TWO CLASSES OF PERFORMANCE BOUNDS FOR CLOSED QUEUEING NETWORKS*

Ching-Tarng Hsieh and Simon S. Lam

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

TR-85-09 June 1985

*This research was supported by National Science Foundation Grant No. ECS-8304734.

ABSTRACT

Two classes of performance bounds for separable queueing networks are presented, one for single-chain networks and one for multichain networks. Unlike most bounds for single-chain networks, our bounds are not based upon the consideration of balanced networks. Further, they are proved to be tighter than the balanced job bounds of Zahorjan et al. and tighter than the balanced bounds of Kriz; these comparisons are between bounds with comparable amounts of computational effort. We also present generalized bounds that are calculated using sequences of population sizes; our method extends that of Eager and Sevcik. These generalized bounds are shown to have a nested property. The optimal population sequence, over all sequences of the same length, for getting the tightest bounds is also shown. The other emphasis of this paper is on performance bounds for networks with many closed chains and many service centers. Bounding techniques are especially important for multichain networks since the computation time and space requirements are often so large that an exact solution is not feasible. Models of communication networks typically have many routing chains which are characterized by a sparseness property. In the computation of our performance bounds for multichain networks, we improve their accuracy by making use of routing information and exploiting the sparseness property.

1. Introduction

Separable queueing networks have been widely used as models for predicting the performance of multiprogramming systems as well as packet communication networks. The solution of separable networks requires substantially less computation than does the solution of nonseparable networks. Yet the computation time required by the best algorithms available is nevertheless proportional to the number of customers for single-chain networks and exponential in the number of routing chains for multichain networks. Such computational requirements are very high for many models of realistic networks and systems. (This is especially true for communication networks with many routing chains.) Since the separable queueing networks are themselves approximate models of real systems and networks, an exact solution of their performance measures is not always warranted. This is often true in the early stages of system design.

1.1. Previous work on bounding techniques

Let us first consider networks with a single routing chain. Techniques for deriving upper and lower bounds of the mean delays and throughputs of separable queueing networks have been presented by several authors. The asymptotic bounds of Muntz and Wong [Munt74] are actually applicable to a larger class of queueing networks than the class of separable networks. They also have the advantages of being simple and easy to compute. However, asymptotic bounds are in general very loose and do not provide adequate information to achieve most system design objectives. The work of Zahorjan et al. [Zaho82] was probably the first development of bounds that are restricted to the class of separable queueing networks. Their balanced job bounds (BJBs) were derived by considering related networks whose servers have identical loads and whose performance measures bound those of the original network. Separable networks with fixed-rate service centers but without delay service centers were considered. (Delay service centers are sometimes referred to in the literature as infinite-server service centers.)

Extensions of BJBs for separable networks with both fixed-rate and delay service centers were developed by Eager and Sevcik [Eage83] and by Kriz [Kriz84]. In addition, Eager and Sevcik presented hierarchies of upper and lower bounds. Each hierarchy is a sequence of successively more accurate upper (or lower) bounds with the BJB bound as the first element in the sequence and the exact solution as its limit. Kriz also presented hierarchies of upper and lower bounds, called balanced bounds, with his extensions of BJBs at the first levels of the hierarchies. Methods for obtaining hierarchies of bounds were also developed by Suri [Suri83] and by Stephens and Dowdy [Step84]. Like the method of Eager and Sevcik and the method of Kriz, Suri's method is based upon the MVA recursion equations. On the other hand, the method of Stephens and Dowdy is based upon the convolution algorithm recursion. In each method, a sequence (or hierarchy) of bounds is generated by an iterative procedure which allows one to trade computation time for accuracy. It is interesting to note that BJBs or extensions of BJBs were used as the first-level bounds in all of the methods for generating increasingly more accurate bounds. Only Kriz presented bounds in [Kriz84] that are not based upon the consideration of balanced networks. However, these bounds appear to be better than his balanced bounds only for some networks.

Very little work has been done to develop bounds for multichain networks. BJBs were proposed by Zahorjan et al. [Zaho82] and were extended by Kriz [Kriz84] to multichain networks with both fixed-rate and delay service centers. These bounds are very loose, in particular for networks with many chains and many service centers.

1.2. Overview of our work

We have developed two classes of performance bounds, one for single-chain networks and one for multichain networks. Like BJBs, our first-level bounds are derived from the MVA recursion. But unlike BJBs, our first-level bounds are not based upon the consideration of balanced networks. Instead, our bounds are obtained by assuming that mean queue lengths are proportional to the loads of the corresponding servers. Hence, these bounds are called ***proportional bounds.*** For networks with no delay servers, proportional bounds are proved to be tighter than the balanced job bounds of Zahorjan et al. For networks with both fixed-rate and delay servers, proportional bounds are proved to be tighter than the balanced bounds of Kriz. (As we shall see, these comparisons are between bounds with comparable amounts of computational effort.) We shall also present a method for computing generalized bounds given sequences of population sizes. Our method can be viewed as an extension of the Eager-Sevcik approach for trading computation time for accuracy. We prove that the generalized bounds have a nested property. We also present the optimal population sequence, over all sequences of the same length, for getting the tightest bounds.

Another emphasis of this report is a class of performance bounds for networks with many chains and many service centers. Bounding techniques are especially important for multichain networks for which the computation time and space of an exact solution may be too large to be feasible.

In recent years, several authors, including us, have argued for the use of closed multichain queueing networks to predict the performance of store-and-forward communication networks and to solve network design problems such as the optimal selection of routes and channel capacities [Lam82]. A recent experimental study of ours [Lam85] further illustrated the inadequacy of the open queueing network model and the desirability of the closed network model. The obstacle that currently prevents the closed network model from being widely used by network designers and analysts is the large computational time and space required to calculate performance measures. Models of realistic communication networks should have tens of closed chains or more, each modeling a flow-controlled virtual channel. Such models cannot be solved by the conventional convolution and MVA algorithms [Buze73,Chan75,Reis75,Reis80]. Lam and Lien observed in [Lam83] that models of communication networks have routes that are often characterized by sparseness and locality properties. They developed the tree convolution algorithm that exploits routing information and can solve networks with tens of closed chains. Tree MVA algorithms were subsequently developed independently by Tucci and Sauer [Tucc82] and by Hoyme et al. [Hoym82].

Tree algorithms are too expensive to be used in network design algorithms which need to evaluate very efficiently a network's performance given certain design perturbations or parameter changes. Reasonably tight performance bounds are very useful for speeding up heuristic search procedures based upon the branch-and-bound technique. Another place where we have found a useful application of performance bounds of closed multichain networks is in the implementation of dynamic scaling in convolution algorithms [Lam82] to prevent the occurrences of floating point underflows and overflows. (We employ the tree convolution algorithm and its associated tree of arrays whenever an exact solution is called for in our network design techniques [Lam85].) In this role, the bounds can be very loose but must be efficient to compute.

We have developed two algorithms for computing performance bounds for closed multichain networks. Like the tree convolution algorithm, routing information is exploited in the computation of these performance bounds. The first algorithm is based upon the BJB idea. The second algorithm further exploits routing information to improve the bounds obtained by the first algorithm. The accuracy of these bounds is much better than BJBs for networks with many sparse routing chains.

In Section 2, we present proportional bounds for closed single-chain networks. In Section 3, we present hierarchies of nested proportional bounds. In Section 4, our performance bounds for closed multichain networks are presented. Throughout this paper the networks considered are BCMP networks with fixed-rate (F) servers and delay (D) service centers [Bask75]. M denotes the total number of service centers.

2. Proportional Bounds

To derive proportional bounds for closed single-chain networks, we consider formulas in the MVA recursion. The mean delay $D_m(n)$ of center m in a queueing network with n customers is

$$D_m(n)=\begin{cases}\tau_m(1+q_m(n-1)) & \text{if } m \text{ is a fixed-rate server}\\ \tau_m & \text{if } m \text{ is a delay server}\end{cases} \tag{1}$$

where τ_m is the mean service time at center m and $q_m(n-1)$ is the mean queue length at service center m in a network with $n-1$ customers [Sevc79,Reis80].

From Little's formula [Litt61] , the throughput $T(n)$ and mean queue length $q_m(n)$ are

$$T(n)=\frac{n}{\sum_{m=1}^{M} D_m(n)} \tag{2}$$

and

$$q_m(n)=T(n)D_m(n). \tag{3}$$

Equations (1), (2) and (3) form the main recursion of the MVA method [Reis80]. Starting from these equations, we present several lemmas which lead to the proportional bounds. Proofs of lemmas, theorems and corollaries stated below can be found in the appendix.

Without loss of generality, assume that the fixed-rate service centers are labeled $1,2,\ldots,M_F$, and the remaining M_D centers are delay centers, where $M_D \geq 0$, $M_F \geq 0$ and $M_D+M_F=M$ We further assume that $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{M_F}$. Define $L_F=\sum_{m=1}^{M_F}\tau_m$, $L_D=\sum_{m=M_F+1}^{M}\tau_m$ and $L=L_F+L_D$.

Lemma 1: The mean queue lengths of any two fixed-rate service centers satisfy the following inequality

$$\frac{q_i(n)}{q_j(n)} \leq \frac{\tau_i}{\tau_j} \quad \text{for } i \leq j. \tag{4}$$

Lemma 2: If $q_j(n) \leq \frac{\tau_j}{L_F}\sum_{m=1}^{M_F} q_m(n)$, $\quad j \leq M_F$ then

$$q_i(n) \leq \frac{\tau_i}{L_F} \sum_{m=1}^{M_F} q_m(n) \qquad \text{for all } i \text{ such that } 1 \leq i \leq j. \tag{5}$$

Lemma 3: If $q_i(n) \geq \frac{\tau_i}{L_F} \sum_{m=1}^{M_F} q_m(n), \quad i \leq M_F$ then

$$q_j(n) \geq \frac{\tau_j}{L_F} \sum_{m=1}^{M_F} q_m(n) \qquad \text{for all } j \text{ such that } i \leq j \leq M_F. \tag{6}$$

Lemma 4: The mean queue lengths of the first and the last fixed-rate service centers satisfy the following inequalities

$$q_1(n) \leq \frac{\tau_1}{L_F} \sum_{m=1}^{M_F} q_m(n) \tag{7}$$

and

$$q_{M_F}(n) \geq \frac{\tau_{M_F}}{L_F} \sum_{m=1}^{M_F} q_m(n). \tag{8}$$

Theorem 1: The network delay $D(n)$ and network throughput $T(n)$ satisfy the following inequalities

$$D(n) \geq L + \sum_{m=1}^{M_F} \frac{\tau_m^2}{L_F} \left[n-1-L_D \times T(n-1) \right] \tag{9}$$

and

$$T(n) \leq \frac{n}{L + \sum_{m=1}^{M_F} \frac{\tau_m^2}{L_F} \left[n-1-L_D \times T(n-1) \right]} \tag{10}$$

Corollary 1: For a network with no delay servers, the network throughput $T(n)$ is bounded above by

$$\frac{n}{L + \sum_{m=1}^{M_F} \frac{\tau_m^2}{L_F} \left[n-1 \right]}$$

which is smaller than (or equal to) the following balanced job bound in [Zaho82]

$$\frac{n}{L+\frac{L_F}{M_F}\left[n-1\right]}.$$

For networks with one or more delay servers, the RHSs of Eqs. (9) and (10) in Theorem 1 are functions of $T(n-1)$. Sequences of bounds for $D(n)$ and $T(n)$ can be obtained as follows.

Define $\overline{T}(n,0)=min(n/L, 1/\tau_{M_F})$ and $\underline{D}(n,0)=\max(L, n\tau_{M_F})$ for all n. These are the asymptotic bounds. Next, define

$$\underline{D}(n,i)=\max\left\{n\tau_{M_F}, L+\sum_{m=1}^{M_F}\frac{\tau_m^2}{L_F}\left[n-1-L_D\times\overline{T}(n-1,i-1)\right]\right\} \tag{11}$$

and

$$\overline{T}(n,i)=n/\underline{D}(n,i) \tag{12}$$

for $1\leq i\leq n$.

Theorem 2:

$$D(n)\geq\underline{D}(n,i+1)\geq\underline{D}(n,i) \tag{13}$$

and

$$T(n)\leq\overline{T}(n,i+1)\leq\overline{T}(n,i) \tag{14}$$

for $0\leq i\leq n-1$.

Corollary 2: For each i, the proportional throughput upper bound in Eq. (12) is smaller than (or equal to) the corresponding balanced bound in [Kriz84].

The counterparts of Lemmas 1-4 for a proportional throughput lower bound are given below and the bound itself is presented in Theorem 3.

Lemma 5: The mean queue lengths of any two fixed-rate service centers i and j with $i\leq j$ satisfy the following inequality

$$\frac{q_i(n)}{q_j(n)} \geq (\frac{\tau_i}{\tau_j})^n \qquad \text{for all} \quad n \geq 1. \tag{15}$$

We next define $L_F^n = \sum_{m=1}^{M_F} \tau_m^n$.

