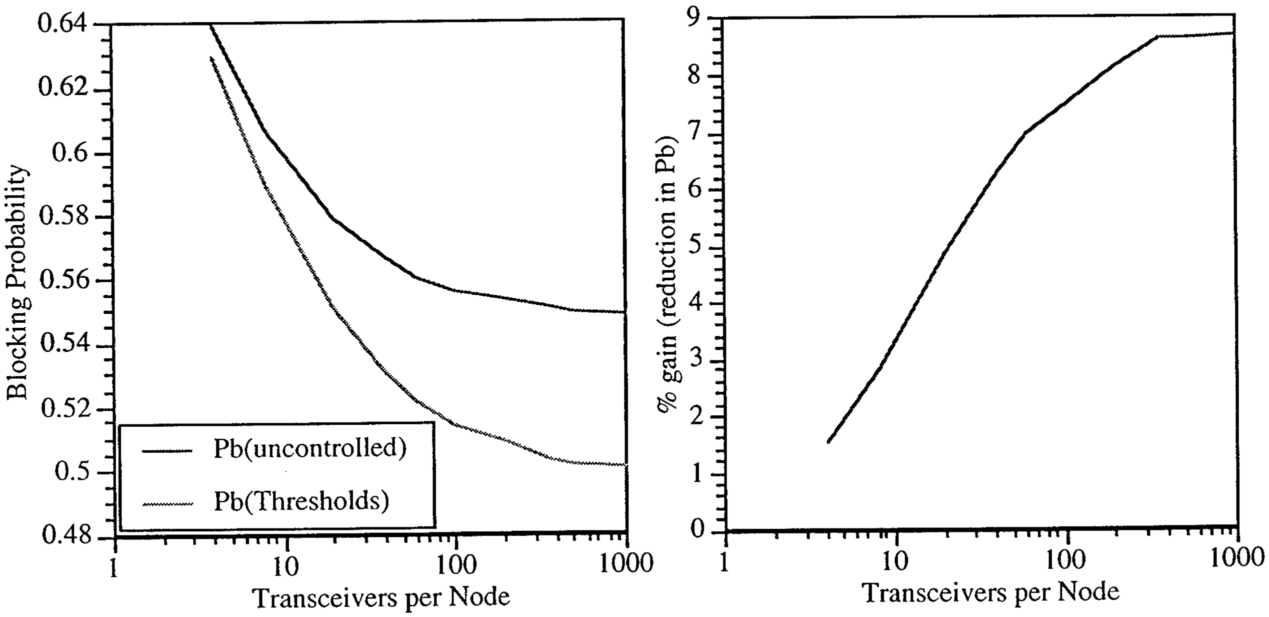
Fig. 6.4 — Blocking probability and performance gain vs N_t for $\rho = (9/16)N_t$

Figure 6.4(a) shows blocking probability as a function of the number of transceivers per node for an offered load of $\rho^V_j = \rho = (9/16)N_t$, $j = 1, \ldots, 5$, and Fig. 6.4(b) shows the corresponding percentage gain (i.e., the percentage reduction in blocking probability when the

[9] In Section 5 we noted that the optimal values of thresholds X_2, X_3, and X_4 for this problem are all equal to their maximum value when considering networks with up to 8 transceivers per node; thus we conjecture that it is sufficient to vary only X_1 and X_5 in networks with a larger number of nodes as well.

optimal admission-control policy is compared to the uncontrolled admission-control policy) when the optimal threshold admission-control policy is compared to the uncontrolled system. We see from Fig. 6.4(a) that blocking probability decreases significantly as the number of transceivers increases. This is apparently because of the increased statistical multiplexing capabilities that are provided by a large number of transceivers. We also see that the percentage gain in performance increases as the number of transceivers increases. Thus, the degree of self-regulation is apparently less in high-capacity systems than in low-capacity systems. The smaller systems are more characteristic of wireless networks, in which the number of transceivers is normally relatively small. The larger systems are more characteristic of high-speed networks. Thus it appears that network self-regulation may be more pronounced in smaller wireless systems than in large high-speed systems.

Figures 6.5(a) and (b) show similar results for the case of $\rho^V_j = \rho = N_t$, $j = 1, \ldots, 5$. Blocking probability is, of course, higher than in the previous example because the offered load is higher. We see that the difference in performance between the uncontrolled and optimally controlled systems is somewhat less than for the previous example.



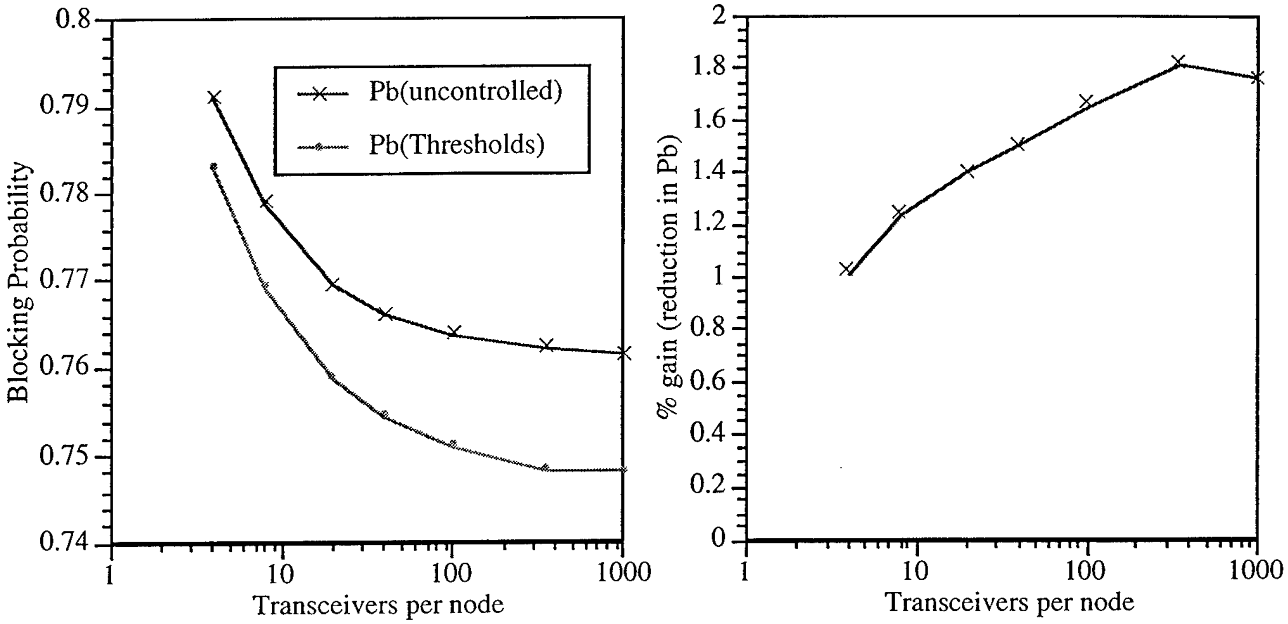
(a) (b)

Fig. 6.5 — Blocking probability and performance gain vs N_t for $\rho = N_t$

Finally, Figs. 6.6(a) and (b) show similar results for the case of $\rho^V_j = \rho = 2\,N_t$, $j = 1, \ldots, 5$. Blocking probability is further increased, and the improvement achievable by means of admission control is further decreased.

Fig. 6.7(a) is a three-dimensional view of the blocking probability surface as X_1 and X_5 are varied from values of 0 to 75 for the case of 100 transceivers per node. The P_b values are based on a simulation of 10^6 events. Although the simulation is relatively short (and hence the P_b values are relatively inaccurate), the figure shows a smooth surface with a well defined basin of low blocking probability. The minimum blocking probability value is 0.28341 with $X_1 = 23$,

$X_5 = 20$, and the maximum $P_b = 0.40217$ with $X_1 = X_5 = 0$. The uncontrolled system ($X_1 = X_5 = 75$) has a blocking probability of 0.31294. The same data is shown in contour form in Fig. 6.7 (b).



(a) (b)

Fig. 6.6 — Blocking probability and performance gain vs N_t for $\rho = 2N_t$

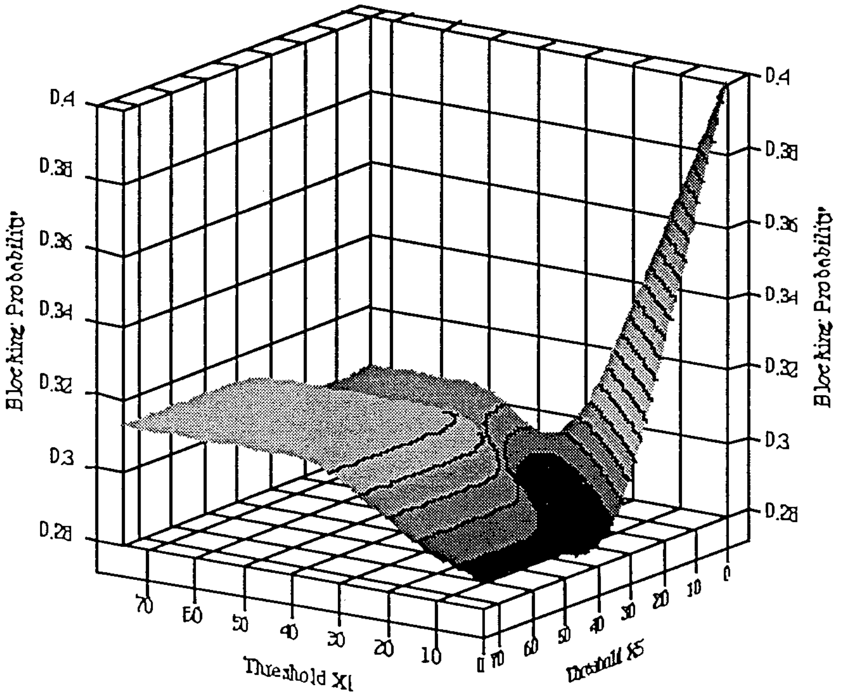


Fig. 6.7(a) — 3-dimensional plot of blocking probability vs thresholds X_1, X_5; $N_t = 100$, 10^6-event simulation

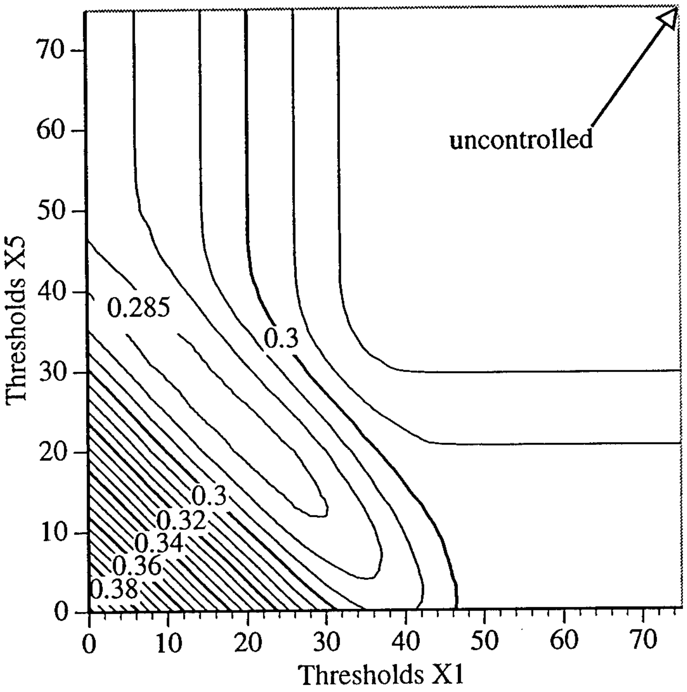


Fig. 6.7 (b) — Contour plot of blocking probability vs thresholds X_1, X_5; N_t = 100, 10^6-event simulation

6.3 Observations on Self-Regulation

Our earlier studies of admission control suggested that networks may exhibit a form of self-regulation in the sense that the uncontrolled system provides nearly optimal performance over a wide range of desirable performance values. These observations were based on the study of relatively small systems, such as the network of Fig. 5.2 with up to 8 transceivers per node. The study of the same network but with up to 4000 transceivers per node, made possible by the use of the CM-5E, has offered further insight into the self-regulation property of networks.

We have observed that blocking probability decreases significantly as the number of transceivers increases (when the offered load is maintained at a constant percentage of N_t). This behavior can be attributed to the increased statistical multiplexing capabilities that are provided by a large number of transceivers. We have also observed that systems with a large number of transceivers can benefit considerably more from admission control than systems with a small number of transceivers. For example, for the case of $\rho = (9/16)N_t$ the percentage gain (reduction in blocking probability) achievable through admission control increased from approximately 1% to nearly 16% as the number of transceivers was increased from 4 to 4000. We have shown that the degree of improvement achievable by admission control decreases as the offered load increases further. Our studies have concentrated on relatively high offered loads (for which blocking probabilities are unrealistically high). Typically, operation would be at lower offered loads (for which blocking probabilities would be much lower), for which little improvement is obtained by using admission control; thus the network is essentially self-regulating for such loads. Thus, the most interesting region (the region in which the greatest gain may be achieved by active admission control) is probably near that shown in Fig. 6.4, i.e., for the case of ρ =

$(9/16)N_r$. Even for this case, an improvement of 10% is achievable only when the number of transceivers per node is greater than 100.

Our conclusion from these studies is that some degree of self-regulation does, in fact, exhibit itself, although the degree of self-regulation decreases as the number of transceivers per node increases. For networks with dimensions characteristic of wireless systems ($N_t \leq 10$), little improvement is achieved by means of admission control. By contrast, larger systems (whose dimensions approach those of high-speed networks) do exhibit a higher degree of improvement when active admission control is applied; however, the improvement hasn't been greater than 16% in our examples. In view of the relatively small improvement that can be achieved by means of admission control, and because of the difficulty in evaluating system performance in all but relatively small examples, it may be appropriate in many applications to operate in an uncontrolled mode of operation.

7 ORDINAL OPTIMIZATION: Background and Results Obtained on Sequential Machines

Ordinal optimization [6] is the determination of control policies that perform relatively well compared to other candidate policies, without necessarily obtaining accurate estimates of the performance values associated with these policies. The motivation underlying ordinal optimization is that finding the optimal solution (or control policy) is often too costly or time consuming, although a suboptimal solution (that may be found quite easily) may provide sufficiently good performance.

In the literature, several different approaches have been considered for ordinal optimization. In the original paper on this subject [6], short simulation runs were used in conjunction with SC techniques to obtain an approximate ranking of control policies, and it was observed that many of the high-performance policies also perform well over the long run. It was also observed that, when the search space is too large to be examined exhaustively, acceptable performance can be obtained by examination of a randomly chosen subset of policies. The principles of order statistics [36] suggest that if a sufficient number of such randomly chosen policies are examined, some good ones will be encountered; these good policies can then be examined in greater detail. One way to do so is to perform a longer simulation run for each of these.

Other approaches to ordinal optimization have also been proposed for different types of problems. For example, the simulation time in systems with rare events can be prohibitive if standard brute-force techniques are used. However, the use of a surrogate design problem, in which the events of interest are not so rare, can speed up the simulation considerably while providing good ordinal rankings of control policies [37, 38]. For example, the policy that optimizes system performance based on the minimization of the probability of buffer overflow for a relatively small buffer size (a not-so-rare event) may also minimize the probability of buffer overflow for larger buffer sizes (which may be rare events), a property referred to as "ordinal equivalence" [37]. The first use of ordinal optimization in conjunction with SC techniques on a single-instruction, multiple-data (SIMD) machine was demonstrated in [39], where a reduction in simulation time of several orders of magnitude was achieved.

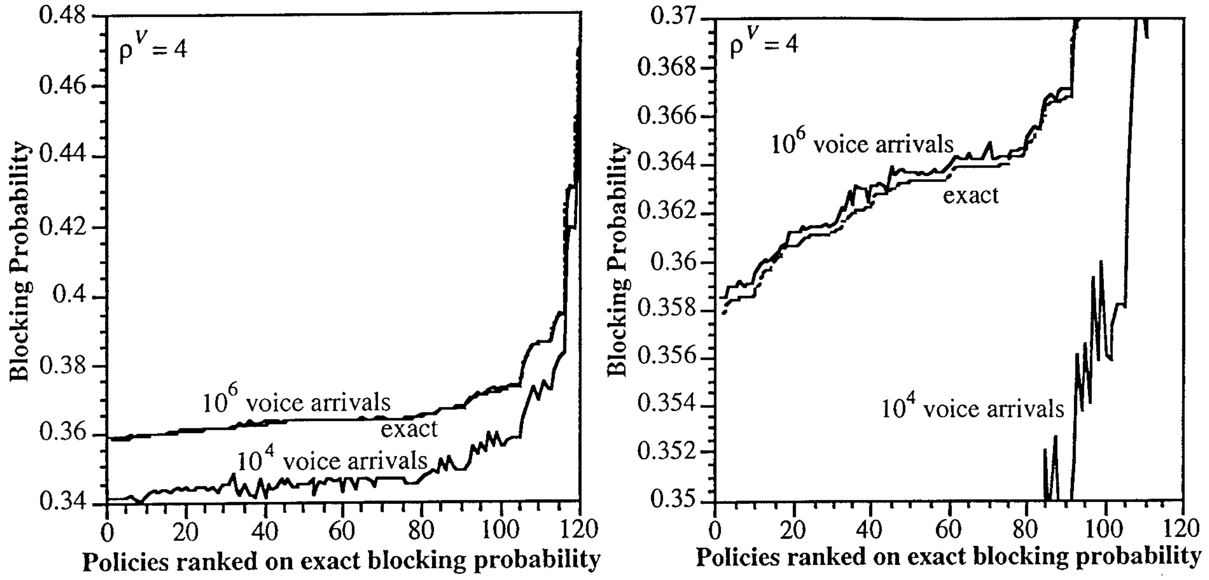
In [1, 2] we implemented ordinal optimization in three different ways. In the first we used relatively short simulation runs (as in [6]) to obtain a ranking of a number of policies in terms of voice-call blocking probability. We found that, although the measured performance may not be very accurate, the ranking of policies is relatively immune to the effects of "estimation noise." Our second approach, which we used for data-packet delay, was the use of crude analytical models that capture the crucial aspects of system behavior. Again, remarkably accurate policy rankings were achieved (in this case without using simulation at all), despite the fact that the performance estimates are poor. Our third approach, again for data-packet delay, was the use of imprecise simulation models that, like the crude analytical models, incorporate the salient features of the communication model. Before addressing the results of our new studies on the CM-5E, we first review some of the ordinal-optimization results that were presented in [1, 2].

7.1 Performance Evaluation and Ordinal Optimization

We performed SC simulations of the network shown in Fig. 5.2 by using the model discussed in Section 4. Eight transceivers are assumed present at each node in the network. Recall from Section 3.2 that a policy for this network can be written as $\Omega = \{X_1, \ldots, X_5, Y_1, \ldots, Y_5\}$. We have simultaneously evaluated the 120 different control policies $\{X_1, 6, 6, 6, X_5, 8, 8, Y_3, 8, 8\}$, where $X_1, X_5 = 0,\ldots 6$; $Y_3 = 0,\ldots,8$; $Y_3 \leq X_1 + X_5$; $X_1 \leq Y_3$; and $X_5 \leq Y_3$. We have used various values of $\lambda_j^V = \lambda^V$, $\mu_j^V = \mu^V$, $j = 1, \ldots, 5$, all of which are subject to $\lambda^V/\mu^V = \rho^V = 4.0$. To assess the effectiveness of our simulation techniques, we have evaluated the accuracy of the simulation-based values of the desired performance measures, as well as the accuracy of the corresponding ordinal rankings of the policies.

7.1.1 Evaluation of Voice-Call Blocking Probability

In Fig. 7.1(a) we show the exact and simulated voice-call blocking probability associated with the 120 control policies. The exact results are determined numerically from the product-form solution, and simulated results are based on runs of 10^4 and 10^6 voice-arrival events. Figure 7.1(b) is an expanded view of the same curves, which permits a more-detailed comparison of the results. As noted earlier, the results for blocking probability are independent of data-traffic parameters in integrated voice/data networks; in fact, they do not depend on the individual values of λ_j^V and μ_j^V, but only on the ratios $\rho_j^V = \lambda_j^V/\mu_j^V$. The horizontal axis is simply the ordering from the best (minimum blocking probability) policy to the worst, based on the exact model; thus blocking probability is a monotonically nondecreasing function of the horizontal axis. It is apparent from the closeness of the curves in Figs. 7.1(a) and 7.1(b) that the results of the longer simulation are extremely accurate; the simulation error is never more than 0.2%. It is also apparent that the shorter simulation is not long enough to predict blocking probability accurately.



(a) the complete picture

(b) an expanded view

Fig. 7.1 — Probability of blocking across the range of policies

7.1.2 Ordinal Ranking of Policies

In Fig. 7.2 we compare ordinal rankings of the voice-call blocking probability obtained from two SC simulations to the exact rankings. For these studies we used $\lambda^V = 4$, and $\mu^V = 1$. The two SC simulations differed only in their durations, which were based on 10^4 and 10^6 voice-call arrivals. The ordinal ranking of the simulation results shows remarkable agreement with the exact ordinal ranking (ideally the curve would be a straight line with unit slope). For long simulations this is not surprising; however, it is remarkable that the short simulation placed eight of the top ten policies in the top ten positions.[10] This agreement is achieved despite the insensitivity of blocking probability to the policy used. For example, as the policies are examined from the best to the 80th (in a total of 120), the blocking probability increases by only a small amount, i.e., from 0.358 to 0.365.

7.2 Observations on the Use of Ordinal Optimization

The results of this section provide evidence of the potential value of ordinal-optimization methods. When we are interested in optimizing performance, the ordinal ranking of the different policies (i.e., the ranking from best to worst) is much more informative than the actual measure of performance under a given policy. In examples of integrated networks presented in [1, 2] it was shown that a crude analytical model that provides extremely inaccurate estimates of delay does very well in selecting the best control policies with respect to data-packet delay. For example, although the time constants associated with voice (λ_j^v and μ_j^v) have a major impact on data-packet delay, they only minimally affect the policy ranking. We feel that it is especially

[10] In [40] we discussed the use of the Spearman rank correlation coefficient [41] to provide a quantitative measure of the accuracy of ordinal rankings, and we presented plots demonstrating the improvement of this parameter (whose value is equal to 1.0 for an exact ordering) as the duration of the simulation increases.

significant that simple analytical models, such as the M/D/1 queue can provide highly accurate rankings.

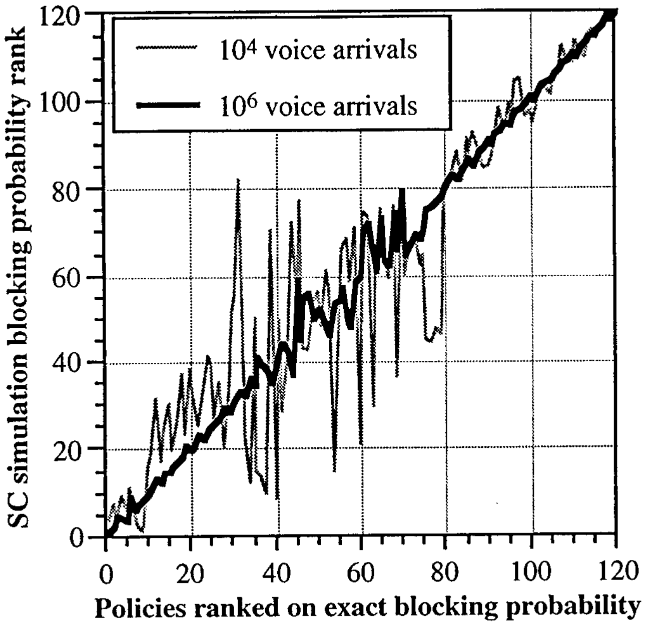


Fig. 7.2 — Blocking probability found in SC simulations compared to the exact value, $\rho^V = 4$

The remarkable similarity of the delay rankings obtained by using the different models may have interesting, and possibly far-reaching, implications in the study of optimization methods. These results suggest that accurate ordinal rankings can be obtained even when highly inaccurate models (in the sense of predicting actual performance values) are used. The principal requirement here appears to be the identification of the key aspects of the model that affect relative performance. In our examples the residual capacity available for data is of primary importance, whereas the time constants associated with voice traffic and the availability of receivers at destination nodes are relatively unimportant. Thus, simulation under one set of parameters provides a good indication of relative performance under other parameter sets (for the same values of utilization ρ_j^v).[11] Moreover, the accuracy of the rankings obtained by using the analytical approximation suggests that simulation may actually be unnecessary to determine the best policy or best set of policies. However, simulation will still be necessary to evaluate the system performance under these policies.