Lemma 6: If $q_j(n) \geq \frac{\tau_j^n}{L_F^n} \sum_{m=1}^{M_F} q_m(n)$, $\quad 1 \leq j \leq M_F$, then

$$q_i(n) \geq \frac{\tau_i^n}{L_F^n} \sum_{m=1}^{M_F} q_m(n) \quad \text{for all } i \text{ such that } 1 \leq i \leq j. \tag{16}$$

Lemma 7: If $q_i(n) \leq \frac{\tau_i^n}{L_F^n} \sum_{m=1}^{M_F} q_m(n)$, $\; i \leq M_F$ then

$$q_j(n) \leq \frac{\tau_j^n}{L_F^n} \sum_{m=1}^{M_F} q_m(n) \quad \text{for all } j \text{ such that } i \leq j \leq M_F. \tag{17}$$

Lemma 8: The mean queue lengths of the first and the last fixed-rate service centers satisfy the following inequalities

$$q_1(n) \geq \frac{\tau_1^n}{L_F^n} \sum_{m=1}^{M_F} q_m(n) \tag{18}$$

and

$$q_{M_F}(n) \leq \frac{{\tau_{M_F}}^n}{L_F^n} \sum_{m=1}^{M_F} q_m(n). \tag{19}$$

Theorem 3: The network delay $D(n)$ and network throughput $T(n)$ satisfy the following inequalities

$$D(n) \leq L + \frac{L_F^n}{L_F^{n-1}} \left[n-1-L_D \times T(n-1) \right] \tag{20}$$

and

$$T(n) \geq \frac{n}{L + \frac{L_F^n}{L_F^{n-1}} \left[n-1-L_D \times T(n-1) \right]} \tag{21}$$

Corollary 3: For a network with no delay servers, the network throughput $T(n)$ is bounded below by

$$\frac{n}{L + \frac{L_F^n}{L_F^{n-1}} \left[n-1 \right]}$$

which is larger than (or equal to) the following balanced job bound in [Zaho82]

$$\frac{n}{L + \tau_{M_F}[n-1]}.$$

For networks with delay servers, the RHSs of Eqs. (20) and (21) in Theorem 3 are functions of $T(n-1)$. Sequences of bounds for $D(n)$ and $T(n)$ can be obtained as follows.

Define $\underline{T}(n,0)=0$ for all n. Let

$$\overline{D}(n,i)=L + \frac{L_F^n}{L_F^{n-1}} \left[n-1-L_D \times \underline{T}(n-1,i-1) \right] \tag{22}$$

and

$$\underline{T}(n,i)=\frac{n}{\overline{D}(n,i)} \tag{23}$$

for $1 \leq i \leq n$.

Theorem 4:

$$D(n) \leq \overline{D}(n,i+1) \leq \overline{D}(n,i) \qquad 1 \leq i \leq n-1$$

and

$$T(n) \geq \underline{T}(n,i+1) \geq \underline{T}(n,i) \qquad 0 \leq i \leq n-1.$$

Corollary 4: For each i, the proportional throughput lower bound in Eq. (23) is larger than (or equal to) the corresponding balanced bound in [Kriz84].

Observation: When $n \rightarrow \infty$ the LHS of Eq. (21) converges to the RHS. The proportional bounds given in Theorem 4 (as well as the corresponding balanced bounds) are asymptotically exact.

The following examples are taken from [Kriz84] to illustrate the accuracy of proportional bounds and of balanced bounds.

Example 1: The network has only fixed-rate service centers and is almost balanced. τ_1=0.08, τ_2=0.09 and τ_3=τ_4=0.1. The balanced job bounds, and the proportional bounds are given in Table 1 below.

Population size	Throughput bounds				
	$\underline{X}$	$\underline{T}$	exact	$\overline{T}$	$\overline{X}$
2	4.255	4.317	4.317	4.317	4.324
5	6.494	6.660	6.715	6.729	6.757
10	7.874	8.022	8.206	8.27	8.316
20	8.811	8.867	9.168	9.338	9.401
30	9.174	9.194	9.499	9.759	9.828
40	9.368	9.375	9.654	9.984	10
60	9.569	9.570	9.792	10	10
80	9.674	9.674	9.853	10	10

where $\underline{X}$ is the balanced job lower bound
$\overline{X}$ is the balanced job upper bound
$\underline{T}$ is the proportional lower bound
and $\overline{T}$ is the proportional upper bound.

Table 1. Throughput bounds for an almost-balanced network with no delay server.

Notice that although the network is almost balanced, the proportional bounds are better than the balanced bounds. Also notice that the proportional bounds give the exact throughput when the population size is 2.

Example 2: The network of Example 1 is extended by a delay server with mean service time τ=1. The first- and second-level balanced bounds of Kriz and proportional bounds are shown in Table 2.

Population size	Throughput bounds								
	$\underline{Y}_1$	$\underline{Y}_2$	$\underline{T}_1$	$\underline{T}_2$	exact	$\overline{T}_2$	$\overline{T}_1$	$\overline{Y}_2$	$\overline{Y}_1$
2	1.361	1.432	1.367	1.434	1.434	1.434	1.434	1.434	1.434
5	2.825	3.267	2.856	3.284	3.364	3.367	3.400	3.369	3.402
10	4.405	5.390	4.451	5.427	5.872	5.970	6.263	5.980	6.270
20	6.116	7.489	6.143	7.517	8.375	8.679	9.053	8.716	9.081
30	7.027	8.393	7.037	8.405	9.186	9.413	9.549	9.468	9.592
40	7.590	8.858	7.595	8.863	9.502	9.773	9.818	9.836	9.870
60	8.253	9.306	8.254	9.307	9.740	10	10	10	10
80	8.630	9.514	8.630	9.514	9.828	10	10	10	10

where T denotes proportional bounds and Y denotes Kriz's bounds.

Table 2. Throughput bounds for an almost-balanced network with one delay server.

Example 3: The network is unbalanced with no delay server. The mean service times at the four service centers are τ_1=0.04, τ_2=0.05, and τ_3=τ_4=0.1.

Population size	Throughput bounds				
	$\underline{X}$	$\underline{T}$	exact	$\overline{T}$	$\overline{X}$
2	5.128	5.360	5.360	5.360	5.517
5	7.246	7.341	7.803	8.033	8.621
10	8.403	8.407	8.930	9.635	10
15	8.876	8.876	9.302	10	10
20	9.132	9.132	9.483	10	10
30	9.404	9.404	9.659	10	10
40	9.547	9.547	9.746	10	10
60	9.693	9.693	9.831	10	10
80	9.768	9.768	9.874	10	10

Table 3. Throughput bounds for an unbalanced network with no delay server.

Example 4: The network of Example 3 is extended by a delay server with mean service time τ=1.

Population size	Throughput bounds								
	$\underline{Y}_1$	$\underline{Y}_2$	$\underline{T}_1$	$\underline{T}_2$	exact	$\overline{T}_2$	$\overline{T}_1$	$\overline{Y}_2$	$\overline{Y}_1$
2	1.439	1.524	1.457	1.528	1.528	1.528	1.528	1.531	1.531
5	2.959	3.476	2.974	3.484	3.628	3.635	3.664	3.667	3.690
10	4.556	5.684	4.567	5.685	6.422	6.586	6.858	6.743	6.960
15	5.576	6.978	5.576	6.979	8.133	8.836	9.245	9.139	9.494
20	6.270	7.767	6.270	7.767	8.945	9.703	9.814	10	10
30	7.160	8.618	7.160	8.618	9.483	10	10	10	10
40	7.707	9.042	7.707	9.042	9.659	10	10	10	10
60	8.345	9.437	8.345	9.437	9.797	10	10	10	10
80	8.705	9.614	8.705	9.614	9.856	10	10	10	10

Table 4. Throughput bounds for an unbalanced network with one delay server.

3. Generalized Proportional Bounds

We next present algorithms which permit us to trade computation time for improved accuracy. Consider a single-chain network with M fixed-rate service centers, population size N, and a population sequence of S integers, $n_1, n_2, \ldots, n_S$, where $1 < n_1 < n_2 < \cdots < n_S = N$. Algorithm 1 computes a generalized throughput upper bound for each population in the sequence.

```
Algorithm 1 generalized_upper_bound;
begin
  max_throughput := 1 / load[M];
  total_load := 0;
  for m := 1 to M do total_load := total_load + load[m];
  for m := 1 to M do ratio[m] := load[m] / total_load;
  for i := 1 to S do
    begin
      total_delay := 0;
      for m := 1 to M do
        begin
          delay[m] := load[m] * ( 1 + ratio[m] * (n[i] - 1));
          total_delay := total_delay + delay[m];
        end;
        throughput_upper := n[i] / total_delay;
        if throughput_upper > max_throughput
          then throughput_upper := max_throughput;
```

```
        for m := 1 to M do ratio[m] := delay[m] / total_delay;
    end;
end;
```

Before presenting several theorems stating some properties of the algorithm, we give three lemmas (Lemmas 9-11) that form the basis of our algorithm. The lemmas, their proofs as well as proofs of the theorems can be found in the appendix. We shall use $r_j(n_i)$ to denote the value of ratio[j] when the population size is n_i during the execution of Algorithm 1.

For simplicity, these lemmas and theorems are stated and proved for networks with fixed-rate service centers only. Extension of our results to networks including delay service centers is straightforward. As in Section 2, the mean service times for the service centers have the following relation: $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_M$.

Theorem 5: For all i, $1 \leq i \leq S$

$$\sum_{m=1}^{M} \tau_m \left[1+q_m(n_i-1)\right] \geq \sum_{m=1}^{M} \tau_m \left[1+r_m(n_{i-1}) \times (n_i-1)\right] \qquad (24)$$

where $n_0=1$ and $r_m(1)=\tau_m/L$ for $m=1,2,...,M$.

Eq. (24) in Theorem 5 assures that Algorithm 1 computes lower bounds on delay and upper bounds on throughput.

Corollary 5: If the population sequence is $2,...,N$ then the algorithm computes the exact network throughput.

In the next two theorems we present properties of the generalized proportional throughput upper bounds. Given a population sequence $n_1, n_2, \ldots, n_S$ of S elements, a subsequence is said to be valid if it includes the population size $n_S (=N)$.

Applying Algorithm 1 to different population sequences yields different throughput upper bounds. These throughput upper bounds are said to be *nested* if the throughput upper bound computed from a population sequence is smaller than or equal to the throughput upper bound computed from any of the valid subsequences.

Theorem 6: The generalized proportional throughput upper bounds are nested.

Theorem 7: *Optimal population sequence.* Given an integer $S \leq N-1$, the population sequence of length S that yields the smallest throughout upper bound is the sequence $N-S+1, N-S+2, \ldots, N$.

We calculated generalized proportional throughput upper bounds for the network considered earlier in Example 1. The results are plotted in Figures 1 and 2. Figure 1 illustrates the nested property. Figure 2 illustrates the optimal population sequence of length 7.

A slightly modified algorithm which can handle delay servers is presented in Algorithm 2 below. Service centers 1 to MF are fixed-rate servers and service centers MF+1 to M are delay servers. For convenience, n[0] is set to 1.

```
Algorithm 2 generalized_upper_bound_delay_server;
begin
  max_throughput := 1 / load[M];
  throughput_upper := max_throughput;
  load_fixed := 0;
  for m := 1 to MF do load_fixed := load_fixed + load[m];
  for m := 1 to MF do ratio[m] := load[m] / load_fixed;
  for i := 1 to S do
    begin
      total_delay := 0;
      queue_F := n[i] - 1;
      for m := MF + 1 to M do
        begin
          total_delay := total_delay + load[m];
          queue_F := queue_F - load[m] * throughput_upper * (n[i]-1) / n[i-1];
        end;
      for m := 1 to M do
        begin
          delay[m] := load[m] * ( 1 + ratio[m] * queue_F );
          total_delay := total_delay + delay[m];
        end;
        throughput_upper := n[i] / total_delay;
        if throughput_upper > max_throughput
          then throughput_upper := max_throughput;
      for m := 1 to M do ratio[m] := delay[m] / total_delay;
    end;
end;
```

Next, we consider lower bounds. Consider a single-chain network with M fixed-rate service centers, population size N, and a population sequence of S integers, $n_1, n_2, \ldots, n_S$, where $n_1 \geq n_2 \geq \cdots \geq n_S = N$ and $N \geq 3$. The following algorithm computes a generalized throughput lower bound for each