8 ORDINAL OPTIMIZATION IN LARGE NETWORK EXAMPLES

The availability of the Connection Machine CM-5E has permitted us to extend our study of Standard Clock ordinal optimization techniques to problems that are of dramatically larger size than those that could be addressed on sequential machines. Whereas with sequential machines our study of admission control in the network of Fig. 5.2 was limited to the case of eight transceivers per node, we have extended our study to network examples with as many as

[11] We have observed that this is true provided that the expected voice-call duration is greater than or equal to ten times the data-packet length.

4000 transceivers per node when using the CM-5E. The implementation of SC simulation on the CM-5E was discussed in Section 2.2.

In our first example, we consider a system in which the performance levels of 141,376 policies are compared. The computational burden of such an exhaustive search makes this approach impractical for many real-world examples. However, the results of this search (which will be presented shortly) have drawn us to the conclusion that such exhaustive search is generally not necessary, and that good policies can be found by carefully chosen non-exhaustive search methods. Furthermore, policies that perform quite well can usually be found on the basis of short simulation runs, as in the case of the smaller examples studied in Section 7. Thus, satisfactory performance can be achieved without the use of the CM-5E or other high-performance computer systems.

We have observed that the ordinal rankings obtained in this exhaustive search are not as accurate as those obtained in the SC simulation of smaller systems. However, we have also observed that many policies provide nearly identical performance; thus, even large errors in the ranking among policies that perform almost equally well do not impact adversely on the ability to determine nearly optimal policies. Indeed, our goal is to identify any policy whose performance is close to that of the optimal policy. In this section we examine in detail the structure of good admission-control policies, and address the issue of extrapolating conclusions from small systems to large ones. In summary, our basic conclusion is that good policies can be obtained on the basis of short simulations and with a far less than exhaustive search of all control policies.

8.1 Performance Evaluation via Parallel Simulation

Because of the huge number of possible admission-control policies in systems with a large number of transceivers, we have limited our search to the class of threshold-only policies, i.e., we do not consider the linear-combination controls discussed in Section 3.2. In Section 5 we studied the network of Fig. 5.2 for the case of $T_i = 8$ transceivers at each node. Exhaustive search of all policies showed that little improvement was achieved when linear-combination controls were included. Furthermore, it was observed that the thresholds on circuits 2, 3, and 4 were never reduced, i.e., (for a specified uniform offered load) the best policy was obtained by adjusting the thresholds on only circuits 1 and/or 5. We conjecture that similar behavior is exhibited by larger systems as well, although we have not been able to verify this assertion because of the computational requirements needed to do so. Thus, it appears sufficient to examine policies for which the thresholds X_2, X_3, and X_4 are kept at their maximum permissible value (which we arbitrarily set to 75% of the number of transceivers, i.e., $N_c = 0.75 \times N_t$, as discussed in Section 5), while the thresholds X_1 and X_5 are varied.

We consider the network of Fig. 5.2 with $N_t = 500$ transceivers per node; thus $N_c = 375$, and the number of policies to be examined in an exhaustive search of threshold policies (where X_1 and X_5 are varied while X_2, X_3, and X_4 are kept constant at 375) is $376^2 = 141{,}376$. The offered load on each of the five circuits is also expressed as a fraction of the quantity N_t. In this example, $\rho_1 = \rho_2 = \cdots = \rho_5 = (9/16) \times N_t = 281.25$. A two-step simulation study was performed as follows. First, a SC simulation based on 10^6 events was performed on the CM-5E for all 141,376 policies. A SC simulation based on 10^8 events was then run on an HP workstation for the 1000 best policies obtained in the exhaustive search of step 1.

It is of interest to compare the rankings achieved on the basis of short (10^6 events) simulation runs with those achieved on the basis of long (10^8 events) simulation runs. We introduce the term "*K* short-best policies" to refer to the *K* policies with lowest blocking probability determined on the basis of the simulation of 10^6 events. We also introduce the term "*K* long-best policies" to refer to the *K* policies with lowest blocking probability among the 1000 policies that were simulated for 10^8 events (i.e., the 1000 "short-best" policies found in a simulation of 10^6 events), as discussed above.

Figure 8.1 shows the voice occupancy distribution (i.e., the distribution of the instantaneous total number of active voice calls of any type) under the uncontrolled system, and under the 6 "short-best" policies[12] for simulations of duration 200,000 events, 10^6 events, and 10^8 events. Note that the distributions of the six best policies are virtually indistinguishable, even after only a 200,000-event simulation, but their distribution is quite different from that of the uncontrolled policy. Note that, on the average, a "good" admission control policy can improve the number of accepted calls by approximately 10% as compared to the uncontrolled system.

8.2 Quality of Ordinal Rankings

We now consider the accuracy of the ordinal rankings associated with the 500 "short-best" admission-control policies out of the 141,376 that were simulated. Figure 8.2(a) shows the blocking probability associated with these policies. Note that, although the 500 policies in the sample were chosen on the basis of rankings after a 10^6-event simulation, the horizontal axis shows the rankings of these policies after a 10^8-event simulation, which are assumed to be closer to the true rankings.[13] The performance estimates after the 10^8-event simulation are assumed to be essentially equal to the "exact" values of blocking probability. We see that the simulation that is based on 84.5×10^6 events produces estimates of blocking probability that are within 0.01% of the "exact" values, and the shorter simulation (i.e., the one based on 10^6 events) produces estimates that are accurate to within about 0.2% of the exact values.

Ordinal rankings of the 500 "short-best" policies are shown in Fig. 8.2(b), from which it can be seen that the $84.5 \text{x} 10^6$-event simulation produces highly accurate rankings (which is not surprising since the duration of the simulation is 84.5% of the longer one), whereas the 10^6-event simulation produces rankings that appear to be nearly randomly dispersed among the 500 policies. The poor quality of rankings obtained in the shorter simulation is a result of the relative insensitivity of blocking probability with respect to the chosen policy from within this set of policies. Since P_b varies by less than 0.0005 over this set of policies, it is not surprising that the rankings given by the short simulation are inaccurate. In fact, any of these 500 policies may be considered to provide acceptable performance since they all provide a blocking probability that

[12] For simulations of all three durations, these are the six "short-best" policies, i.e., the best six policies based on the 10^6-event simulation.

[13] These are the "true" rankings among only the set of the 500 "short-best" policies.

is within 0.2% of the minimum possible value. This behavior seems to suggest that the absolute ranking of policies should not necessarily be our goal. In problems for which many policies provide similar performance it may be sufficient to find policies that provide performance that is sufficiently close to the optimal.

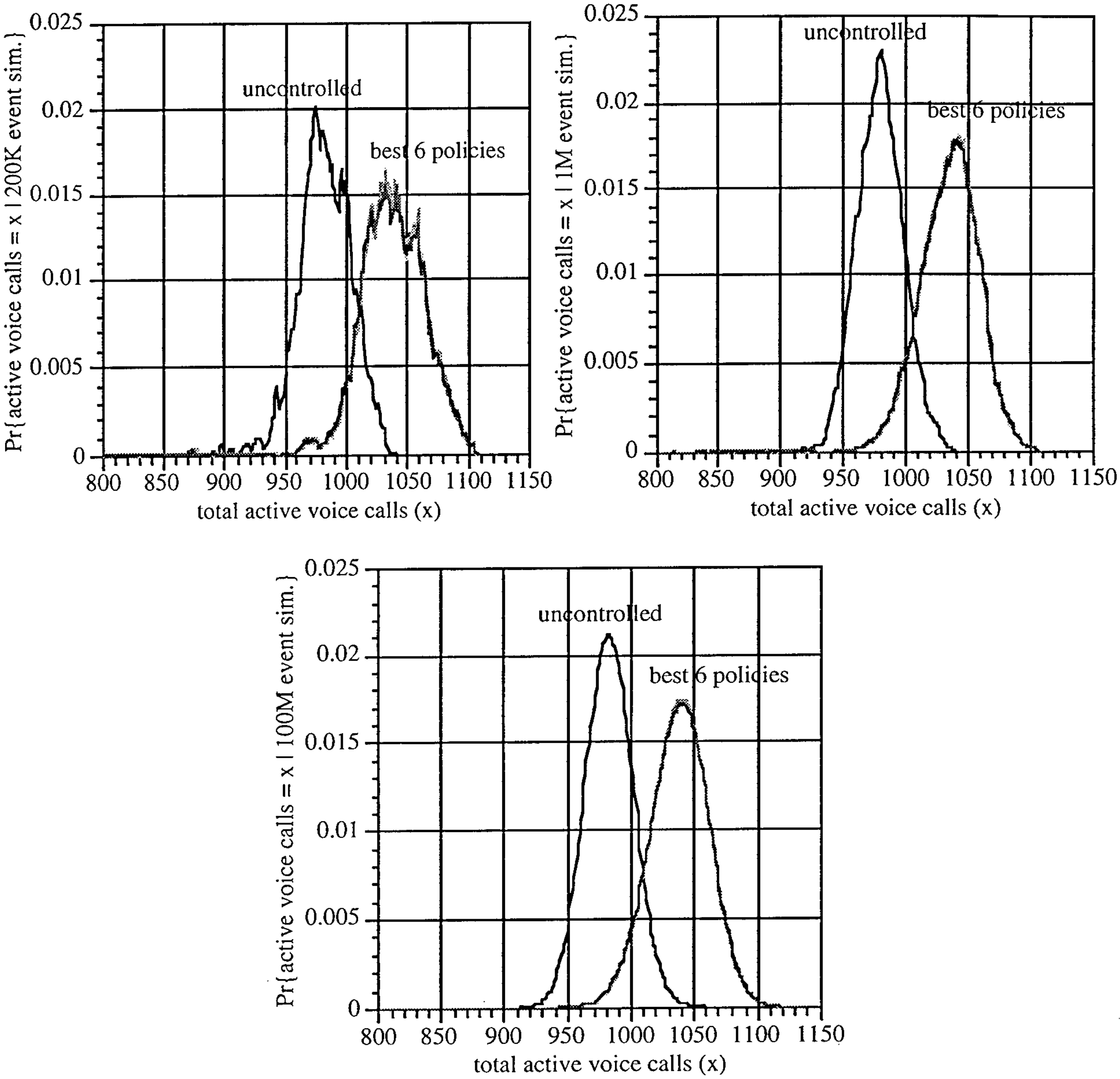
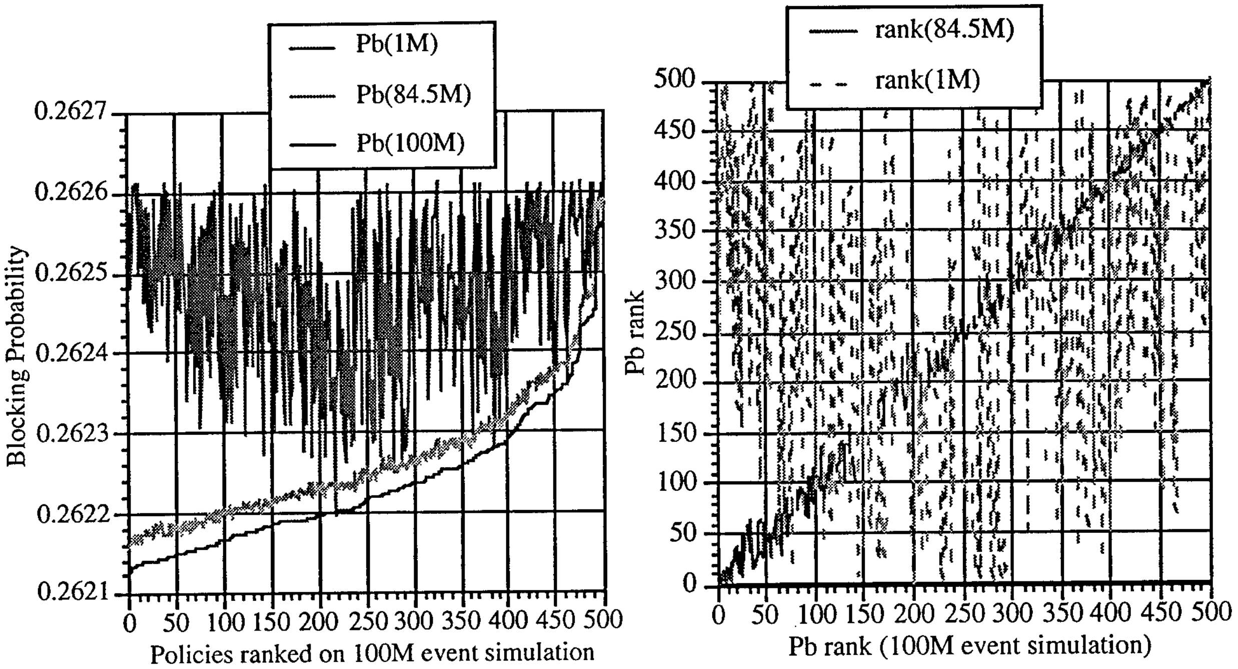


Fig. 8.1 — Voice occupancy distribution with increasingly longer simulations
($N_t = 500$, $N_c = 375$, "short-best" 6 policies based on 10^6-event simulation exhaustive search)

An important question is whether a 10^6-event simulation is sufficiently long to find the best policy, or at least to find a policy whose performance is sufficiently close to that of the best policy. In other words, we would like to determine whether significantly better policies were overlooked by the shorter simulation. To address this question, we now consider a 10^8-event simulation of the 1000 "short-best" policies determined by the 10^6-event simulation, i.e., in addition to the 500 "short-best" policies considered before, we now consider the next 500 best policies as well.