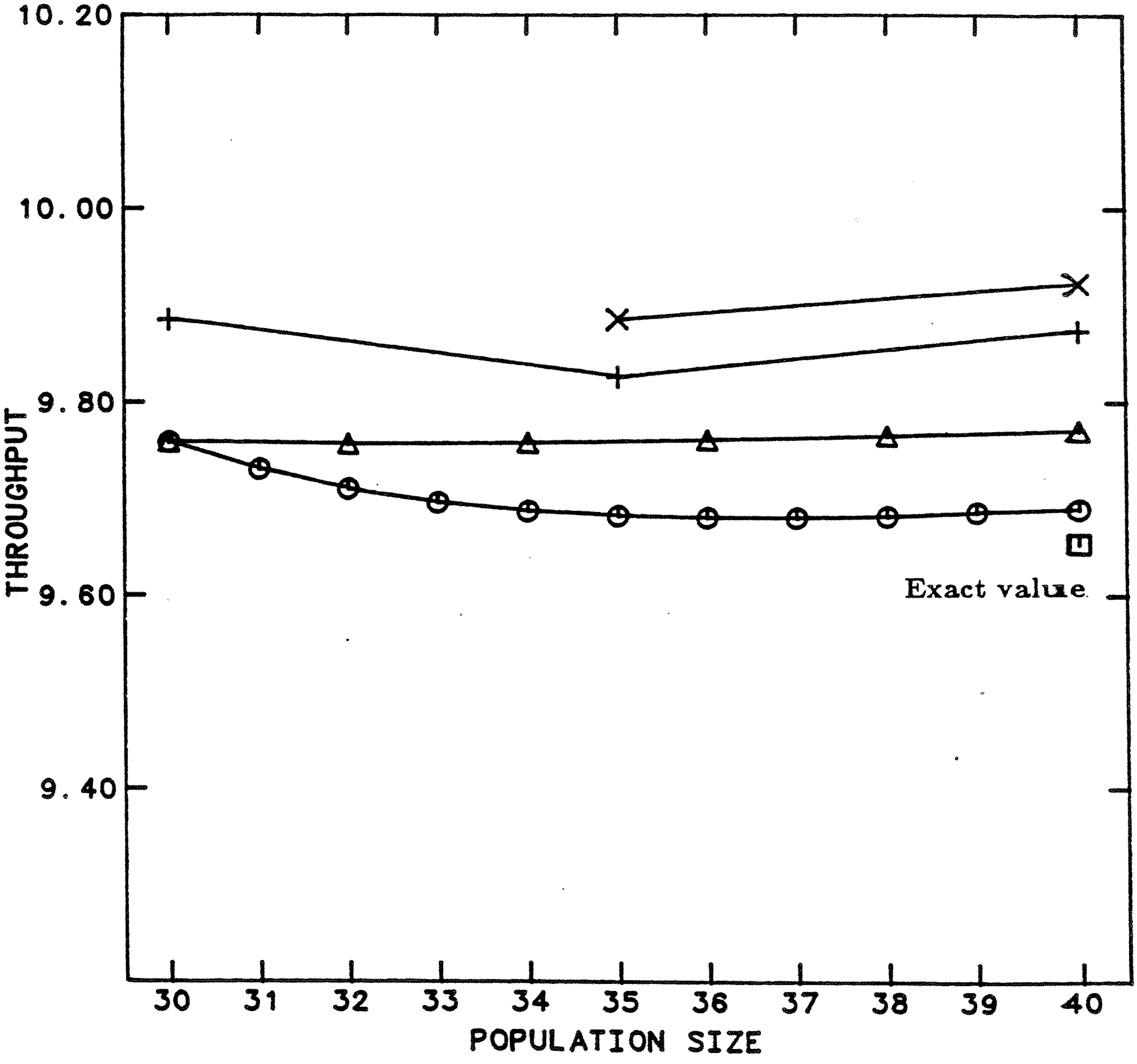

Figure 1. Nested property of generalized proportional upper bounds.

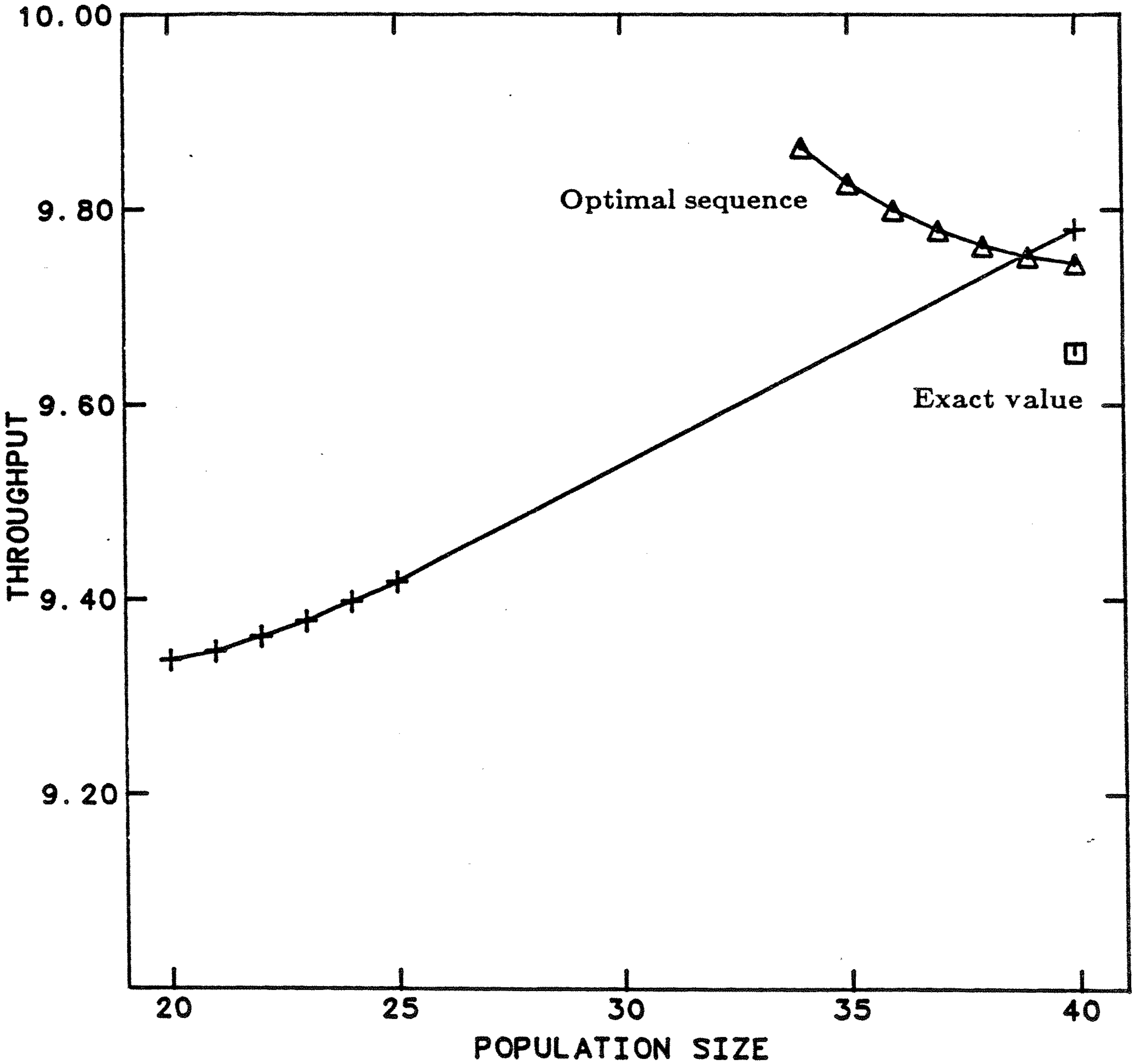

Figure 2. Optimal population sequence of length 7 for generalized proportional upper bounds.

population in the sequence.

```
Algorithm 3 generalized_lower_bound;
begin
  sum := 0;
  for m := 1 to M do
    begin
      ratio[m] := load[m] ** (n[1] - 1);
      sum := sum + ratio[m];
    end;
  for m := 1 to M do ratio[m] := ratio[m] / sum;
  for i := 1 to S do
    begin
      total_delay := 0;
      for m := 1 to M do
        begin
          delay[m] := load[m] * ( 1 + ratio[m] * (n[i] - 1));
          total_delay := total_delay + delay[m];
        end;
        throughput_lower := n[i] / total_delay;
      for m := 1 to M do ratio[m] := delay[m] / total_delay;
    end;
end;
```

In what follows, $r_m(n_i)$ denotes the value of ratio[m] when the population size is n_i during the execution of Algorithm 3.

Theorem 8: For all i, $1 \le i \le S$,

$$\sum_{m=1}^{M} \tau_m [1+q_m(n_i-1)] \le \sum_{m=1}^{M} \tau_m [1+r_m(n_{i-1}) \times (n_i-1)] \tag{25}$$

where $n_0 = n_1 - 1$ and $r_m(n_0) = \tau_m^{n_0} / L_F^{n_0}$.

Theorem 8 assures that Algorithm 3 computes upper bounds on delay and lower bounds on throughput. These bounds also have properties similar to those shown earlier for generalized throughput lower bounds.

Consider a population sequence $n_1, n_2, \ldots, n_S = N$. For the purpose of computing generalized throughput lower bounds, a subsequence is said to be valid if it contains both n_1 and n_S. Generalized throughput lower bounds are said to be *nested* if the throughput lower bound computed from a population sequence is larger than or equal to the throughput lower bound computed from any of the valid population subsequences.

Theorem 9: The generalized proportional throughput lower bounds are nested.

Theorem 10: *Optimal population sequence.* Given an integer S, the population sequence that yields the largest throughout lower bound among all population sequences of length S is $N, N, \ldots, N$.

We calculated generalized proportional throughput lower bounds for the network considered earlier in Example 1. Figure 3 illustrates the nested property. Figure 4 illustrates the optimal population sequence of length 7.

A slightly modified algorithm which can handle delay servers is presented in Algorithm 4 below. For convenience, n[0] is set to 1.

```
Algorithm 4 generalized_lower_bound_delay_server;
begin
  sum := 0;
  throughput_lower := 0;
  for m := 1 to MF do
    begin
      ratio[m] := load[m] ** (n[1] - 1);
      sum := sum + ratio[m];
    end;
  for m := 1 to MF do ratio[m] := ratio[m] / sum;
  for i := 1 to S do
    begin
      total_delay := 0;
      for m := MF + 1 to M do
        begin
          total_delay := total_delay + load[m];
          queue_F := queue_F - load[m] * throughput_lower * (n[i]-1) / n[i-1];
        end;
      for m := 1 to MF do
        begin
          delay[m] := load[m] * ( 1 + ratio[m] * queue_F);
          total_delay := total_delay + delay[m];
        end;
        throughput_lower := n[i] / total_delay;
      for m := 1 to M do ratio[m] := delay[m] / total_delay;
    end;
end;
```

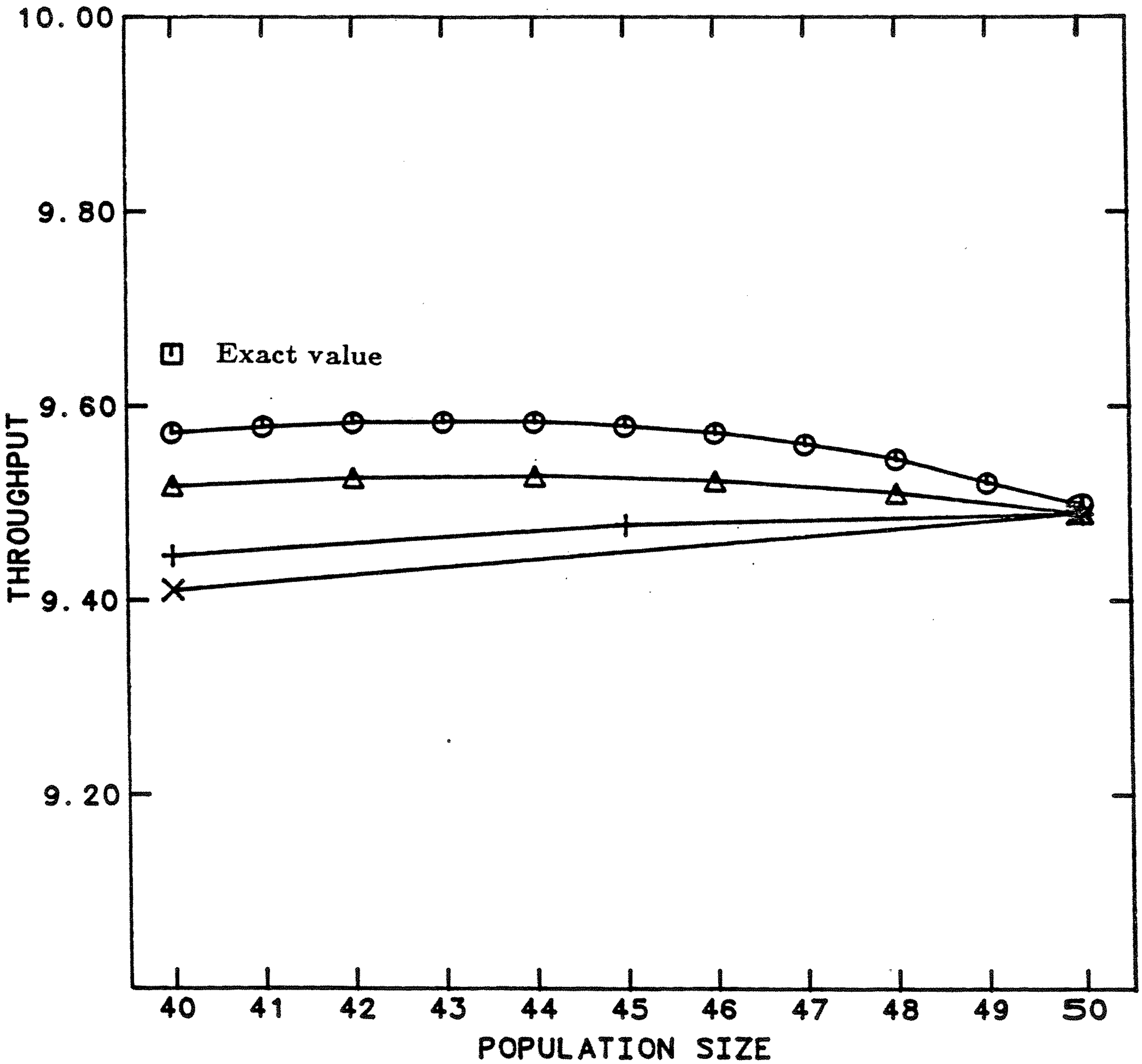

Figure 3. Nested property of generalized proportional lower bounds.