(a)

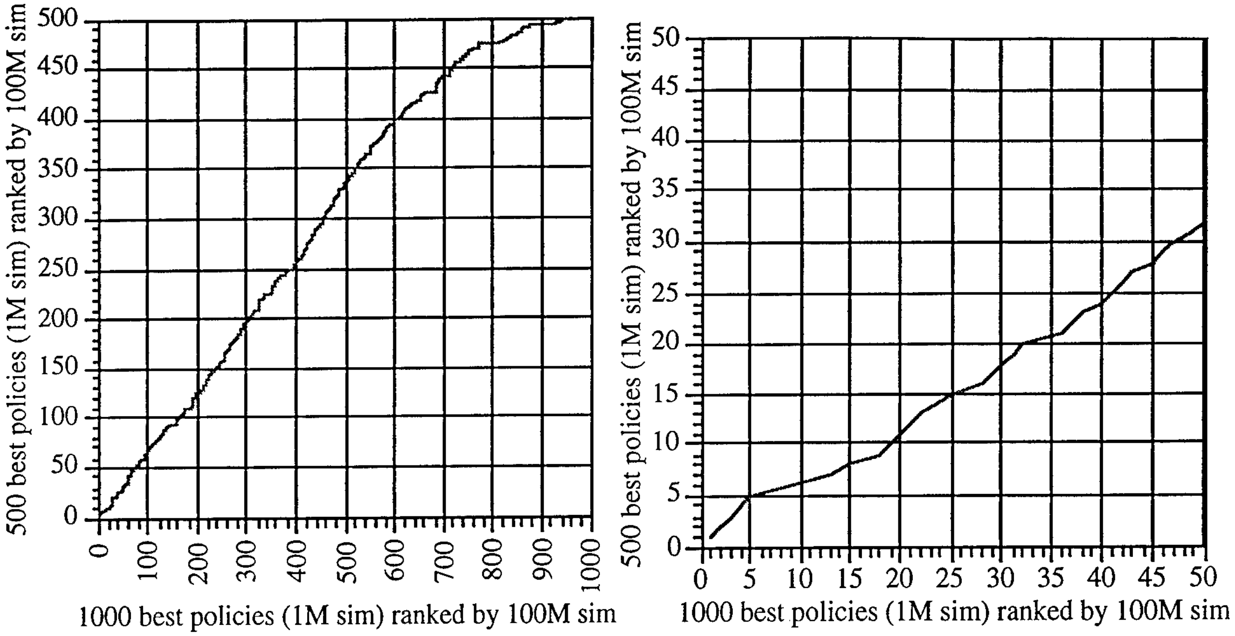
(b)

Fig. 8.2 — Blocking probability and rank vs results of 10^8-event simulation

The question is how many of the policies in the 500 "short best" remain in the 500 "long best" when the 1000 "short-best" policies are simulated in a run of 10^8 events.[14] The vertical axis of Fig. 8.3(a) indicates how many of the 500 "short-best" policies are among the set of X "long-best" policies, where X is the value on the horizontal axis. For example, 335 of the 500 "short-best" policies continue to be among the 500 "long-best" ones. Figure 8.3(b) shows an expanded view of the 50 "short-best" (of the 1000) policies. We see that 32 of the 500 "short-best" are among the 50 "long-best." In fact, all top five of the 1000 "long-best" are also in the 500 "short best."

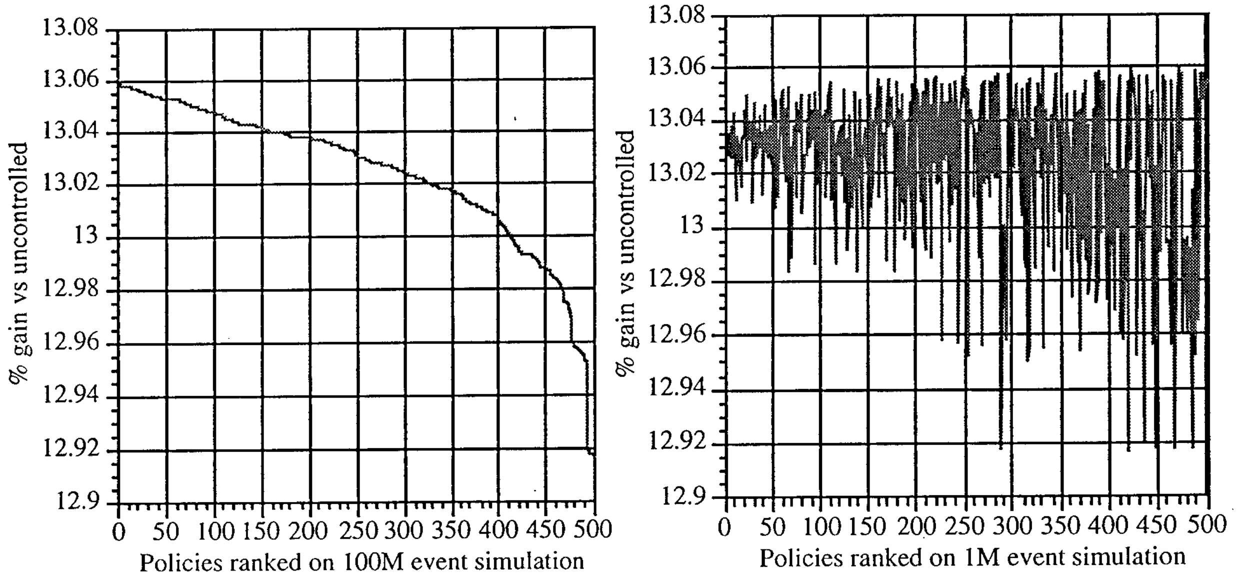
Fig. 8.4(a) shows that the performance improvement obtained by using any of the 500 "short-best" policies is approximately 13% as compared to that of the uncontrolled system. Both the rankings and the performance improvement are based on simulations of 10^8 events. Fig. 8.4(b) shows the performance improvement achieved by the same set of policies, except that they are ordered based on the rankings obtained in the shorter simulation. We observe that despite the inaccuracy of the ranking given by the 10^6-event simulation, the "long-worst" (i.e., of the 500 "best" policies the true worst are those with performance gain less that 12.94%) policies do show at the bottom half of the ranking.

[14] As noted earlier, the "short-best" policy refers to the policy that produces the minimum value of blocking probability in a run of 10^6 events, and "long best" refers to rankings (among only the 1000 "short-best" policies) obtained in the 10^8-event simulation.



(a) (b)

Fig. 8.3 — Rank (based on 10^8-event simulation) of 500 "short-best" policies vs rank (based on 10^8-event simulation) of 1000 "short-best" policies

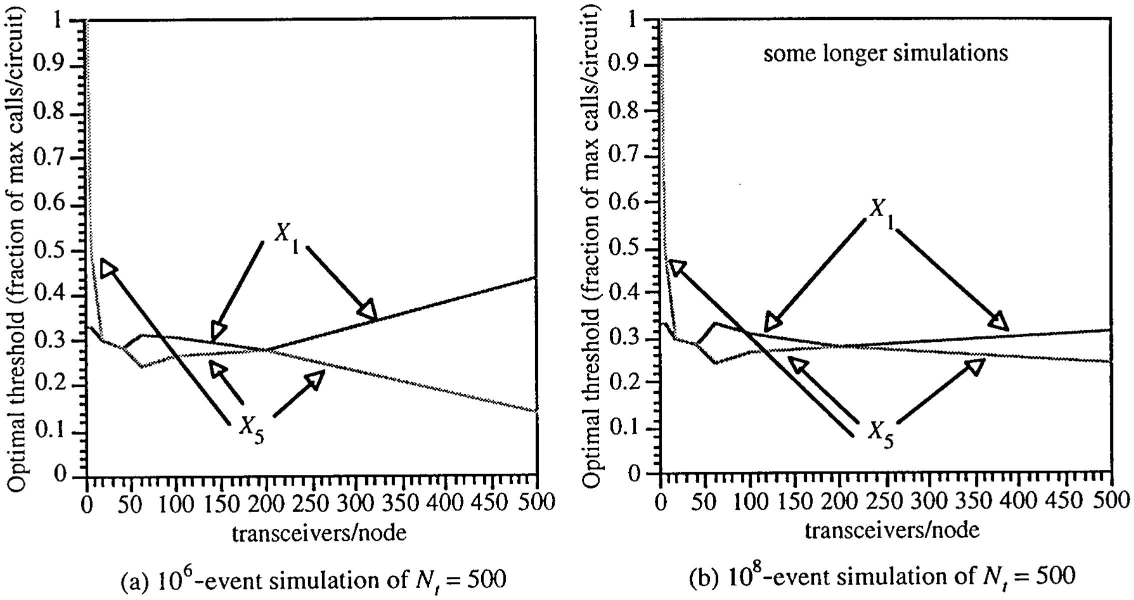


(a) (b)

Fig. 8.4 — Performance gain of "short-best" 500 policies

8.3 Policies Determined by Ordinal Optimization

We now shift our attention to the optimal policies determined by ordinal optimization, rather than to the performance of these policies. We noted earlier that in our network example the admission-control thresholds X_2, X_3, and X_4 should be kept at their maximum permitted values, while a search is conducted for the optimal values of X_1 and X_5. Figure 8.5 shows the optimal values of X_1 and X_5 (normalized with respect to the number of transceivers per node) as a function of the number of transceivers per node for an offered load of $\rho_1 = \cdots = \rho_5 = (9/16)N_t$; the maximum permissible value for each of the thresholds is equal to 3/4 of the number of transceivers at each node, i.e., $N_c = 0.75 \times N_t$. The cases of the 10^6- and 10^8-event simulations are shown in Figs. 8.5(a) and (b), respectively. For up to 200 transceivers at each node, the optimal thresholds determined in the shorter runs are close to those determined in the longer runs.[15] However, there are significant differences in the optimal threshold values for the case of 500 transceivers per node. This behavior might seem to cast doubt on the quality of the ordinal rankings produced by short simulations. However, we shall show that useful ordinal optimization results can be obtained when the observed performance is interpreted properly.



(a) 10^6-event simulation of $N_t = 500$ (b) 10^8-event simulation of $N_t = 500$

Fig. 8.5 — Optimal values of X_1 and X_5 and their average value vs N_t for $\rho = (9/16)N_t$

Figure 8.6(a) extends the domain of Fig. 8.5(a) by including results for 350 transceivers per node as well as for 1000, 2000, and 4000 (again, for the case of a 10^6-event simulation). At first, these results may seem discouraging because the optimal threshold fractions change drastically as the number of transceivers changes; in fact the normalized optimal value of X_1 is zero for the case of $N_t = 350$, and as high as 0.45 for $N_t = 500$. It seems as though the (normalized) optimal policy is very sensitive to the number of transceivers per node. However, an interesting pattern emerges when we look at the value of $(X_1 + X_5)/2N_t$. Figure 8.6(b) shows that the optimal value of this ratio is nearly constant for systems with more than about 40 transceivers per node.

[15] Analytical solutions are used for $N_t \leq 40$, so these results are identical in the two figures.

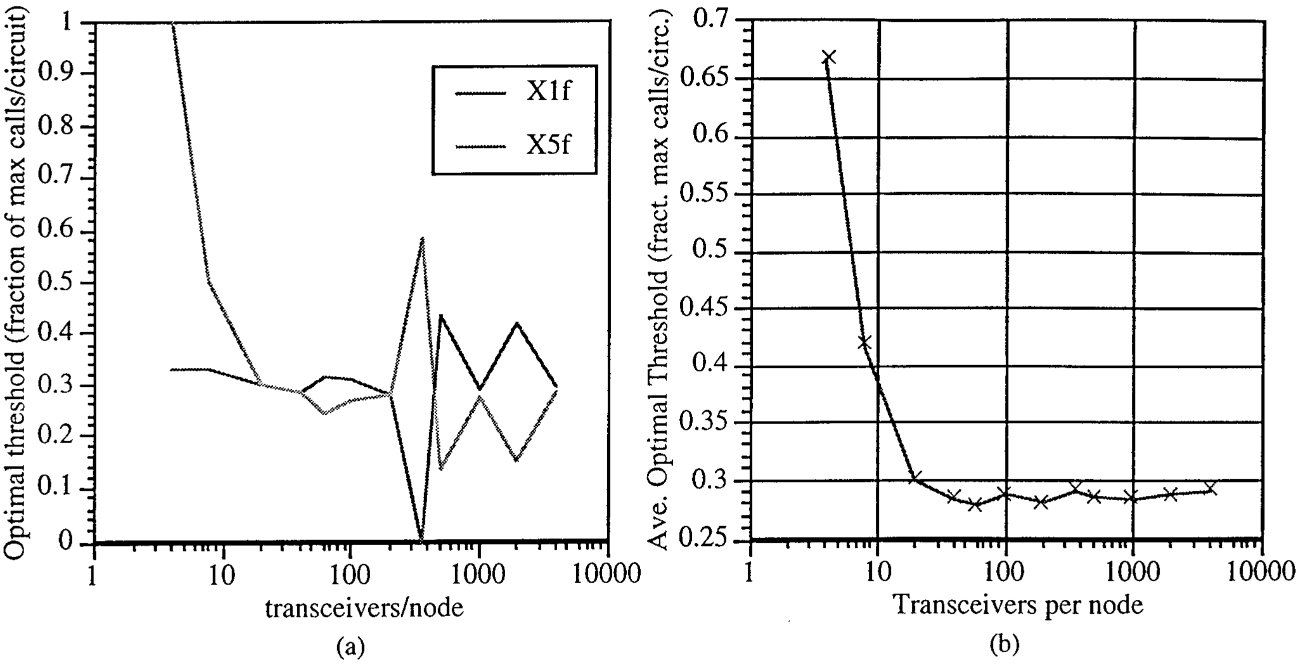


Fig. 8.6 — Optimal values of X_1 and X_5 and their average value vs N_t for $\rho = (9/16)N_t$

Figures 8.7 and 8.8 show similar results for the case of more-heavily loaded networks with $\rho = N_t$ and $\rho = 2N_t$, respectively. We see that, except for cases involving a very small number of transceivers per node, the value of the average of the optimal values of the normalized sum of thresholds X_1 and X_5 is virtually constant as the number of transceivers changes.

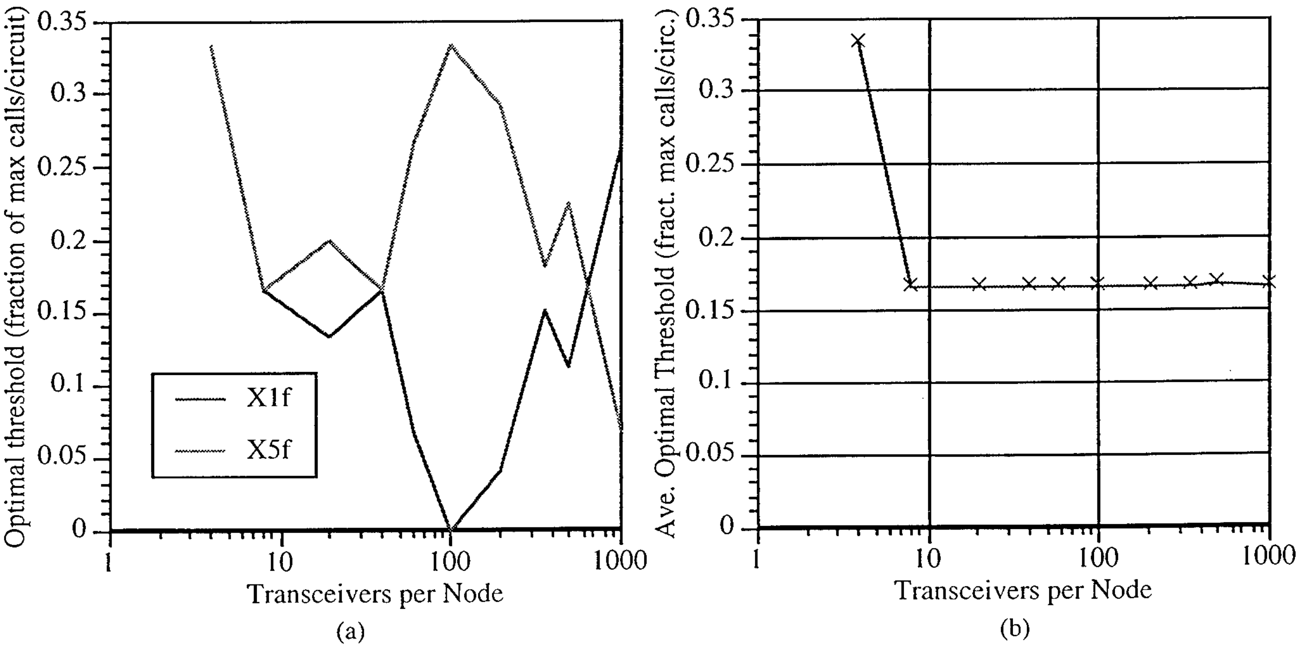


Fig. 8.7 — Optimal values of X_1 and X_5 and their average value vs N_t for $\rho = N_t$

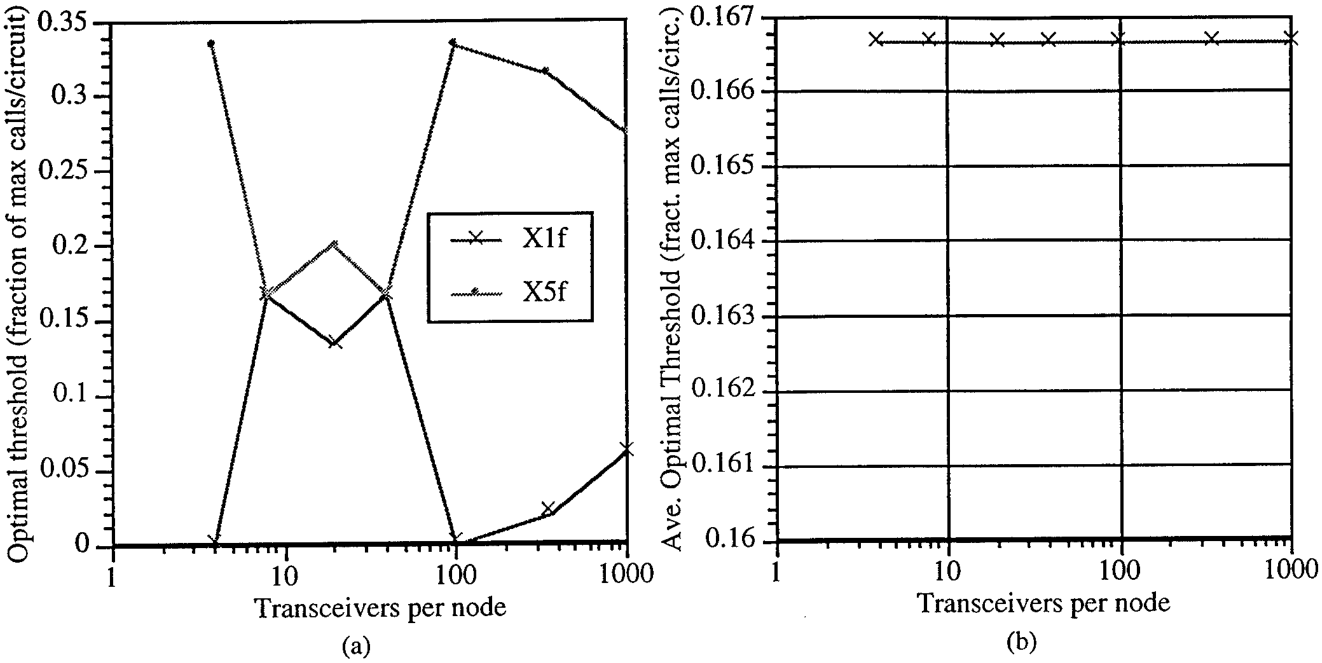


Fig. 8.8 — Optimal values of X_1 and X_5 and their average value vs N_t for $\rho = 2N_t$

8.4 An Interpretation of the Optimal Policies

In an attempt to understand the apparent sensitivity of the optimal values of X_1 and X_5 to N_t, we considered several alternative explanations. For example, the simplest interpretation of the above figures is that (normalized) system behavior actually changes drastically as the number of transceivers per node increases. If that were the case, a policy optimized for a given value of N_t (with thresholds normalized to the value of N_t) would be expected to perform poorly for other values of N_t. However, the fact that the optimal value of the sum $(X_1 + X_5)$ remains essentially constant for N_t greater than about 10 suggests that it may be helpful to examine this sum rather than the individual values of X_1 and X_5. Perhaps it is sufficient to pick (almost) any (X_1, X_5) pair with the appropriate sum.

In an attempt to evaluate this conjecture, we have examined the behavior of the normalized sum $(X_1 + X_5)/N_c$ as a function of policy ranking. In all cases, the maximum number of calls permitted on any circuit is $N_c = 0.75\ N_t$, where all nodes have the same number (N_t) of transceivers. Analytical results (based on the numerical evaluation of the product-form solution) are presented in Section 8.4.1 for the case of $N_t = 40$ transceivers per node, which is the largest system that we have been able to evaluate numerically by means of the product-form solution. In Section 8.4.2 simulation results (based on a simulation of duration 10^6 events) are presented for N_t = 100, 500, and 1000.

8.4.1 Analytical Results: $N_t = 40$

For $N_t = 40$, we set $N_c = 30$. Figure 8.9 shows the case of $\rho = N_t/4 = 10$, which corresponds to a relatively lightly loaded system. Two quantities are plotted vs policy rank (based on P_b), namely $(X_1 + X_5)/N_c$ and P_b. There are a total of $31^2 = 961$ policies, all of which are represented in this figure. The minimum value of blocking probability is $P_b = 0.013982$, which is obtained for 48 sets of (X_1, X_5) thresholds, including the uncontrolled case of (30,30). Actually the policies (24, 25), (25, 24), and (≥25, ≥25) all provide the same blocking probability (to within 10^{-6}). The fact that each of the best 48 policies has the rank of 1 explains the gap in the left part of the curve in Fig. 8.9. In this example, any policy with $(X_1 + X_5)/N_c \geq 1.8$ will provide optimal performance (again, to within 10^{-6}).

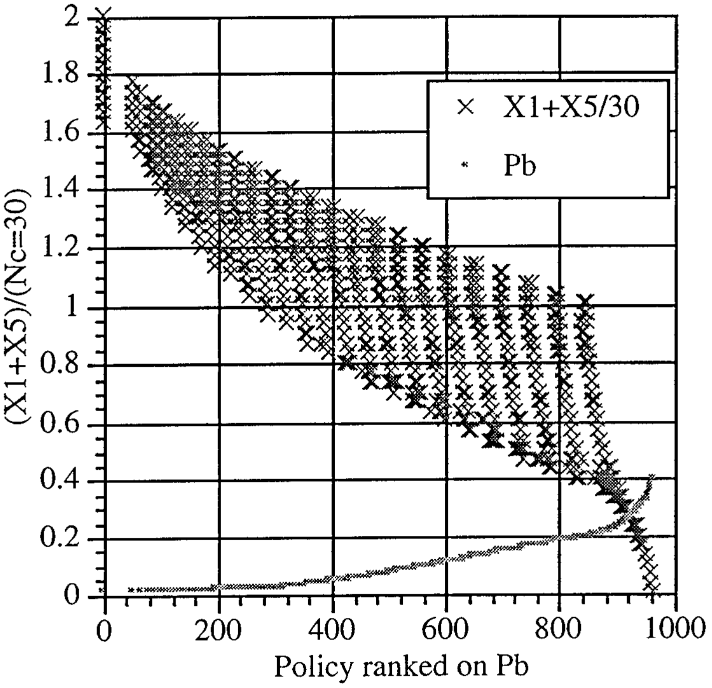


Fig. 8.9 — Normalized value of $(X_1 + X_5)$ vs policy ranking: Analytical result ($N_t = 40$, $N_c = 30$, $\rho = 10$)

Figure 8.10 shows the case of $\rho = (9/16)N_t = 22.5$. Two policies, namely $(X_1, X_5) = (8,9)$ and (9,8) (hence $X_1 + X_5 = 17$ and $(X_1 + X_5)/N_c = 0.5666$), result in the minimum value of blocking probability, which is $P_b = 0.309405$. The next best solutions, (10,7) and (7,10) (again $(X_1 + X_5) = 17$) result in $P_b = 0.30944$. Typically, however, P_b is not symmetrical in X_1 and X_5. As shown in Table 8.1 there are 18 policies for which the value of $(X_1 + X_5)/N_c$ is equal to 0.5666, including the five best policies. However, we also see that $(X_1 + X_5) = 17$ for the 378th policy, thus indicating that the value of $(X_1 + X_5)/N_c$ is not sufficient to predict the ordinal ranking of the solution. On the other hand, the ratio of P_b for this 378th policy to that of the best policy is 1.049, indicating that little is lost by not picking the best of the policies with $(X_1 + X_5) = 17$.