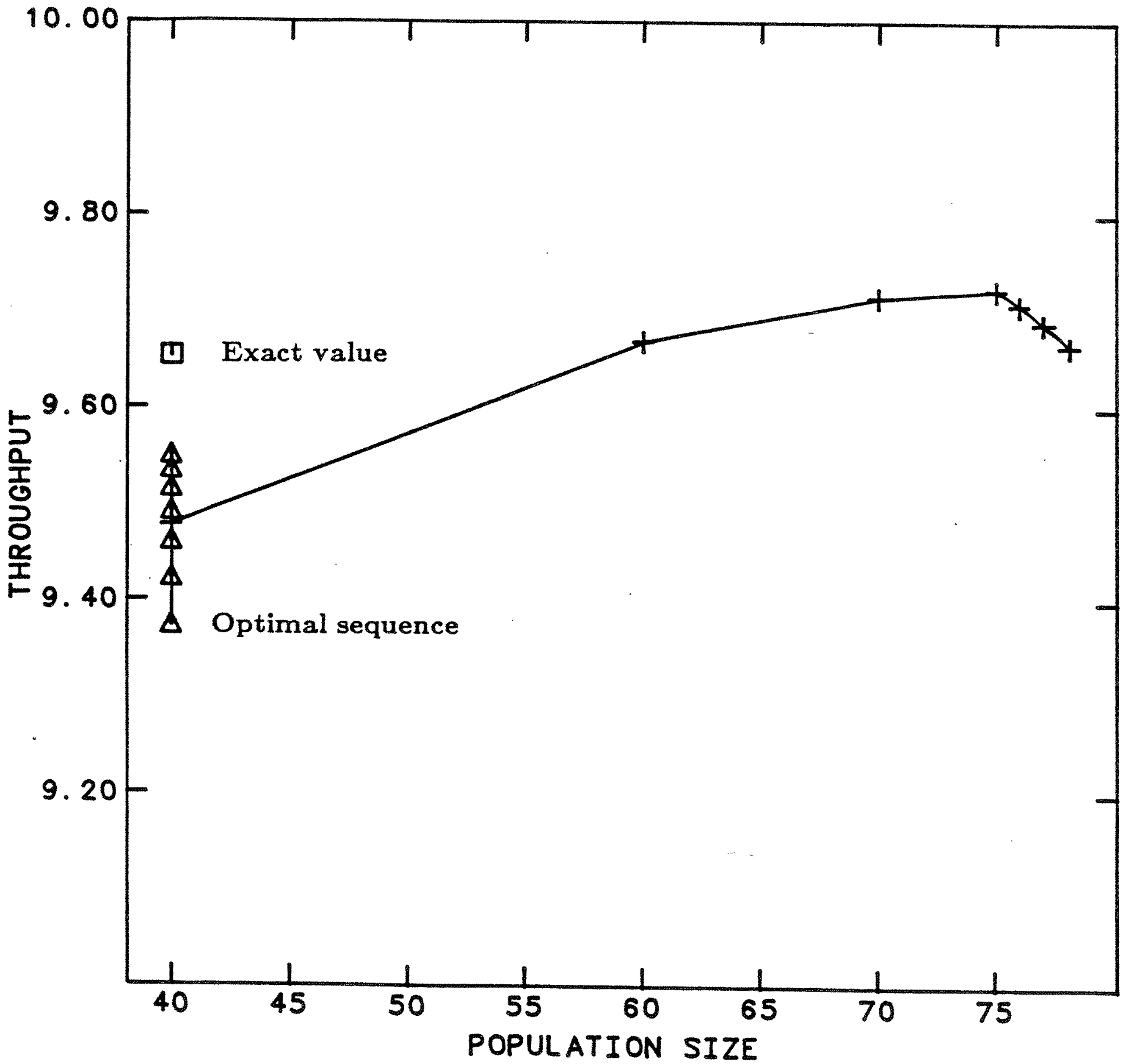

Figure 4. Optimal population sequence of length 7 for generalized proportional lower bounds.

Our generalized proportional throughput bounds are especially useful if the throughput bounds for many different population sizes are required.

4. Bounds for Closed Multichain Networks

In this section we present throughput bounds for multichain queueing networks with fixed-rate and delay service centers. The meaning of each variable is the same as its corresponding one in single-chain networks except that an additional subscript, h or k, is used to denote a specific chain. Proofs of Theorem 11 and Corollary 6 are given in the appendix.

Theorem 11: The mean delay $D_k(\underline{n})$ of chain k customers satisfies the following inequalities

$$\underline{D}_k(\underline{n}) \leq D_k(\underline{n}) \leq \overline{D}_k(\underline{n})$$

where

$$\underline{D}_k(\underline{n}) = L_k + \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{mk} \sum_{\substack{h=1 \\ \text{chain } h \text{ visits } m}}^{K} \tau_{mh} \underline{T}_h(\underline{n}-\underline{1}_k)$$

$$+\tau_{min,k}\left[n_k - 1 - L_{D,k}\overline{T}_k(\underline{n}-\underline{1}_k) - L_{F,k}\underline{T}_k(\underline{n}-\underline{1}_k)\right] \qquad (26)$$

and

$$\overline{D}_k(\underline{n}) = L_k + \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{mk} \sum_{\substack{h=1 \\ \text{chain } h \text{ visits } m}}^{K} \tau_{mh} \underline{T}_h(\underline{n}-\underline{1}_k)$$

$$+ \sum_{\substack{h=1 \\ \text{chain } h \\ \text{intersects chain } k}}^{K} \tau_{max,h,k}\left[n_h - L_h \underline{T}_h(\underline{n}-\underline{1}_k)\right] - \tau_{max,k,k} \qquad (27)$$

where

$\tau_{max,h,k}$ is the maximum mean service time among the fixed-rate service centers traversed by both chain h and chain k customers,

$\tau_{min,k}$ is the minimum mean service time among the fixed-rate service centers traversed by chain k customers,

$\underline{T}_h(\underline{n}-\underline{1}_k)$ is a lower bound of $T_h(\underline{n}-\underline{1}_k)$,

and

$\overline{T}_k(\underline{n}-\underline{1}_k)$ is an upper bound of $T_k(\underline{n}-\underline{1}_k)$.

Chain h is said to intersect chain k if it visits a fixed-rate service center that is also visited by chain k.

Corollary 6: The throughput $T_k(\underline{n})$ of chain k satisfies the following inequalities

$$\underline{T}_k(\underline{n}) \leq T_k(\underline{n}) \leq \overline{T}_k(\underline{n}) \tag{28}$$

where

$$\underline{T}_k(\underline{n}) = \frac{n_k}{\overline{D}_k(\underline{n})} \tag{29}$$

and

$$\overline{T}_k(\underline{n}) = \frac{n_k}{\underline{D}_k(\underline{n})} \tag{30}$$

4.1. Algorithms

The procedure to compute throughput bounds involves the following steps:

i. Find fast lower bounds of $T_h(\underline{n}-\underline{1}_k)$ for all $h,k=1,2,\ldots,K$ and fast upper bounds of $T_k(\underline{n}-\underline{1}_k)$ for all $k=1,2,\ldots,K$.

ii. Plug the fast upper bounds of $T_k(\underline{n}-\underline{1}_k)$ into Eq. (26) and the fast lower bounds of $T_k(\underline{n}-\underline{1}_k)$ and $T_h(\underline{n}-\underline{1}_k)$ into Eqs. (26) and (27) to calculate bounds of mean delay. Apply Eqs. (29) and (30) to obtain throughput bounds.

Some of the variables used in the algorithms are defined in the following. The meaning of other variables is self-explanatory.

load_total[k] = sum of mean service times of chain k at fixed-rate and delay service centers

load_max[h,k] = the largest mean service time among all fixed-rate service centers visited by both chain h and chain k

visit_common_queue[h,i] = True if chain h and chain i visit at least one

common fixed-rate service center
load[k,m] = mean service time of chain k at service center m
load_D[k] = sum of mean service times at delay centers for chain k
load_F[k] = sum of mean service times at fixed-rate centers for chain k
load_min[k] = the smallest mean service time among fixed-rate service centers visited by chain k
visit[h,m] = True if service center m is visited by chain h

Algorithm 5, given below, finds a fast lower bound of $T_h(\underline{n}-\underline{1}_k)$. It utilizes some routing information.

```
Algorithm 5 fast_throughput_lower_bounds;
begin
  for k := 1 to num_chains do
  begin
    population[k] := population[k] - 1;
    for h := 1 to num_chains do
      if population[h] > 0
      then
       begin
         delay := (population[h] - 1) * load_max[h,h];
         for i := 1 to num_chains do
           if visit_common_queue[h,i] and (h ≠ i)
          then delay := delay + population[i] * load_max[h,i];
           delay := delay + load_total[h];
           throughput_lower[h,k] := population[h]/delay;
       end
      else throughput_lower[h,k] := 0;
  end;
end;
```

The above procedure is actually a special case of Algorithm 6 below. Its throughput bound of $T_h(\underline{n}-\underline{1}_k)$ is obtained by replacing all throughput lower bounds in Eq. (27) with zero.

Algorithm 5 calculates throughput lower bounds only. There are two methods to obtain fast throughput upper bounds of $T_k(\underline{n}-\underline{1}_k)$. First, we can use BJB upper bounds for a multichain network [Zaho82]. Second, we can consider a network in which all chains, except chain k, are removed and use the proportional upper bound for such a single-chain network.

The second procedure, to be given next, uses the fast bounds described above to calculate improved bounds for mean delays. It then applies Little's formula to obtain bounds for $T_k(\underline{n})$, for all $k=1,2,...,K$. The computation sequence follows Eqs. (26) and (27) exactly.

```
Algorithm 6 multichain_throughput_bounds;
begin
  for k := 1 to num_chains do
  begin
    population[k] := population[k] - 1;      /* remove one chain k customer */
    min_delay := load_total[k];
    delay_others := 0;
    for m := 1 to num_queues do
    if visit[k,m]
   then
      begin
        queue := 0;
        for h := 1 to num_chains do if visit[h,m]
       then queue := queue + load[h,m] * throughput_lower[h,k];
        min_delay := min_delay + load[k,m] * queue;
      end;
    for h := 1 to num_chains do
    if visit_common_queue[h,k]
   then
      begin
        queue_others := population[h] - load_total[h] * throughput_lower[h,k];
        delay_others := delay_others + load_max[h,k] * queue_others;
      end;
    population[k] := population[k] + 1;
    queue_others := population[k] - load_D[k] * throughput_upper[k]
                    - load_F[k] * throughput_lower[k,k];
    delay_lower := min_delay + load_min[k] * queue_others;
    delay_upper := min_delay + delay_others;
    final_throughput_upper[k] := population[k]/delay_lower;
    bottleneck := max(1/load_max[k,k], population[k]/load_total[k]);
    if final_throughput_upper[k] > bottleneck
   then final_throughput_upper[k] := bottleneck;
    final_throughput_lower[k] := population[k]/delay_upper;
  end;
end;
```

4.2. Numerical examples

The first network used is a 26-node network with 32 full-duplex communication links and 32 virtual channels. The window size (chain population size) is 2 for each virtual channel. Because we assume full-duplex virtual channels with symmetric traffic, the network reduces to a queueing network model with 32 fixed-rate service centers and 16 closed chains. Additionally, we employ 16 fixed-rate servers, one for each closed chain, to model the external sources of virtual channels. The mean service time for each service center is 0.1 sec and is the same for all virtual channels. The mean service time at the source servers is 1.0 sec. Table 5 below shows the routes for the 16 virtual channels. The calculated throughput bounds are shown in Figure 5. The maximum, minimum, and average utilizations of the 31 fixed-rate service centers actually used in the network are 0.272, 0.0895, and 0.193 respectively. (One service center has zero utilization and was excluded.) In this case, the network is lightly loaded.