Actually, there is little variation in P_b over the best approximately 900 policies. In this example, the uncontrolled system (for which $X_1 = X_5 = 30$) results in a blocking probability of $P_b = 0.330224$, which corresponds to a rank of 815 (actually 72 policies are tied for this rank) out of the total of 961 policies. The fact that the uncontrolled policy provides a blocking probability only 1.067 times that of the best policy suggests that some degree of self-regulation is present here. Returning to the question of how well the value of $(X_1 + X_5)/N_c$ can predict the quality of the solution we make the following observation. Use of the criterion that $(X_1 + X_5)/N_c = 0.5666$, coupled with the requirement that the solution be nearly symmetrical, does produce a nearly optimal solution.

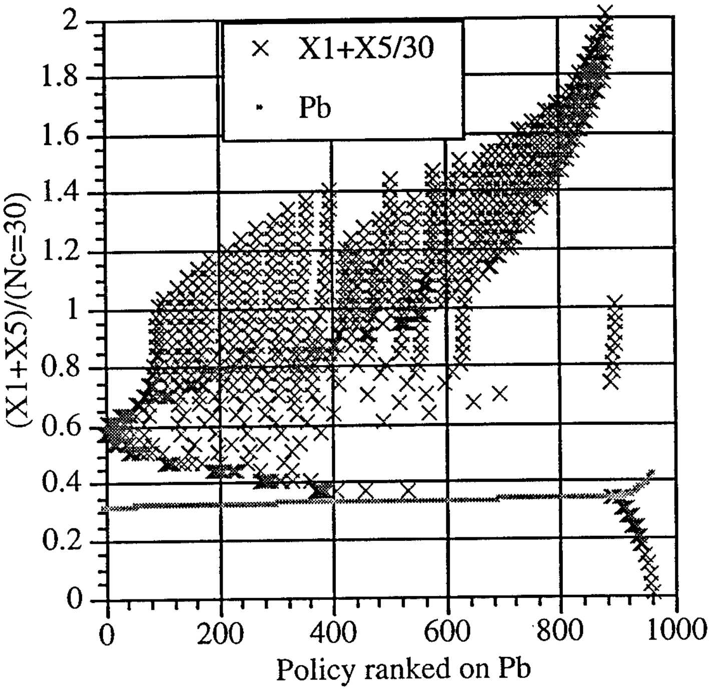


Fig. 8.10 — Normalized value of ($X_1 + X_5$) vs policy ranking: Analytical result ($N_t = 40$, $N_c = 30$, $\rho = 22.5$)

Table 8.1 — Admission-Control Policies Corresponding to $X_1 + X_5 = 17$ ($N_t = 40$, $N_c = 30$, $\rho = 22.5$)

X_1	X_5	rank	P_b
8	9	1	0.309405
9	8	2	0.309405
10	7	3	0.30944
7	10	4	0.30944
6	11	5	0.309513
5	12	7	0.309627
4	13	12	0.309791
3	14	20	0.310013
2	15	24	0.310307
11	6	26	0.310341
1	16	38	0.310689
0	17	49	0.311179
12	5	69	0.3118
13	4	133	0.313692
14	3	192	0.315956
15	2	248	0.31855
16	1	312	0.321439
17	0	378	0.324595
uncontrolled: 30	30	815	0.330224

We demonstrate in Section 8.4.2 that the optimal value of $(X_1 + X_5)/N_c$ in network examples with a much larger number of transceivers per node (and the same normalized offered load) is almost identical to that observed for the current example with $N_t = 40$; such behavior is indicated by Figs. 8.6(b), 8.7(b), and 8.8(b). This behavior suggests that optimal admission-control policies tend to scale almost linearly with N_t (for a sufficiently large initial value of N_t, which seems to decrease as the offered load increases). Thus, nearly optimal solutions for large systems can be obtained by scaling the solutions obtained for considerably smaller systems.[16] Certainly, additional study is needed before definitive conclusions can be made in this regard. However, the results obtained here seem to indicate that simple heuristics may indeed provide nearly optimal admission-control policies.

Figure 8.11 shows the case of $\rho = N_t = 40$. In this example, the 11 best policies (and no others) are all characterized by $(X_1 + X_5) = 10$ (hence $(X_1 + X_5)/N_c = 0.3333$), as shown in Table 8.2. We also see that, among these policies, the more-symmetrical ones perform slightly better. Thus, although use of the optimal value of $(X_1 + X_5)$ does not guarantee the optimal solution (since there are several policies of this type), all of these policies perform extremely well, and no policy with a different value of $(X_1 + X_5)$ performs better than any of them. Here, the uncontrolled policy has $P_b = 0.566237$ (as in the case of $\rho = 22.5$, 1.067 times that of the best policy) and a rank of 849 (actually tied with 107 policies, so only six are worse).

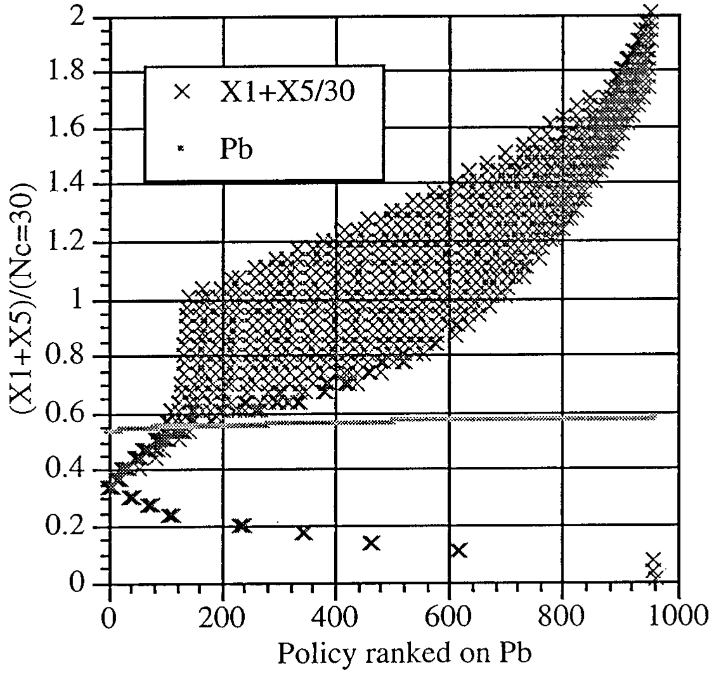


Fig. 8.11 — Normalized value of $(X_1 + X_5)$ vs policy ranking: Analytical result ($N_t = 40$, $N_c = 30$, $\rho = 40$)

[16] However, we showed in Section 6.2 that system performance (e.g., blocking probability) improves as the number of transceivers increases (for a constant normalized offered load) because of the increased statistical multiplexing capability offered by a large number of transceivers.

Table 8.2 — Admission-Control Policies Corresponding to $X_1 + X_5 = 10$
($N_t = 40$, $N_c = 30$, $\rho = 40$)

X_1	X_5	Rank	P_b
5	5	1	0.530928
4	6	2	0.530936
6	4	3	0.530936
3	7	4	0.530959
7	3	5	0.530959
2	8	6	0.530997
8	2	7	0.530997
1	9	8	0.531051
9	1	9	0.531051
0	10	10	0.531122
10	0	11	0.531122
uncontrolled: 30	30	849	0.566237

8.4.2 Simulation Results

The examination of systems with $N_t > 40$ requires simulation, which was performed on the CM-5E by using Standard Clock techniques. In this subsection we examine the behavior of the quantity $(X_1 + X_5)/N_c$ for systems with $N_t = 500$ and 1000 transceivers per node.

In Fig. 8.12 we plot the blocking probability and the normalized sum $(X_1 + X_5)/N_c$ for the 1000 "short-best" policies (i.e., based on a 10^6-event simulation) of the case of $N_t = 500$ transceivers per node. The offered load on each circuit is $\rho = (9/16)N_t = 281.25$, and the maximum number of calls of any single type permitted to be active simultaneously is $N_c = .75(N_t) = 375$. Because of the large number of admission-control policies to be considered ($376^2 = 141{,}376$) we examined only every ninth policy (hence a total of 15,708).[17] Figure 8.12 shows that the blocking probability is virtually constant over this range of policies; the slope is 5.53×10^{-7}. Therefore, any of the best 1000 policies provides essentially the same level of performance. Also, the normalized sum $(X_1 + X_5)/N_c$ does not vary significantly over the set of 1000 best policies. Its value for the best policy is 0.568, which is almost the same as that observed for the analytical/numerical solution of 0.5666 for $N_t = 40$.

For this example, since we actually performed an exhaustive search of all 141,376 policies as well as a "uniform" sampling of every ninth policy, we have been able to verify that none of the omitted policies actually provided significantly lower blocking probability than the best of these 1,000 policies.

[17] Consider the set policies to be described by a 376 × 376 array, and sample every ninth policy, starting from (0,0), with wraparound at the end of each row. Thus the sequence of policies examined was (X1,X5) = (0,0), (0,9), (0,18),...(0,369), (1,2), (1,11), ...

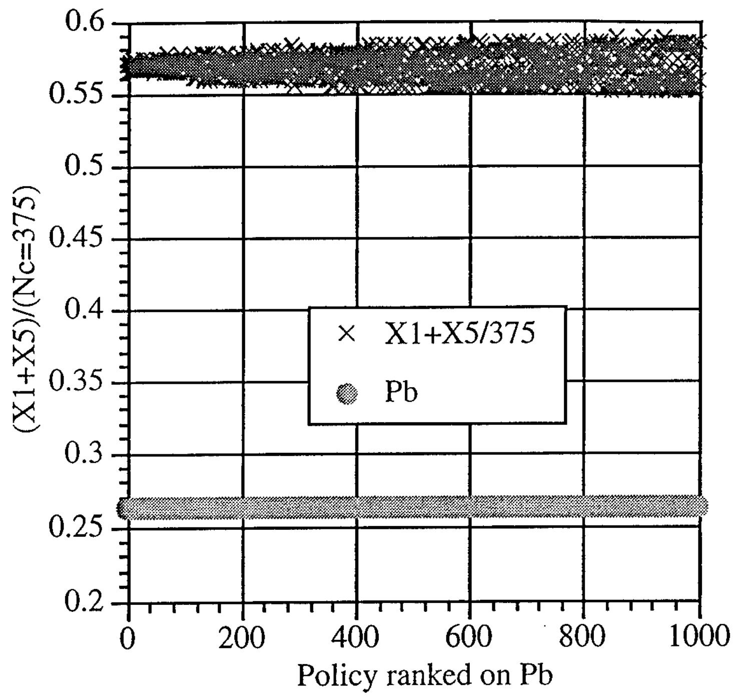


Fig. 8.12 — Normalized value of $(X_1 + X_5)$ vs policy ranking: 10^6-event simulation, 1000 best policies sampled every 9th policy ($N_t = 500$, $N_c = 375$, $\rho = (9/16)N_t = 281.25$)

The reason that the uniform sampling of policies works well in this example is that there is little sensitivity of the blocking probability with respect to the admission-control thresholds. Thus, "neighboring" policies (i.e., policies obtained by increasing or decreasing X_1 and/or X_5 by a small number) have nearly identical performance. In addition, "simulation noise" (especially in shorter simulations) can perturb the policy rankings significantly, even though the actual difference in performance is small. Thus, rankings can be poor, even though performance is good. The large number of possible policies creates a "finely quantized" control-policy space; thus, an incremental change in threshold of one call has a much smaller effect on system performance for the case of $N_c = 375$ than it does for the case of $N_c = 8$. A question for future study is the level of denseness of the sampling that is needed to avoid missing significantly better solutions.