VC	route (in node sequence)
1	16 17 18 19 4 5
2	1 2 3 17 18 19
3	6 25 22 23 24 26
4	24 10 11 12 13
5	13 1 2 3 4
6	17 3 4 5 6
7	1 2 3 4 5 6 7 8 9
8	1 13 12
9	9 10 24
10	21 20 25
11	15 16 17
12	23 24
13	21 22 25 6 7 8
14	23 22 21 20
15	21 15 14 13 12 11 10 9
16	1 13 14 15 21 22 25

Table 5. Routes of virtual channels.

In Figure 6, for the same network, the mean service time at each source server is set to 0.1 sec. The maximum, minimum, and average utilizations of the 31 communication channels are 0.753, 0.212, and 0.519 respectively. This represents a fairly heavily loaded network.

The second network used in our numerical study is a randomly generated network with 12 nodes, 30 virtual channels and 34 communication channels. The communication channels and their mean service times are shown in Table 6. The notation (i,j) in Table 6 denotes a communication channel from node i to node j. The route, window size, and mean service time of the source server for each virtual channel are given in Table 7. The maximum, minimum, and average utilizations of the 32 communication channels with nonzero utilizations are 0.998, 0.101,

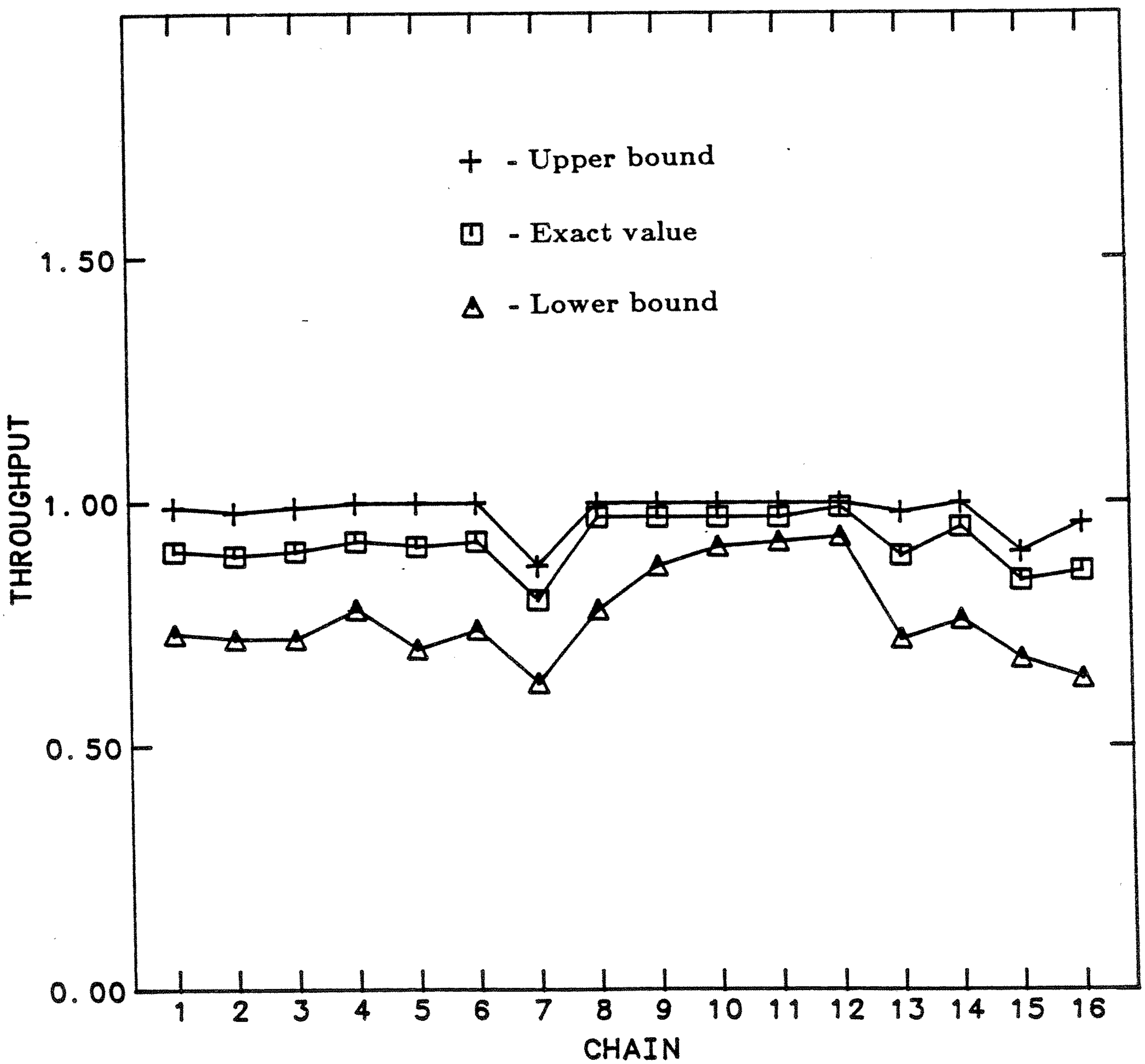

Figure 5. Throughput bounds for the first multichain network example with low utilization.

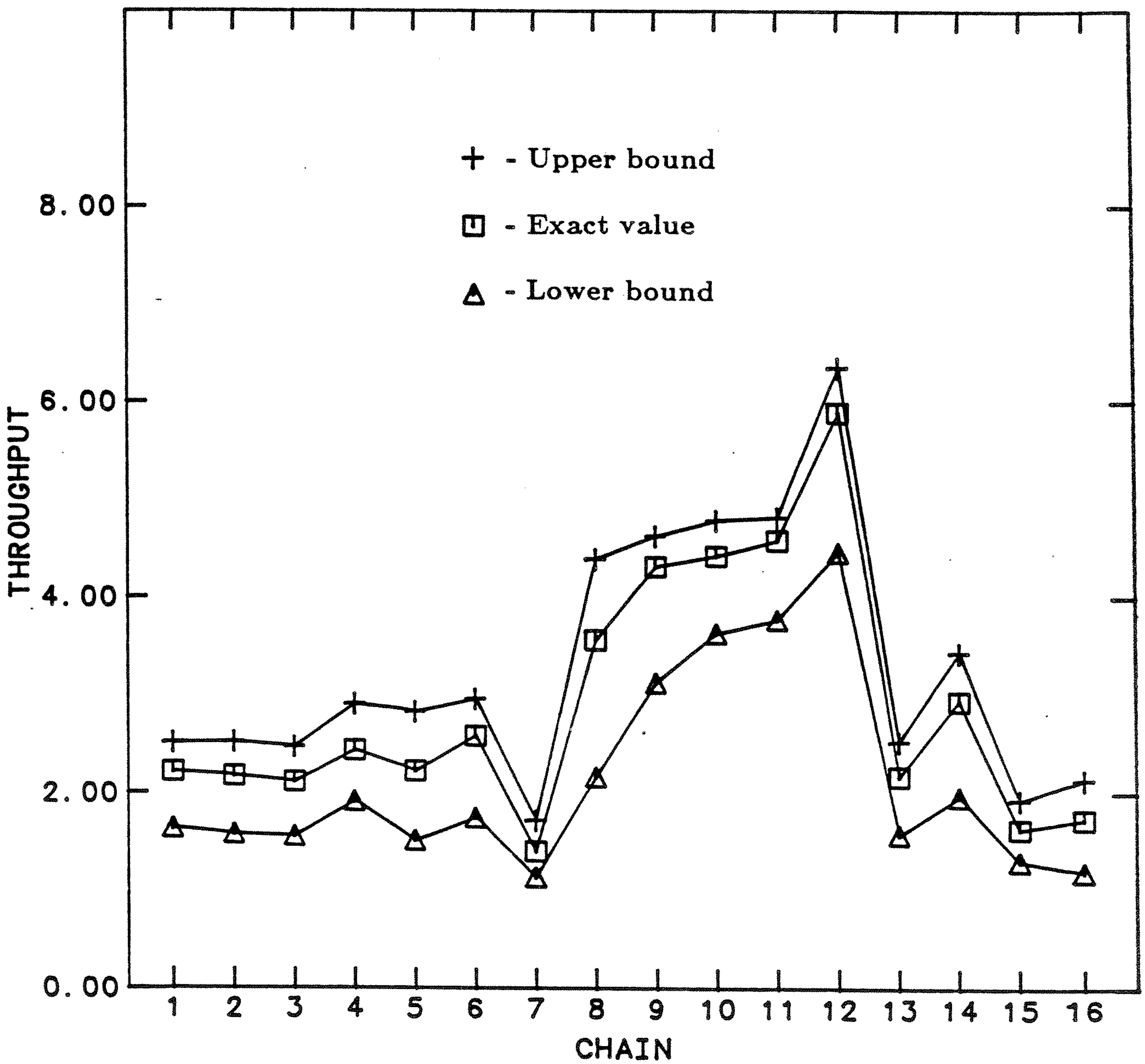

Figure 6. Throughput bounds for the first multichain network example with high utilization.

and 0.517 respectively. Because of symmetric traffic, only the results of 15 virtual channels are shown in Figure 7.

Communication channel	Mean service time (sec)
(9 , 4)	0.200
(2 , 7)	0.200
(3 ,11)	0.200
(5 ,12)	0.050
(6 ,10)	0.025
(9 , 7)	0.200
(1 , 2)	0.050
(3 , 1)	0.050
(5 ,11)	0.200
(6 , 8)	0.200
(9 , 2)	0.100
(10, 8)	0.100
(12, 3)	0.050
(4 , 7)	0.025
(5 , 9)	0.200
(6 ,12)	0.200
(10, 7)	0.100
(4 , 9)	0.200
(7 , 2)	0.200
(11, 3)	0.200
(12, 5)	0.050
(10, 6)	0.025
(7 , 9)	0.200
(2 , 1)	0.050
(1 , 3)	0.050
(11, 5)	0.200
(8 , 6)	0.200
(2 , 9)	0.100
(8 ,10)	0.100
(3 ,12)	0.050
(7 , 4)	0.025
(9 , 5)	0.200
(12, 6)	0.200
(7 ,10)	0.100

Table 6. Mean service times of communication channels in example 2.

Virtual channel	Route (in node sequence)	window size	mean service time of source server (sec)
1	11 5	2	0.10
2	2 7 4	2	0.20
3	8 6	2	0.10
4	11 3 1 2 9 4	3	0.30
5	10 6 12	2	0.30
6	6 10 8	3	0.20
7	9 2 7 4	2	0.30
8	8 10 7 9 4	2	0.10
9	4 7 9 2 1 3	2	0.10
10	5 12 3 1 2 7 9 4	2	0.20
11	2 9	2	0.30
12	3 11 5	2	0.30
13	3 12 5	2	0.30
14	7 2	2	0.10
15	10 7 9	2	0.10
16	5 11	2	0.10
17	4 7 2	2	0.20
18	6 8	2	0.10
19	4 9 2 1 3 11	3	0.30
20	12 6 10	2	0.30
21	8 10 6	3	0.20
22	4 7 2 9	2	0.30
23	4 9 7 10 8	2	0.10
24	3 1 2 9 7 4	2	0.10
25	4 9 7 2 1 3 12 5	2	0.20
26	9 2	2	0.30
27	5 11 3	2	0.30
28	5 12 3	2	0.30
29	2 7	2	0.10
30	9 7 10	2	0.10

Table 7. Routes, window sizes and mean service time of source servers for virtual channels in example 2.

From Figure 7 and the table on routes, we observe that if a virtual channel does not interact much with other virtual channels, then its throughput bounds are tight and the exact value is close to the upper bound. On the other hand, if a virtual channel interacts significantly with many other virtual channels then its

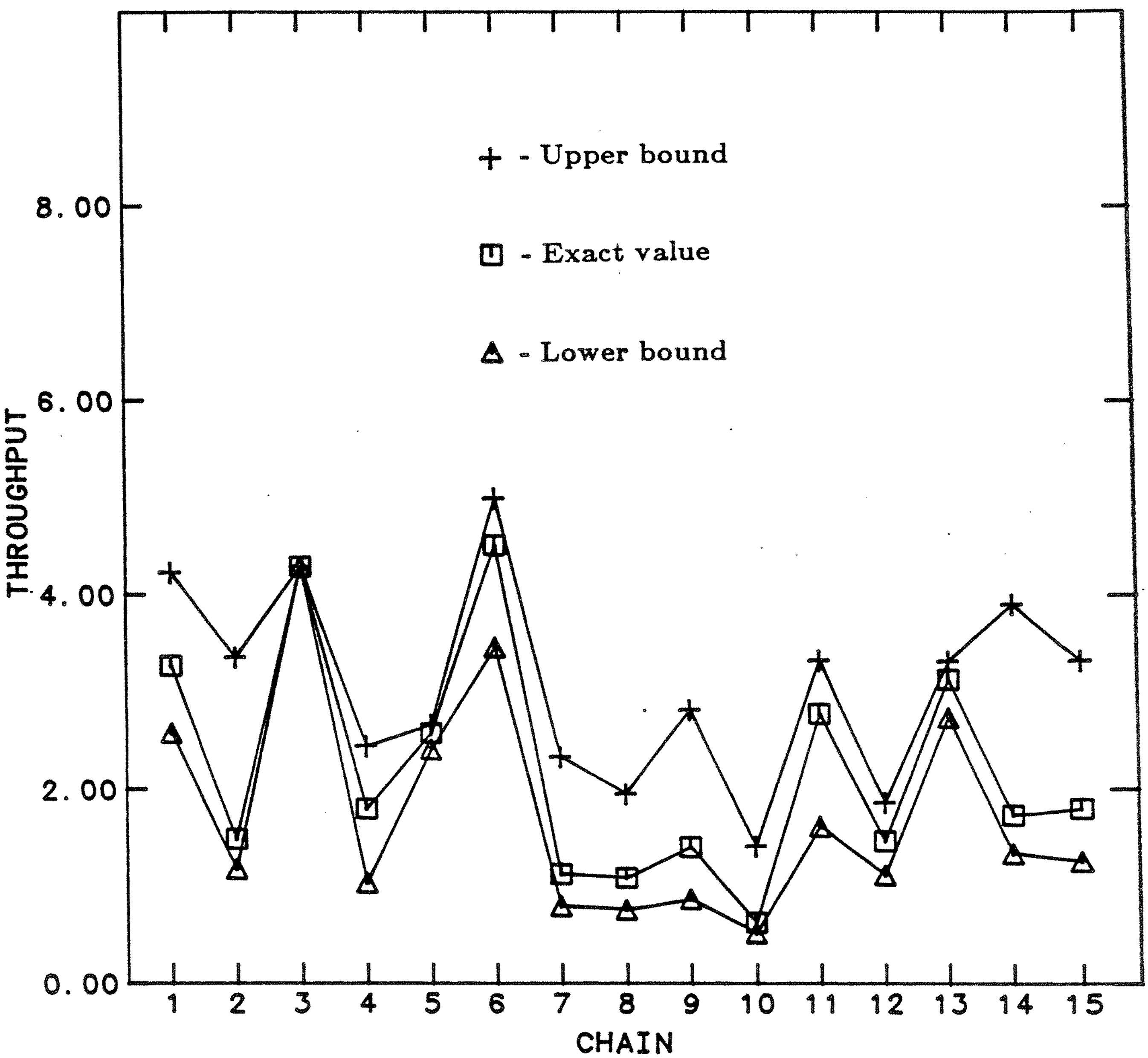

Figure 7. Throughput bounds for the second multichain network example.

throughput bounds are not so tight and the exact throughput is closer to the lower bound than the upper bound. Notice that virtual channel 3 does not intersect any other virtual channel; both its upper bound and its lower bound obtained are equal to the exact throughput. Thus the tightness of the throughput bounds presented in this section is affected by the degree of sparseness of routes in a network.

Appendix. Proofs of lemmas, theorems and corollaries.

Proof of Lemma 1:

By induction.

(1) Base case: $q_i(1)/q_j(1)=\tau_i/\tau_j$. Therefore it is true for $n=1$.

(2) Induction step: Assume it is true for $n=k$, i.e., $\frac{q_i(k)}{q_j(k)}\leq\frac{\tau_i}{\tau_j}$.

From MVA formula,

$$\frac{q_i(k+1)}{q_j(k+1)}=\frac{D_i(k+1)T(k+1)}{D_j(k+1)T(k+1)}=\frac{D_i(k+1)}{D_j(k+1)}$$

$$=\frac{\tau_i(1+q_i(k))}{\tau_j(1+q_j(k))}$$

$$\leq\frac{\tau_i}{\tau_j} \qquad \text{(because } q_i(k)\leq q_j(k))$$

Therefore it is also true for $n=k+1$. Q.E.D.

Proof of Lemma 2:

If $q_j(n)\leq\frac{\tau_j}{L_F}\sum_{m=1}^{M_F}q_m(n)$, then from Lemma 1, for any $i\leq j$

$$q_i(n)\leq\frac{\tau_i}{\tau_j}q_j(n)$$

$$\leq\frac{\tau_i}{\tau_j}\frac{\tau_j}{L_F}\sum_{m=1}^{M_F}q_m(n)$$

$$=\frac{\tau_i}{L_F}\sum_{m=1}^{M_F} q_m(n). \quad \text{Q.E.D.}$$

Proof of Lemma 3: Similar to the proof of Lemma 2.

Proof of Lemma 4:

From Lemma 3, $q_m(n) \geq \frac{\tau_m}{\tau_1} q_1(n)$ for $m=1,\ldots,M_F$. We then have,

$$\sum_{m=1}^{M_F} q_m(n) \geq \sum_{m=1}^{M_F} \frac{\tau_m}{\tau_1} q_1(n) = \frac{L_F}{\tau_1} q_1(n).$$

Hence, we have

$$q_1(n) \leq \frac{\tau_1}{L_F}\sum_{m=1}^{M_F} q_m(n).$$

Similarly, $q_{M_F}(n) \geq \frac{\tau_{M_F}}{L_F}\sum_{m=1}^{M_F} q_m(n)$. Q.E.D.

Proof of Theorem 1:

From Eq. (1), we have

$$D(n) = \sum_{m=1}^{M} D_m(n)$$

$$= \sum_{m=M_F+1}^{M} \tau_m + \sum_{m=1}^{M_F} \tau_m (1+q_m(n-1))$$

$$= \sum_{m=M_F+1}^{M} \tau_m + \sum_{m=1}^{M_F} \tau_m + \sum_{M=1}^{M_F} \tau_m q_m(n-1)$$

$$= \sum_{m=1}^{M} \tau_m + (\tau_1 q_1(n-1) + \tau_2 q_2(n-1) + \cdots + \tau_{M_F} q_{M_F}(n-1))$$

$$=L+\left[\left(\frac{\tau_1^2}{L_F}+\frac{\tau_2^2}{L_F}+\cdots+\frac{\tau_{M_F}^2}{L_F}\right)\times\left(n-1-L_D\times T(n-1)\right)\right]+\Delta$$

where

$$\Delta=\tau_1\left[q_1(n-1)-\frac{\tau_1}{L_F}\left(n-1-L_D\times T(n-1)\right)\right]$$

$$+\tau_2\left[q_2(n-1)-\frac{\tau_2}{L_F}\left(n-1-L_D\times T(n-1)\right)\right]$$

$$+\cdots$$

$$+\tau_{M_F}\left[q_{M_F}(n-1)-\frac{\tau_{M_F}}{L_F}\left(n-1-L_D\times T(n-1)\right)\right]$$

From Lemmas 2, 3 and 4, there exists an m, $1\leq m\leq M_F$ such that

$$q_i(n-1)\leq\frac{\tau_i}{L_F}\left(n-1-L_D\times T(n-1)\right)\qquad\text{for all } 1\leq i\leq m$$

and

$$q_i(n-1)\geq\frac{\tau_i}{L_F}\left(n-1-L_D\times T(n-1)\right)\qquad\text{for all } m<i\leq M_F.$$

Therefore, we have $\Delta=\Delta_1-\Delta_2$ where

$$\Delta_1=\sum_{i=m+1}^{M_F}\tau_i\left[q_i(n-1)-\frac{\tau_i}{L_F}\left(n-1-L_D\times T(n-1)\right)\right]$$

and

$$\Delta_2=\sum_{i=1}^{m}\tau_i\left[\frac{\tau_i}{L_F}\left(n-1-L_D\times T(n-1)\right)-q_i(n-1)\right].$$

Replacing $(n-1-L_D\times T(n-1))$ with $\sum_{j=1}^{M_F}q_j(n-1)$, we have

$$\Delta_1 = \sum_{i=m+1}^{M_F} \tau_i \left[q_i(n-1) - \frac{\tau_i}{L_F} \sum_{j=1}^{M_F} q_j(n-1) \right]$$

$$\geq \tau_{m+1} \left[\sum_{i=m+1}^{M_F} q_i(n-1) - \sum_{i=m+1}^{M_F} \frac{\tau_i}{L_F} \sum_{j=1}^{M_F} q_j(n-1) \right] \qquad \text{(A1)}$$

and

$$\Delta_2 = \sum_{i=1}^{m} \tau_i \left[\frac{\tau_i}{L_F} \sum_{j=1}^{M_F} q_j(n-1) - q_i(n-1) \right]$$

$$\leq \tau_m \left[\sum_{i=1}^{m} \frac{\tau_i}{L_F} \sum_{j=1}^{M_F} q_j(n-1) - \sum_{i=1}^{m} q_i(n-1) \right]. \qquad \text{(A2)}$$

The expressions inside the brackets of the right-hand sides of Eqs. (A1) and (A2) are equal because

$$\sum_{i=1}^{m} q_i(n-1) + \sum_{i=m+1}^{M_F} q_i(n-1)$$

$$= \sum_{i=1}^{m} \frac{\tau_i}{L_F} \sum_{j=1}^{M_F} q_j(n-1) + \sum_{i=m+1}^{M_F} \frac{\tau_i}{L_F} \sum_{j=1}^{M_F} q_j(n-1) = \sum_{j=1}^{M_F} q_j(n-1).$$

Hence, $\Delta_1 \geq \Delta_2$. Therefore

$$D(n) \geq L + \sum_{m=1}^{M_F} \tau_m^2 \left[n-1-L_D \times T(n-1) \right] / L_F.$$

Proof of Theorem 2:

When $n=1$ it is clearly true. Therefore we only have to prove the theorem for $n>1$. The proof can be divided into two parts.

(1) $D(n) \geq \underline{D}(n,i)$ and $T(n) \leq \overline{T}(n,i)$ for $0 \leq i \leq n-1$. This can be proved by induction on i and by using Theorem 1.

(2) $\underline{D}(n,i+1) \geq \underline{D}(n,i)$ and $\overline{T}(n,i+1) \leq \overline{T}(n,i)$ for $0 \leq i \leq n-1$. This is proved by induction as follows:

(i) When $i=1$, $\underline{D}(n,1) \geq \underline{D}(n,0)$ and $\overline{T}(n,1) \leq \overline{T}(n,0)$ from Eqs. (11) and (12).

(ii) Assume that it is true for $i=k$ then from Theorem 1 it is also true for $i=k+1$. Q.E.D.

Proof of Corollaries 1 and 2:

Compare the balanced delay bound of Kriz (which generalizes the balanced job bound)

$$L+\frac{L_F}{M_F}\left[n-1-L_D\times\overline{T}(n-1,i-1)\right]$$

with the corresponding proportional bound

$$L+\sum_{m=1}^{M_F}\frac{\tau_m^2}{L_F}\left[n-1-L_D\times\overline{T}(n-1,i-1)\right].$$

For the proportional bound to be tighter, it is sufficient to show

$$\sum_{m=1}^{M_F}\frac{\tau_m^2}{L_F}\leq\frac{L_F}{M_F}$$

which is true by virtue of the Chebyshev Inequality [Abra72], and then apply induction on i. Q.E.D.

Proof of Lemma 5:

By induction.

(1) Base case: $q_i(1)/q_j(1)=\tau_i/\tau_j$. Therefore it is true for $n=1$.

(2) Induction step: Assume it is true for $n=k$, i.e., $\frac{q_i(k)}{q_j(k)}\geq(\frac{\tau_i}{\tau_j})^k$.

From MVA formula,

$$\frac{q_i(k+1)}{q_j(k+1)}=\frac{D_i(k+1)T(k+1)}{D_j(k+1)T(k+1)}=\frac{D_i(k+1)}{D_j(k+1)}$$

$$=\frac{\tau_i(1+q_i(k))}{\tau_j(1+q_j(k))}$$

$$\geq\frac{\tau_i}{\tau_j}\left(\frac{q_i(k)}{q_j(k)}\right)^k \qquad \text{(because } q_i(k)\leq q_j(k)\text{)}$$

$$=\left(\frac{\tau_i}{\tau_j}\right)^{k+1}.$$

Therefore it is also true for $n=k+1$. Q.E.D.

Proofs of Lemmas 6-8 and Theorems 3-4 are similar to those of Lemmas 2-4 and Theorems 1-2 and are omitted.

Proof of Corollaries 3 and 4:

Compare the balanced delay upper bound of Kriz (which generalizes the balanced job bound)

$$L+\tau_{M_F}\left[n-1-L_D\times\underline{T}(n-1,i-1)\right]$$

with the corresponding proportional bound

$$L+\frac{L_F^n}{L_F^{n-1}}\left[n-1-L_D\times\underline{T}(n-1,i-1)\right].$$

For the proportional bound to be tighter, it is sufficient to show that

$$\frac{L_F^n}{L_F^{n-1}}\leq\tau_{M_F}$$

which is true by the fact that $\tau_j\leq\tau_{M_F}$ for all $j\leq M_F$. Q.E.D.

The following lemma is from [Eage83].

Lemma 9: For any i, $1\leq i\leq S$, and all j, $1\leq j\leq M$ if

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})}\geq\frac{q_j(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)}$$

then

$$\sum_{m=j}^{M} q_m(n_i-1)\geq(n_i-1)\sum_{m=j}^{M} r_m(n_{i-1}),$$

$$\frac{\sum_{m=j}^{M} \tau_m q_m(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)} \geq \frac{\sum_{m=j}^{M} \tau_m r_m(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})},$$

and

$$\frac{r_j(n_i)}{\sum_{m=j}^{M} r_m(n_i)} \geq \frac{q_j(n_i)}{\sum_{m=j}^{M} q_m(n_i)}.$$

Proof: The proof is similar to the proof of the corresponding lemmas in [Eage83] and is omitted.