Figure 8.13 shows the results of a longer (10^8-event) simulation of the same example shown in Fig. 8.12. This simulation was carried out for only the 1000 policies that were identified as the 1000 "short best" in the 10^6-event simulation (the horizontal axis is based on the rankings produced by the longer simulation). Results are similar to those for the shorter simulation, except that the values of the normalized sum $(X_1 + X_5)/N_c$ are confined to a somewhat narrower band, apparently because of the reduced variance of performance estimates that result from the longer simulation. The six best solutions (as well as many others) have $(X_1 + X_5) = 212$ (hence $(X_1 + X_5)/N_c = 0.5653$).

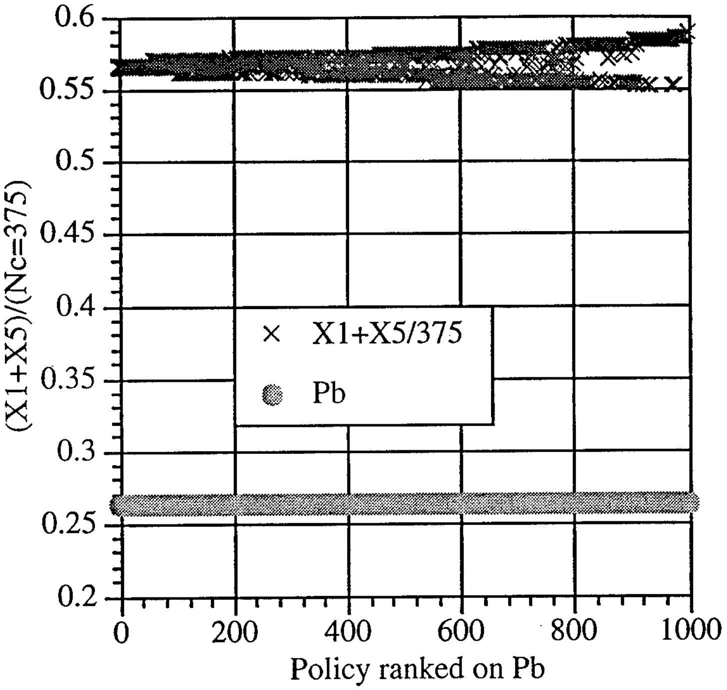


Fig. 8.13 — Normalized value of $(X_1 + X_5)$ vs policy ranking: 10^8-event simulation of 1000 best policies found in 10^6-event simulation ($N_t = 500$, $N_c = 375$, $\rho = (9/16)N_t = 281.25$)

Figure 8.14 shows similar results for the case of a heavier offered load, namely $\rho = N_t = 500$. The duration of the simulation is 10^6 events, and again every ninth policy is sampled. For this increased offered load, there is a greater sensitivity of blocking probability to the admission-control policy. Most interesting, perhaps, is the "V-shape" defined by the scatter plot of the normalized sum $(X_1 + X_5)/N_c$. The 43 best policies are all characterized by $(X_1 + X_5) = 126$, and the simulated value of P_b ranges from .501995 to .502618 over this set of 43 policies. The corresponding value of $(X_1 + X_5)/N_c$ is 0.336, which is nearly identical to the value of 0.333 obtained for the case of $N_t = 40$. Performance degrades as $(X_1 + X_5)$ is either increased or decreased. In fact, any policy with $(X_1 + X_5)/N_c$ close to its optimal value will provide nearly optimal performance.

Similar behavior is observed for the case of $N_t = 1000$ transmitters per node (in which we sample every 77th policy), as shown in Figs. 8.15 and 8.16. Again, as the offered load increases, the blocking probability becomes more sensitive to the choice of policy and the spread of the "V-shaped" curve increases. For the case of $\rho = (9/16)N_t$ it is clear that the optimal solution is provided when the normalized sum $(X_1 + X_5)/N_c$ is about 0.57, which is the same value that produced optimal performance for $N_t = 500$. Also, for the case of $\rho = N_t$ the optimal value of the normalized sum is 0.335, which is equal to that for the case of $N_t = 500$. The curves for $N_t = 1000$ are, in fact, quite similar to those produced for $N_t = 500$.

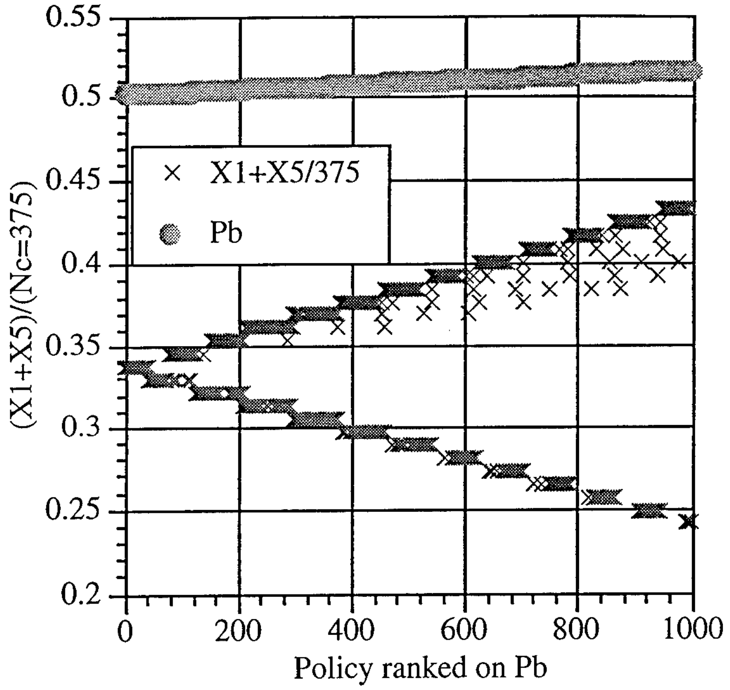


Fig. 8.14 — Normalized value of $(X_1 + X_5)$ vs policy ranking: 10^6-event simulation, sampled every 9th policy ($N_t = 500$, $N_c = 375$, $\rho = N_t = 375$)

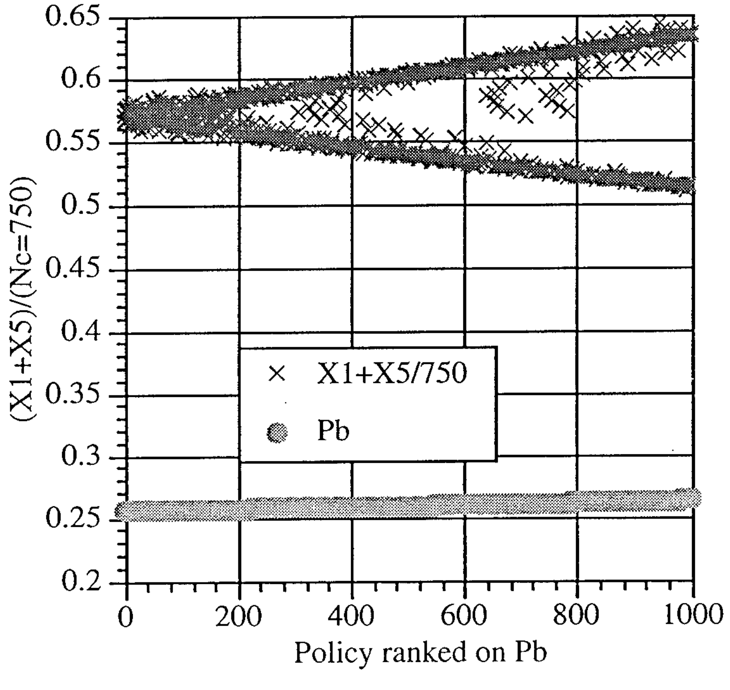


Fig. 8.15 — Normalized value of $(X_1 + X_5)$ vs policy ranking: 10^6-event simulation, sampled every 77th policy ($N_t = 1000$, $N_c = 750$, $\rho = (9/16)N_t = 562.5$)

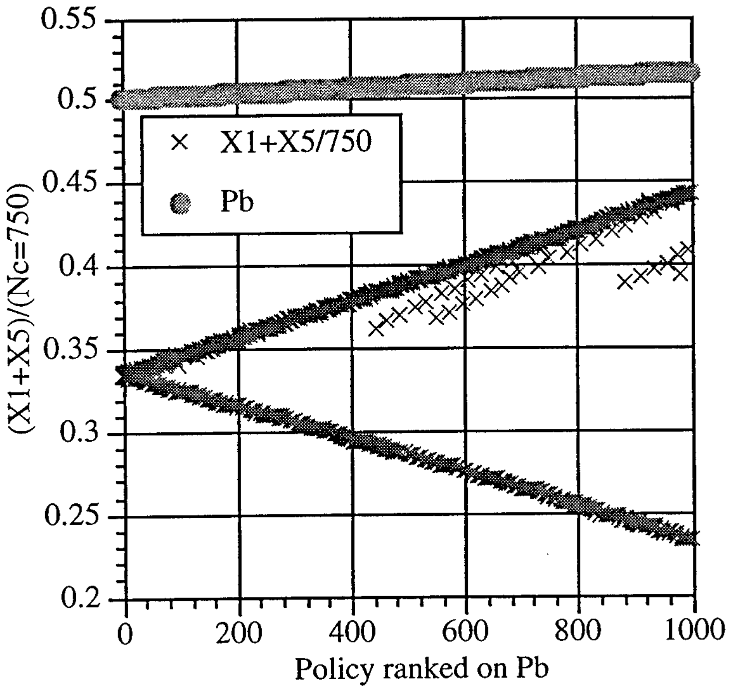


Fig. 8.16 — Normalized value of $(X_1 + X_5)$ vs policy ranking: 10^6-event simulation, sampled every 77th policy ($N_t = 1000$, $N_c = 750$, $\rho = N_t = 1000$)

Our conclusion from these curves is that, when the system is severely overloaded (e.g., $\rho \geq N_t$), there appears to be a unique value of $(X_1 + X_5)/N_c$ that provides optimal performance. At lower levels of offered load, it appears that some policies with $(X_1 + X_5)/N_c$ equal to the value that produces the optimal solution may not be among the top few policies. However, in such cases, there is little difference in performance among the several hundred or so best policies, and, in fact, even the uncontrolled policy provides close to optimal performance. Thus admission control is needed only when the system is very heavily loaded.

9 SUMMARY AND CONCLUSIONS

The motivation underlying ordinal optimization is that it is often appropriate to "soften" the goal of determining the truly optimal solution (a process that may be unacceptably costly or time consuming), and replace it with the objective of determining a policy that performs "sufficiently well." In earlier studies we demonstrated the power of ordinal optimization techniques, used in conjunction with Standard Clock (SC) simulation techniques, to rapidly determine nearly optimal solutions to complex network-control problems. These techniques can potentially save several orders of magnitude of computer processing expense, because the correlation among sample paths (i.e., different policies) introduced by the SC method permits the determination of highly accurate policy rankings after short simulation runs. Our earlier studies were based on the use of sequential machines. In this report we have discussed our extension of ordinal optimization techniques to the massively parallel Thinking Machines Corp. Connection Machine CM-5E.

The SC simulation technique is ideally suited to parallel processing, and the use of the CM-5E has permitted the study of considerably larger problems than were possible on sequential machines. For example, whereas our studies on sequential machines were typically limited to networking examples with up to $N_t = 8$ transceivers per node, the use of the CM-5E has permitted the study of examples with up to 4,000 transceivers per node, thus permitting the study of examples with the dimensions of high-speed networks. Additionally, the CM-5E has been used for the analytical evaluation of product-form solutions that were too complex for sequential machines. By studying large problems such as these, we have been able to improve our understanding of the ordinal optimization methodology.

In our networking problems, as the number of transceivers per node N_t increases, the search for the truly optimal admission-control policy becomes more difficult for several reasons: (1) the number of candidate policies that must be examined increases; (2) the number of policies that provide nearly optimal performance also increases; and (3) the complexity of performance evaluation for each of them increases as well. For our example network of Fig. 5.2, the largest value of N_t that can be handled numerically on the CM-5E is approximately $N_t = 40$; SC simulation has been used for values of N_t as large as 4000.

We have studied the effect of increasing the value of N_t, while maintaining a constant normalized network load. Our earlier studies (for values of N_t as high as 8) had shown that our network examples tended to be self-regulating, in the sense that the use of the optimal threshold admission-control policy provides little improvement over the performance of the uncontrolled system, in which calls are accepted as long as resources are available. However, our studies on the CM-5E have shown that as N_t increases the degree of improvement that is achieved through admission control increases, especially at moderate network loads, but less so at low loads and high loads (see Section 6.2). Thus, admission control is potentially more effective in high-capacity networks than in low-capacity networks such as wireless networks. It is also interesting that the blocking probability decreases as N_t increases, apparently because of the increased statistical multiplexing capabilities that are provided by a large number of transceivers. This effect also appears to be most significant at moderate network loads.