Lemma 10: For two arrays whose elements are ratios of mean queue lengths $r_m(i_1)$, $r'_m(i_2)$, $m=1,2,...,M$ and $i_1, i_2>0$, if

$$\frac{r_j(i_1)}{\sum_{m=j}^{M} r_m(i_1)} \leq \frac{r'_j(i_2)}{\sum_{m=j}^{M} r'_m(i_2)} \qquad \text{for } 1 \leq j \leq M$$

then

$$\frac{\tau_j[1+r_j(i_1)\times l_1]}{\sum_{m=j}^{M} \tau_m[1+r_m(i_1)\times l_1]} \leq \frac{\tau_j[1+r'_j(i_2)\times l_2]}{\sum_{m=j}^{M} \tau_m[1+r'_m(i_2)\times l_2]}$$

for all nonnegative integers l_1 and l_2 such that $l_1 \geq l_2$ and $1 \leq j \leq M$.

Proof: Similar to part of the proof of Lemma 9.

Lemma 11:

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})} \geq \frac{r_j(n_i)}{\sum_{m=j}^{M} r_m(n_i)} \tag{A3}$$

for all i and j such that $1 \leq j \leq M$ and $1 \leq i \leq S-1$.

Proof: By induction on i.

(i) When $i=1$,

$$\frac{r_j(1)}{\sum_{m=j}^{M} r_m(1)} - \frac{r_j(n_1)}{\sum_{m=j}^{M} r_m(n_1)}$$

$$= \frac{\tau_j}{\sum_{m=j}^{M} \tau_m} - \frac{\tau_j(L+\tau_j \times n_0)}{\sum_{m=j}^{M} \tau_m(L+\tau_m \times n_0)}$$

$$= \frac{\tau_j \sum_{m=j}^{M} \tau_m(L+\tau_m \times n_0) - \tau_j(L+\tau_j \times n_0) \sum_{m=j}^{M} \tau_m}{\sum_{m=j}^{M} \tau_m \sum_{m=j}^{M} \tau_m(L+\tau_m \times n_0)}$$

$$= \frac{n_0 \times \left[\tau_j \sum_{m=j}^{M} \tau_m^2 - \tau_j \sum_{m=j}^{M} \tau_m \tau_j\right]}{\sum_{m=j}^{M} \tau_m \sum_{m=j}^{M} \tau_m(L+\tau_m \times n_0)} \geq 0.$$

This establishes the induction base.

(ii) Assume that it is true for $i=k$. From Lemma 10, it is also true for $i=k+1$.
Q.E.D.

Notice that Lemmas 9, 10 and 11 are also true for exact mean queue lengths $q_m(\cdot)$, $m=1,2,\ldots,M$ since they correspond to the special case in which the population sequence is 2,3,...,N.

Proof of Theorem 5: We only have to prove that

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})} \geq \frac{q_j(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)} \qquad \text{(A4)}$$

for all i, $1 \leq i \leq S$ and all j, $1 \leq j \leq M$.

The proof is by induction on i.

(i) When $i=1$, $r_j(1)=q_j(1)$ for $1 \leq j \leq M$. Eq. (A4) is clearly true.

(ii) Assume that it is true for $i=k$. From Lemma 9 it follows that

$$\frac{r_j(n_k)}{\sum_{m=j}^{M} r_m(n_k)} \geq \frac{q_j(n_k)}{\sum_{m=j}^{M} q_m(n_k)}. \tag{A5}$$

From Lemma 11, we have

$$\frac{q_j(n_{k+1}-1)}{\sum_{m=j}^{M} q_m(n_{k+1}-1)} \leq \frac{q_j(n_k)}{\sum_{m=j}^{M} q_m(n_k)}. \tag{A6}$$

From Eqs. (A5) and (A6), it is also true for $i=k+1$. From Lemma 9, we know that Eq. (24) is true. Q.E.D.

Proof of Corollary 5: We only have to prove that

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})} = \frac{q_j(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)}$$

for all i, $1 \leq i \leq S$ and all j, $1 \leq j \leq M$. The proof is similar to that of Theorem 5.

Proof of Theorem 6: We only have to prove that the upper bound obtained from population sequence $1 < n_1 < n_2 < \ldots < n_S = N$ is smaller than that obtained from population sequence $m_1, m_2, \ldots, m_{S-1}$, where $m_i = n_i$ for $i=1,2,\ldots,l-1$, and $m_i = n_{i+1}$ for $i=l, l+1, \ldots, S-1$, where l is an integer such that $1 \leq l \leq S-1$. It is equivalent to showing that

$$\frac{r_j(n_i)}{\sum_{m=j}^{M} r_m(n_i)} \leq \frac{r'_j(m_{i-1})}{\sum_{m=j}^{M} r'_m(m_{i-1})} \qquad \text{for } 1 \leq j \leq M \text{ and } 1 \leq i \leq S-1.$$

where $r'_j(m_i)$ denotes the value of ratio[j] in Algorithm 1 when the sequence $m_1, m_2, \ldots, m_{S-1}$ is used.

The proof is by induction on i.

(i) When $i=1$, from Lemma 11

$$\frac{r_j(n_1)}{\sum_{m=j}^{M} r_m(n_1)} \le \frac{r_j(n_0)}{\sum_{m=j}^{M} r_m(n_0)} = \frac{r'_j(m_0)}{\sum_{m=j}^{M} r'_m(m_0)} \qquad \text{for all} \quad 1 \le j \le M.$$

where $r_j(n_0)$ and $r'_j(m_0)$ denote the initial values of ratio[j] in Algorithm 1 when the corresponding sequences $n_1, n_2, \ldots, n_S$ and $m_1, m_2, \ldots, m_{S-1}$ are used. This establishes the base of the induction.

(ii) Suppose that it is true for $i=k$, i.e.,

$$\frac{r_j(n_k)}{\sum_{m=j}^{M} r_m(n_k)} \le \frac{r'_j(m_{k-1})}{\sum_{m=j}^{M} r'_m(m_{k-1})} \qquad \text{for all} \quad 1 \le j \le M.$$

from Lemma 10 and the fact that $n_k \ge m_{k-1}$ it is also true for $i=k+1$. Q.E.D.

Proof of Theorem 7: Immediate from Lemmas 9 and 10.

The following lemma, which is similar to Lemma 9, is used in the proofs of Theorems 8, 9 and 10. Its proof is similar to that of Lemma 9 and is omitted.

Lemma 12: For any i, $1 \le i \le S$, and all j, $1 \le j \le M$ if

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})} \le \frac{q_j(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)}.$$

then

$$\sum_{m=j}^{M} q_m(n_i-1) \le (n_i-1) \sum_{m=j}^{M} r_m(n_{i-1}),$$

$$\frac{\sum_{m=j}^{M} \tau_m q_m(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)} \le \frac{\sum_{m=j}^{M} \tau_m r_m(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})},$$

and

$$\frac{r_j(n_i)}{\sum_{m=j}^{M} r_m(n_i)} \le \frac{q_j(n_i)}{\sum_{m=j}^{M} q_m(n_i)}.$$

Proof of Theorem 8: We only have to prove that

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})} \le \frac{q_j(n_i-1)}{\sum_{m=j}^{M} q_m(n_i-1)} \tag{A7}$$

for all i, $1 \le i \le S$ and all j, $1 \le j \le M$.

The proof is by induction on i.

(i) When $i=1$, from Lemma 5

$$\frac{r_j(n_0)}{r_m(n_0)} \le \frac{q_j(n_0)}{q_m(n_0)} \qquad \text{for all } 1 \le j \le m \le M.$$

Therefore Eq. (A7) is true.

(ii) Assume that it is true for $i=k$. From Lemma 12 it follows that

$$\frac{r_j(n_k)}{\sum_{m=j}^{M} r_m(n_k)} \le \frac{q_j(n_k)}{\sum_{m=j}^{M} q_m(n_k)}. \tag{A8}$$

Because $n_{k+1} \le n_k$, from Lemma 11, we have

$$\frac{q_j(n_{k+1}-1)}{\sum_{m=j}^{M} q_m(n_{k+1}-1)} \ge \frac{q_j(n_k)}{\sum_{m=j}^{M} q_m(n_k)}. \tag{A9}$$

From Eqs. (A8) and (A9), it is also true for $i=k+1$. From Lemma 12, we know that Eq. (25) is true. Q.E.D.

In Lemma 13, to be given in the following, $r_j(n_i)$ denotes the value of ratio[j] when the population size is n_i during the execution of Algorithm 3. This lemma is used in the proof of Theorem 9.

Lemma 13:

$$\frac{r_j(n_0)}{\sum_{m=j}^{M} r_m(n_0)} \le \frac{r_j(n_1)}{\sum_{m=j}^{M} r_m(n_1)}$$

for all j such that $1 \leq j \leq M$.

Proof: It is sufficient to prove

$$\frac{r_j(n_0)}{r_i(n_0)} \leq \frac{r_j(n_1)}{r_i(n_1)} \quad \text{for } 1 \leq j < i \leq M.$$

For the sake of clarity, we shall replace n_0 with n for the balance of this proof. We then have $r_i(n_0) = \tau_i^n$ and $r_i(n_1) = \tau_i(1+n \times \tau_i^n / L^n)$ for $1 \leq i \leq M$.

Thus

$$\frac{r_j(n_1)}{r_i(n_1)} - \frac{r_j(n)}{r_i(n)}$$

$$= \frac{\tau_j(1+n \times \tau_j^n / L^n)}{\tau_i(1+n \times \tau_i^n / L^n)} - \frac{\tau_j^n}{\tau_i^n}. \tag{A10}$$

Multiply Eq. (A10) by $L^n \tau_i^{n+1}(1+n \times \tau_i^n / L^n) / \tau_j$. We obtain

$$L^n \tau_i^{n-1}(1+n \times \tau_j^n / L^n) - L^n \tau_j^{n-1}(1+n \times \tau_i^n / L^n)$$

$$= L^n(\tau_i^{n-1} - \tau_j^{n-1}) - n\, \tau_i^{n-1} \tau_j^{n-1}(\tau_i - \tau_j) \tag{A11}$$

We shall prove that Eq. (A11) is greater than or equal to zero by induction on M.

(i) Case of $M=2$ (i.e., there are only two fixed-rate service centers): We shall prove this base case by induction on n.

(a) When $n=2$ (i.e., $n_1=3$), we have

$$L^2(\tau_i - \tau_j) - n\, \tau_i \tau_j(\tau_i - \tau_j) = (\tau_i - \tau_j)\left[\tau_i^2 + \tau_j^2 - 2\tau_i \tau_j\right] = (\tau_i - \tau_j) \times (\tau_i - \tau_j)^2 \geq 0$$

where $L^2 = \tau_i^2 + \tau_j^2$. This establishes the induction base for n.

(b) Assume that Eq. (A11) is true for $n=k$. After factorizing Eq. (A11) and eliminating $(\tau_i - \tau_j)$, the induction hypothesis becomes

$$L^k(\tau_i^{k-2}+\tau_i^{k-3}\tau_j+\ldots+\tau_i\tau_j^{k-3}+\tau_j^{k-2}) - k\,\tau_i^{k-1}\tau_j^{k-1} \geq 0 \qquad \text{(A12)}$$

Multiply Eq. (A12) by $\tau_i \tau_j$, we get

$$\tau_j L^k(\tau_i^{k-1}+\tau_i^{k-2}\tau_j+\ldots+\tau_i^2\tau_j^{k-3}+\tau_i\tau_j^{k-2}) - k\,\tau_i^k\tau_j^k$$

$$=\tau_j L^k(\tau_i^{k-1}+\tau_i^{k-2}\tau_j+\ldots+\tau_i\tau_j^{k-2}+\tau_j^{k-1}) - (k+1)\tau_i^k\tau_j^k$$

$$-(\tau_j^k L^k - \tau_i^k\tau_j^k) \geq 0. \qquad \text{(A13)}$$

When $n=k+1$, the LHS of Eq. (A12) becomes the following

$$L^{k+1}(\tau_i^{k-1}+\tau_i^{k-2}\tau_j+\ldots+\tau_i\tau_j^{k-2}\tau_j^{k-1}) - (k+1)\tau_i^k\tau_j^k$$

$$>\tau_j L^k(\tau_i^{k-1}+\tau_i^{k-2}\tau_j+\ldots+\tau_i\tau_j^{k-2}\tau_j^{k-1}) - (k+1)\tau_i^k\tau_j^k \geq 0. \qquad \text{(A14)}$$

Eq. (A14) is true because $(\tau_j^k L^k - \tau_i^k\tau_j^k)$ in Eq. (A13) is nonnegative and τ_j is smaller than τ_i. We have thus proved the base case for M.

(ii) Assume that Eq. (A11) is true for $M=l$, i.e.,

$$\left[\sum_{m=1}^{l}\tau_m^n\right](\tau_i^{n-1}-\tau_j^{n-1}) - n\,\tau_i^{n-1}\tau_j^{n-1}(\tau_i-\tau_j) \geq 0.$$

It is clear that

$$\left[\sum_{m=1}^{l+1}\tau_m^n\right](\tau_i^{n-1}-\tau_j^{n-1}) - n\,\tau_i^{n-1}\tau_j^{n-1}(\tau_i-\tau_j) \geq 0.$$

Therefore

$$\frac{r_j(n_0)}{r_i(n_0)} \leq \frac{r_j(n_1)}{r_i(n_1)} \qquad \text{for } 1 \leq j < i \leq M. \qquad \text{Q.E.D.}$$

Proof of Theorem 9: We only have to prove that the lower bound obtained from population sequence $n_1, n_2, \ldots, n_S = N$ is larger than that obtained from population sequence $m_1, m_2, \ldots, m_{S-1}$, where $m_i = n_i$ for $i = 1,2,\ldots,l-1$, and

$m_i = n_{i+1}$ for $i = l, l+1, \ldots, S-1$, where l is an integer such that $1 < l \le S-1$. It is equivalent to showing that

$$\frac{r_j(n_i)}{\sum_{m=j}^{M} r_m(n_i)} \ge \frac{r'_j(m_{i-1})}{\sum_{m=j}^{M} r'_m(m_{i-1})} \qquad \text{for } 1 \le j \le M \text{ and } 1 \le i \le S-1$$

where $r'_j(m_i)$ denotes the value of ratio[j] in Algorithm 3 when the sequence $m_1, m_2, \ldots, m_{S-1}$ is used.

The proof is by induction on i.

(i) When $i = 1$, from Lemma 13

$$\frac{r_j(n_1)}{\sum_{m=j}^{M} r_m(n_1)} \ge \frac{r_j(n_0)}{\sum_{m=j}^{M} r_m(n_0)} = \frac{r'_j(m_0)}{\sum_{m=j}^{M} r'_m(m_0)} \qquad \text{for all } 1 \le j \le M.$$

where $r_j(n_0)$ and $r'_j(m_0)$ denote the initial values of ratio[j] in Algorithm 3 when the corresponding sequences $n_1, n_2, \ldots, n_S$ and $m_1, m_2, \ldots, m_{S-1}$ are used. This establishes the base of the induction.

(ii) Suppose that it is true for $i = k$, i.e.,

$$\frac{r_j(n_k)}{\sum_{m=j}^{M} r_m(n_k)} \ge \frac{r'_j(m_{k-1})}{\sum_{m=j}^{M} r'_m(m_{k-1})} \qquad \text{for all } 1 \le j \le M$$

from Lemma 10 and the fact that $n_k \le m_{k-1}$ it is also true for $i = k+1$. Q.E.D.

Proof of Theorem 10: We only have to prove that the lower bound obtained from population sequence $n_1 \ge n_2 \ge \cdots \ge n_S = N$ is larger than that obtained from population sequence $m_1 \ge m_2 \ge \cdots \ge m_S = N$, where $m_i \ge n_i$ for $i = 1, 2, \ldots, S-1$. It is equivalent to showing that

$$\frac{r_j(n_{i-1})}{\sum_{m=j}^{M} r_m(n_{i-1})} \ge \frac{r'_j(m_{i-1})}{\sum_{m=j}^{M} r'_m(m_{i-1})} \qquad \text{for } 1 \le j \le M \text{ and } 1 \le i \le S.$$

where $r'_j(m_i)$ denotes the value of ratio[j] in Algorithm 3 when the sequence $m_1, m_2, \ldots, m_{S-1}$ is used. Let $n_0 = n_1 - 1$ and $m_0 = m_1 - 1$.

The proof is by induction on i.

(i) When $i=1$, we have $n_0 \le m_0$ and

$$\frac{r_j(n_0)}{r_m(n_0)} = \frac{\tau_j^{n_0}}{\tau_m^{n_0}} \ge \frac{\tau_j^{m_0}}{\tau_m^{m_0}} = \frac{r'_j(m_0)}{r'_m(m_0)} \qquad \text{for all} \quad 1 \le j \le m \le M.$$

Therefore

$$\frac{r_j(n_0)}{\sum_{m=j}^{M} r_m(n_0)} \ge \frac{r'_j(m_0)}{\sum_{m=j}^{M} r'_m(m_0)} \qquad \text{for all} \quad 1 \le j \le M.$$

This establishes the base of the induction.

(ii) Suppose that it is true for $i=k$, i.e.,

$$\frac{r_j(n_k)}{\sum_{m=j}^{M} r_m(n_k)} \ge \frac{r'_j(m_k)}{\sum_{m=j}^{M} r'_m(m_k)} \qquad \text{for all} \quad 1 \le j \le M.$$

from Lemma 10 and the fact that $n_k \le m_k$ it is also true for $i=k+1$. Q.E.D.

Proof of Theorem 11:

From MVA, we have

$$D_k(\underline{n}) = \sum_{m=1}^{M} \tau_{mk} \left[1 + \sum_{h=1}^{K} q_{mh}(\underline{n} - \underline{1}_k) \right] \text{ (if } m \text{ is a delay service center then } \sum_{h=1}^{K} q_{mh}(\underline{n} - \underline{1}_k) = 0)$$

$$= \sum_{m=1}^{M} \tau_{mk} + \sum_{m=1}^{M_F} \tau_{mk} \sum_{h=1}^{K} q_{mh}(\underline{n} - \underline{1}_k)$$

$$= L_k + \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{mk} \sum_{\substack{h=1 \\ \text{chain } h \text{ visits } m}}^{K} q_{mh}(\underline{n} - \underline{1}_k)$$

$$= L_k + \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{mk} \sum_{\substack{h=1 \\ \text{chain } h \text{ visits } m}}^{K} \tau_{mh} \underline{T}_h(\underline{n} - \underline{1}_k) + \alpha$$

where $\alpha = \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{mk} \sum_{\substack{h=1 \\ \text{chain } h \text{ visits } m}}^{K} \left[q_{mh}(\underline{n} - \underline{1}_k) - \tau_{mh} \underline{T}_h(\underline{n} - \underline{1}_k) \right].$

We then have

$$\alpha \leq \sum_{\substack{h=1 \\ \text{chain } h \\ \text{intersects chain } k}}^{K} \tau_{max,h,k} \sum_{\substack{m=1 \\ m \text{ in chains } h,k}}^{M_F} \left[q_{mh}(\underline{n} - \underline{1}_k) - \tau_{mh} \underline{T}_h(\underline{n} - \underline{1}_k) \right]$$

$$\leq \sum_{\substack{h=1 \\ \text{chain } h \\ \text{intersects chain } k}}^{K} \tau_{max,h,k} \left[n_h - L_{D,h} \underline{T}_h(\underline{n} - \underline{1}_k) - \sum_{\substack{m=1 \\ m \text{ in chain } h \\ m \text{ not in chain } k}}^{M_F} \tau_{mh} \underline{T}_h(\underline{n} - \underline{1}_k) \right.$$

$$\left. - \sum_{\substack{m=1 \\ m \text{ in chains } h,k}}^{M_F} \tau_{mh} \underline{T}_h(\underline{n} - \underline{1}_k) \right] - \tau_{max,k,k}$$

$$= \sum_{\substack{h=1 \\ \text{chain } h \\ \text{intersects chain } k}}^{K} \tau_{max,h,k} \left[n_h - L_{D,h} \underline{T}_h(\underline{n} - \underline{1}_k) - \sum_{\substack{m=1 \\ m \text{ in chain } h}}^{M_F} \tau_{mh} \underline{T}_h(\underline{n} - \underline{1}_k) \right] - \tau_{max,k,k}$$

$$= \sum_{\substack{h=1 \\ \text{chain } h \\ \text{intersects chain } k}}^{K} \tau_{max,h,k} \left[n_h - L_{D,h} \underline{T}_h(\underline{n} - \underline{1}_k) - L_{F,h} \underline{T}_h(\underline{n} - \underline{1}_k) \right] - \tau_{max,k,k}$$

$$= \sum_{\substack{h=1 \\ \text{chain } h \\ \text{intersects chain } k}}^{K} \tau_{max,h,k} \left[n_h - L_h \underline{T}_h(\underline{n} - \underline{1}_k) \right] - \tau_{max,k,k}$$

On the other hand, we have

$$\alpha \geq \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{min,k} \left[q_{mk}(\underline{n} - \underline{1}_k) - \tau_{mk} \underline{T}_k(\underline{n} - \underline{1}_k) \right]$$

$$= \tau_{min,k} \left[\sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} q_{mk}(\underline{n}-\underline{1}_k) - \sum_{\substack{m=1 \\ m \text{ in chain } k}}^{M_F} \tau_{mk}\,\underline{T}_k(\underline{n}-\underline{1}_k) \right]$$

$$\geq \tau_{min,k} \left[n_k - 1 - L_{D,k}\,\overline{T}_k(\underline{n}-\underline{1}_k) - L_{F,k}\,\underline{T}_k(\underline{n}-\underline{1}_k) \right] \quad \text{Q.E.D.}$$

Proof of Corollary 6: Immediate from Theorem 11.

References

Abra72.
M. Abramowitz and I.E. Stegun, eds., *Handbook of mathematical functions,* Dover Publ., New York, 1972.

Bask75.
F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios, "Open, closed and mixed networks of queues with different class of customers," *JACM,* vol. 22, pp. 248-260, 1975.

Buze73.
J.P. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Comm. ACM,* vol. 16, pp. 527-531, 1973.

Chan75.
K.M. Chandy, U. Herzog, and L. Woo, "Parametric analysis of queueing networks," *IBM J. Res. Develop.*, vol. 19, pp. 36-42, 1975.

Eage83.
D.L. Eager and K.C. Sevcik, "Performance bound hierarchies for queueing networks," *ACM Trans. Computer Systems,* vol. 1, no. 2, pp. 99-115, May 1983.

Hoym82.
K.P. Hoyme, S.C. Bruell, P.V. Afshari, and R.Y. Kain, *A tree-structured MVA algorithm,* Computer Science Department, University of Minnesota, Minneapolis, Minnesota, June 1982. Technical Report 82-17

Kriz84.
J. Kriz, "Throughput bounds for closed queueing networks," *Performance Evaluation,* vol. 4, no. 1, 1984.

Lam82.
S.S. Lam, "Dynamic scaling and growth behavior of queueing network normalization constants," *Journal of ACM,* vol. 29, pp. 492-513, 1982.

Lam82.
S.S. Lam and J.W. Wong, "Queueing network models of packet switching networks, part 2: Networks with population size constraints," *Performance Evaluation*, vol. 2, pp. 161-180, 1982.

Lam83.
S.S. Lam and Y.L. Lien, "A tree convolution algorithm for the solution of queueing networks," *Comm. ACM*, vol. 26, no. 3, pp. 203-215, 1983.

Lam85.
S.S. Lam and Ching-Tarng Hsieh, "Models and algorithms for the design of store-and-forward communication networks," *Conference record, International Conference on Communications*, Chicago, June 1985.

Litt61.
J.D.C. Little, "A proof of the queueing formula: $L = \lambda W$," *Operations Research*, vol. 9, pp. 383-387, 1961.

Munt74.
R.R. Muntz and J. Wong, "Asymptotic properties of closed queueing network models," *Proc. of the 8th Annual Princeton Conf. on Info. Sci. and Systems*, Princeton University, March 1974.

Reis75.
M. Reiser and H. Kobayashi, "Queueing networks with multiple closed chains: Theory and computational algorithms," *IBM J. Res. Develop.*, vol. 21, pp. 283-294, 1975.

Reis80.
M. Reiser and S. Lavenberg, "Mean value analysis of closed multichain queueing networks," *Journal of ACM*, vol. 27, no. 2, pp. 313-322, April 1980.

Sevc79.
K.C. Sevcik and I. Mitrani, "The distribution of queueing network states at input and output instants," *Proc. fourth International Symposium on Modeling and Performance Evaluation of Computer Systems*, North-Holland Publishing Co., Vienna, Austria, 1979. The International Institute for Systems Analysis

Step84.
L.E. Stephens and L.W. Dowdy, "Convolutional bound hierarchies," *Proceedings of the 1984 ACM Sigmetrics Conference*, pp. 120-133, Cambridge, Massachusetts, August 1984.

Suri83.
R. Suri, *Generalized quick bounds for performance of queueing networks,* Aiken Computation Laboratory, Harvard University, Cambridge, Massachussetts, 1983. Report TR-05-83

Tucc82.
S. Tucci and C.H. Sauer, *The tree MVA algorithm,* IBM Thomas Watson

Research Center, Yorktown Heights, NY, April 1982. IBM Research Report

Zaho82.
J. Zahorjan, K.C. Sevcik, D.L. Eager, and B. Galler, "Balanced job bound analysis of queueing networks," *Comm. ACM*, vol. 25, no. 2, pp. 132-141, 1982.