We have also observed that nearly optimal solutions for networks with a large number of transceivers at each node can be obtained by scaling the solutions obtained for the same example but with a smaller number of transceivers (and the same normalized load), thereby reducing significantly the computational burden. This is true even though the blocking probabilities of the small example may be significantly different from those of the large example, as just discussed. Thus, the small example can be used to pick tentative solutions (admission-control policies), and an accurate performance evaluation of the larger example can be obtained by performing a simulation of a policy based on the scaling of the best (or several best) policies obtained for the small system.

We have observed that the quality of ordinal policy rankings for large values of N_t, obtained on the basis of relatively short simulations, is not good. This behavior is in marked contrast to the remarkably accurate rankings obtained in earlier studies for small values of N_t. The deterioration of the quality of rankings can be attributed to the finer quantization of the specification of the admission-control policy that is possible for large values of N_t. For example, for $N_t = 1000$ a change in threshold value of 1 corresponds to only a 0.1% change as a fraction of N_t, whereas for $N_t = 8$ such a change corresponds to 12.5%. As a consequence of this fine quantization, there is little difference in performance between neighboring policies, and, in fact, many policies provide nearly identical performance. Therefore, "simulation noise" can produce significant deviations in the observed policy rankings. Hence, the quality of rankings observed for large values of N_t is not as good as that observed for smaller values of N_t.

The difficulties associated with large values of N_t are apparent in our examination of the properties of the admission-control policies determined by ordinal optimization (see Section 8.3). The optimal values of the admission-control thresholds X_1 and X_5 seemed to vary dramatically as either N_t was varied (suggesting poor scalability properties) or as the length of the simulation was changed (suggesting that short simulation runs could not provide a good choice of control policy). However, it was realized that similar performance was obtained for many different values of these threshold parameters, and that the value of the sum $X_1 + X_5$ (one of the linear-combination controls) is of greater significance to system performance than either value individually. All of the best solutions (for sufficiently large values of N_t — moderate values such as 10 were typically large enough) were characterized by the same value of this sum, and the best value of this sum increased in direct proportion to N_t. However, the use of the correct sum alone was not sufficient to guarantee optimal performance; in our problem it was helpful to add the additional constraint that the solution be close to symmetrical, i.e., X_1 and X_5 should have similar values. Further research is needed to reach a better understanding of the role of linear-combination controls in problems like these. However, at this point it is conjectured that their use may be very helpful.

In view of the insensitivity of system performance to the admission-control parameters when N_t is large, it is necessary to emphasize that the goal of ordinal optimization should not necessarily be to find an accurate ranking of policies, but rather to find a policy that provides nearly optimal performance. In particular, we have shown in Section 8.4.2 that it is possible to sample the policy space uniformly without significant loss of accuracy, in the sense that no policies providing significantly better performance were missed. Future research is needed to determine how fine a sampling of the policy space is needed to avoid the loss of good solutions. The approach of sampling the policy space, rather than performing an exhaustive search, is certainly not a new one. In fact, the original study of ordinal optimization [6] discusses the likelihood of finding good solutions when a very large policy space is randomly sampled, an approach that makes use of concepts from order statistics [36].

In conclusion, the CM-5E, which is ideally suited to SC simulation, has made it possible to investigate the ordinal optimization properties of very large network examples that could not be studied with conventional sequential machines. For example, this study has provided insight into the use of ordinal optimization in finely quantized systems by illustrating the effectiveness of sampling the solution space, thereby reducing greatly the number of alternative solutions that must be simulated. We have also shown that in such problems, although it is more difficult to determine the true best policies on the basis of short simulation runs, it is still possible to rapidly determine policies that provide nearly optimal performance. We have demonstrated that our solutions are scalable in the number of transceivers, thus permitting the determination of optimal policies for a problem with many transceivers on the basis of the solution for a much smaller problem; further research is needed to determine whether such observations are applicable to a broader class of problems. The computational saving achieved using these approaches may actually permit the solution of large problems without the need for high-performance computing resources. We have also examined the self-regulation properties of some network examples, and have shown that as the number of transceivers per node increases, more benefit can be achieved from admission control; thus, these results suggest that networks tend to become less self-regulating as the number of transceivers per node increases.

In the end, it is perhaps ironic that the use of the CM-5E, which was motivated by the need to study larger-size network examples, confirmed that excellent solutions for such problems can indeed be obtained without the use of high-performance computing resources. Nevertheless, the use of the CM-5E was crucial to demonstrating that this was indeed the case.

ACKNOWLEDGMENT

This work was supported in part by a grant of HPC time from the DoD HPC Shared Resource Center, Naval Research Laboratory, Washington, D.C., on their Thinking Machines Corp. CM-5E Connection Machine.

REFERENCES

[1] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Standard Clock Simulation and Ordinal Optimization Applied to Admission Control in Integrated Communication Networks," *Journal of Discrete Event Dynamic Systems: Theory and Applications*, **5** pp. 243-279, April/July 1995.

[2] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Ordinal Optimization of Admission Control in Wireless Multihop Integrated Networks via Standard Clock Simulation," NRL Report NRL/FR/5521–95–9781, Naval Research Laboratory, August 1995.

[3] P. Vakili, "Using a Standard Clock Technique for Efficient Simulation," *Operations Research Letters*, **10** pp. 445-452, 1991.

[4] Y.-C. Ho, S. Li, and P. Vakili, "On the Efficient Generation of Discrete Event Sample Paths under Different Parameter Values," *Mathematics and Computation in Simulation*, **30** pp. 347-370, 1988.

[5] C. G. Cassandras, J.-I. Lee, and Y.-C. Ho, "Efficient Parametric Analysis of Performance Measures for Communication Networks," *IEEE Journal on Selected Areas in Communications*, **8**-No. 9 pp. 1709-1722, December 1990.

[6] Y.-C. Ho, R. S. Sreenivas, and P. Vakili, "Ordinal Optimization of DEDS," *Journal of Discrete Event Dynamic Systems*, **2** pp. 61-88, 1992.

[7] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "Admission-Control Policies for Multihop Wireless Networks," *Wireless Networks*, **1**-4 pp. 373-387, December 1995.

[8] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "Admission Control in Integrated Voice/Data Multihop Radio Networks," NRL/MR/5521--93-7196, Naval Research Laboratory, January 18, 1993.

[9] H. C. Tijms, *Stochastic Modelling and Analysis: A Computational Approach*, Chichester: John Wiley & Sons, 1986.

[10] P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation*, New York: Springer Verlag, 1987.

[11] Y.-C. Ho, C. G. Cassandras, and M. Makhlouf, "Parallel Simulation of Real Time Systems via the Standard Clock Approach," *Mathematics and Computers in Simulation*, **35** pp. 33-41, 1993.

[12] C.-H. Chen and Y.-C. Ho, "An Approximation Approach of the Standard Clock Method for General Discrete-Event Simulation," *IEEE Transactions on Control Systems Technology*, **3** pp. 309-317, September 1995.

[13] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Homewood, IL: R. D. Irwin, Inc. and Aksen Associates, Inc., 1993.

[14] R. G. Gallager, *Discrete Stochastic Processes*, Boston: Kluwer Academic Publishers, 1996.

[15] D. J. Baker, A. Ephremides, and J. A. Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control," *IEEE Journal on Selected Areas in Communications*, **SAC-2**-No. 1 pp. 226-237, January 1984.

[16] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling (Invited)," *Proceedings of the IEEE*, **75**-No. 1 pp. 56-73, January 1987.

[17] F. P. Kelly, "Blocking Probabilities in Large Circuit-Switched Networks," *Advances in Applied Probability*, **18** pp. 473-505, 1986.

[18] K. W. Ross and D. Tsang, "Teletraffic Engineering for Product-Form Circuit-Switched Networks," *Advances in Applied Probability*, **22** pp. 657-675, 1990.

[19] S. Jordan and P. Varaiya, "Throughput in Multiple Service, Multiple Resource Communication Networks," *IEEE Transactions on Communications*, **39**-No. 8 pp. 1216-1222, August 1991.

[20] S. Jordan and P. Varaiya, "Control of Multiple Service, Multiple Resource Communication Networks," *IEEE Transactions on Communications*, **42**-11 pp. 2979-2988, November 1994.

[21] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "A Mini-Product-Form-Based Solution to Data-Delay Evaluation in Wireless Integrated Voice/Data Networks," *Proceedings of IEEE INFOCOM'95*, Boston, MA, pp. 1044-1052, April 1995.

[22] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Novel Techniques for the Analysis of Wireless Integrated Voice/Data Networks," NRL Memorandum Report NRL/MR/5521--95-7744, Naval Research Laboratory, July 1995.

[23] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Data-Delay Evaluation in Integrated Wireless Networks based on Local Product-Form Solutions for Voice Occupancy," *Wireless Networks*, **2**-4 pp. 297-314, December 1996.

[24] J. M. Aein, "A Multi-User-Class, Blocked-Calls-Cleared, Demand Access Model," *IEEE Transactions on Communications*, **COM-26**-No. 3 pp. 378-385, March 1978.

[25] G. J. Foschini and B. Gopinath, "Sharing Memory Optimally," *IEEE Transactions on Communications*, **COM-31**-No. 3 pp. 352-360, March 1983.

[26] D. Y. Burman, J. P. Lehoczky, and Y. Lim, "Insensitivity of Blocking Probabilities in a Circuit-Switching Network," *Journal of Applied Probability*, **21** pp. 850-859, 1984.

[27] G. Louth, M. Mitzenmacher, and F. Kelly, "Computational Complexity of Loss Networks," *Journal of Theoretical Computer Science*, **125** pp. 45-59, 1994.

[28] A. E. Conway and N. D. Georganas, *Queueing Networks—Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation*, Cambridge, Mass.: The MIT Press, 1989.

[29] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, London: Springer-Verlag, 1995.

[30] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "An Approach to Voice Admission Control in Multihop Wireless Networks," *Proceedings of IEEE INFOCOM'93*, San Francisco, CA, pp. 246-255, March 1993.

[31] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Efficient Simulation of DEDS by Means of Standard Clock Techniques: Queueing and Integrated Radio Network Examples," NRL/MR/5521--93-7392, Naval Research Laboratory, September 1993.

[32] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Ordinal Optimization of Discrete-Event Dynamic Systems: A Comparison of Standard Clock and Common-Random-Number Methods," *Proceedings of the 1997 Conference on Information Sciences and Systems (CISS)*, Johns Hopkins University, Baltimore, MD, pp. 654-659, March 1997.

[33] J. E. Wieselthier, "Three Techniques for Ordinal Optimization: Short Simulation Runs, Crude Analytical Models, and Imprecise Simulation Models," *Proceedings of the 1997 International Conference on Intelligent Systems and Semiotics: A Learning Perspective (ISAS'97)*, Gaithersburg, MD, pp. 175-180, September 1997.

[34] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "Improvement in Simulation Efficiency by Means of the Standard Clock: A Quantitative Study," *Proceedings of the 32nd IEEE Conference on Decision and Control*, San Antonio TX, pp. 2217-2223, December 1993.

[35] S. G. Strickland and R. G. Phelan, "Massively Parallel SIMD Simulation of Markovian DEDS: Event vs. Time Synchronous Methods," *Journal of Discrete Event Dynamic Systems: Theory and Applications*, **5** pp. 141-166, April/July 1995.

[36] H. A. David, *Order Statistics, Second Edition*, New York: Wiley, 1982.

[37] Y.-C. Ho and M. E. Larson, "Ordinal Optimization Approach to Rare Event Probability Problems," *Journal of Discrete Event Dynamic Systems*, **5** pp. 281-301, April/July 1995.

[38] W.-G. Li, N. T. Patsis, and Y.-C. Ho, "Performance Analysis of Scheduling Algorithms in a Gigabit ATM Switch using Ordinal Optimization," *submitted to IEEE/ACM Transactions on Networking*, 1993.

[39] N. T. Patsis, C.-H. Chen, and M. E. Larson, "SIMD Parallel Discrete-Event Dynamic System Simulation," *IEEE Transactions on Control Systems Technology*, **5** pp. 30-41, January 1997.

[40] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "Ordinal Optimization of Admission Control in Wireless Multihop Voice/Data Networks via Standard Clock Simulation," *Proceedings of IEEE INFOCOM'94*, Toronto, Ontario, Canada, pp. 29-38, June 1994.

[41] S. Kotz and N. L. Johnson, ed., *Encyclopedia of Statistical Sciences,* New York: Wiley, pp. 584-587, 1982.