The Annals of Applied Probability
1994, Vol. 4, No. 1, 43–75

OPTIMIZATION OF MULTICLASS QUEUEING NETWORKS: POLYHEDRAL AND NONLINEAR CHARACTERIZATIONS OF ACHIEVABLE PERFORMANCE[1]

BY DIMITRIS BERTSIMAS, IOANNIS CH. PASCHALIDIS AND JOHN N. TSITSIKLIS

Massachusetts Institute of Technology

We consider open and closed multiclass queueing networks, with Poisson arrivals (for open networks), exponentially distributed class dependent service times and class dependent deterministic or probabilistic routing. The performance objective is to minimize, over all sequencing and routing policies, a weighted sum of the expected response times of different classes. Using a powerful technique involving quadratic or higher order potential functions, we propose methods for deriving polyhedral and nonlinear sets that contain the set of achievable response times under stable and preemptive scheduling policies. By optimizing over these sets, we obtain lower bounds on achievable performance. In the special case of single station networks (multiclass queues and Klimov's model) and homogeneous multiclass networks, the polyhedron derived is exactly equal to the achievable region. Consequently, the proposed method can be viewed as the natural extension of conservation laws to multiclass queueing networks. We apply the same approach to closed networks to obtain upper bounds on the optimal throughput. We check the tightness of our bounds by simulating heuristic policies and we find that the first order approximation of our method is at least as good as simulation-based existing methods. In terms of computational complexity and in contrast to simulation-based existing methods, the calculation of our first order bounds consists of solving a linear programming problem with a number of variables and constraints that is polynomial (quadratic) in the number of classes in the network. The ith order approximation leads to a convex programming problem in dimension $O(R^{i+1})$, where R is the number of classes in the network, and can be solved efficiently using techniques from semidefinite programming.

1. Introduction. A *multiclass queueing network* is one that services multiple types of customers that may differ in their arrival processes, service requirements, routes through the network and costs per unit of waiting time. The fundamental optimization problem that arises in open networks is to

Received January 1993; revised June 1993.

[1]A preliminary version of this paper appeared in the proceedings of the Workshop on Hierarchical Control for Real-Time Scheduling of Manufacturing Systems, Lincoln, New Hampshire, October 1992. The full paper was presented at the ORSA/TIMS Conference on November 2, 1992. Research supported by NSF Grant ECS-85-52419, by a Presidential Young Investigator award DDM-91-58118 with matching funds from Draper Laboratory, by the Leaders for Manufacturing Program at MIT and by ARO Grant DAAL 03-92-G0309.

AMS 1991 *subject classifications*. 60K25, 90B15, 90B22.

Key words and phrases. Queueing networks, optimization, bounds, achievable space.

determine an optimal policy for sequencing and routing customers in the network that minimizes a linear combination of the expected sojourn times of each customer class. The fundamental optimization problem that arises in a multiclass closed network is the maximization of throughput. There are both *sequencing* and *routing* decisions involved in these optimization problems. A *sequencing policy* determines which type of customer to serve at each station of the network, while a *routing policy* determines the route of each customer.

There are several important applications of such problems: packet-switching communication networks with different types of packets and priorities, job shop manufacturing systems, and scheduling of multiprocessors and multiprogrammed computer systems, to name a few.

The control of multiclass queueing networks is a mathematically challenging problem. In order to achieve optimality, stations have to decide how to sequence competing customer types at each point in time, based on information about the load conditions of various other stations. Additionally, customers can choose their route through the network, taking into account the current state of various queues. These interactions between various stations create serious dependencies among them and prevent not only optimization, but even performance analysis of a given policy. To indicate the difficulty of the problem, it is worth mentioning that even with Poisson arrivals and *class dependent* exponential service times, and for the simplest possible policy, FCFS (First Come First Serve), product form or analytical solutions are not available. Naturally, optimizing a multiclass queueing network is an even harder problem. Thus, not surprisingly, simulation is the most common practice among researchers and practitioners as a tool of evaluating heuristic policies. However, even if simulation is used for a proposed heuristic policy, it may not give any indication on how close to optimality this policy is.

These considerations lead us to the first contribution of the present paper. In the tradition of discrete optimization in the mathematical programming community, we develop a sequence of lower bounds to the optimal cost. We also compare the lower bounds with proposed heuristic policies in order to evaluate the closeness to optimality of these policies. In the relatively simple examples that we studied, we found that our first order bounds are comparable to the "pathwise" bounds derived in Ou and Wein (1992) by means of a simulation-based method. Moreover, our first order bound consists of solving a linear programming problem with $O(R^2)$ variables and $O(R^2)$ constraints, R being the number of classes in the network. In general our ith order bound consists of solving a nonlinear programming problem with $O(R^{i+1})$ variables and $O(R^{i+1})$ constraints.

A second, and in our opinion significant, contribution of the present work is to expand on the idea that rather than optimizing a stochastic and dynamic system (in particular, a multiclass queueing network), it is important to characterize all the achievable performance vectors (in the case of a multiclass open queueing network, the vector of expected response times for the different classes in the network). In this way, one is able to formulate a stochastic and dynamic optimization problem as a mathematical program-

ming problem. This has serious advantages because one can use advanced algorithmic methods from a mature field, and also consider more general objective functions (for example, involving variances). With respect to this objective, we obtain a sequence of progressively more complicated nonlinear approximations that are progressively closer to the exact achievable region. We note that, except for a simple example in Gelenbe and Mitrani (1980), we do not know of any other example of a nonlinear characterization.

In the case of simpler systems (a multiclass queue [Gelenbe and Mitrani (1980); Kleinrock (1976)], a single server network [Klimov (1974); Tsoucas (1991)] and a homogeneous open network [Ross and Yao (1987)]), our first order characterization is exact, that is, it is identical to the characterization in Gelenbe and Mitrani (1980), Ross and Yao (1987) for the multiclass queue and homogeneous network, respectively, and consistent with the characterization of Tsoucas (1991) derived using conservation laws. In all of these cases we also find a reformulation of the achievable region with a polynomial number of variables and constraints, which is interesting from a combinatorial point of view. As a result, our approach can be seen as the natural extension of conservation laws to multiclass queueing networks. By optimizing over an approximation of the achievable region, we obtain bounds to the optimal value. In the case where the characterization is exact, we find the exact value as well as an optimal policy.

The third methodological contribution of this paper is the use of potential functions to derive mathematical programming formulations for stochastic systems. Potential function methods in science have a rather rich history and a vast literature. From Liapunov functions to prove stability of dynamical systems, to proof methods in linear programming and network flows in recent times, potential function methods have been established as a very powerful proof technique. For stochastic systems, Kushner in the 1960's used potential function methods to prove stability. Regarding the use of potential function methods to bound performance in queueing systems, Kumar (1992) uses a method of Meyn and Down (1994) (who used it to prove stability of generalized Jackson networks) to derive one inequality (as opposed to a family of inequalities) and obtain a bound on the achievable performance in an open network with deterministic routing (reentrant line). Kumar points out in his paper that his bound is rather weak. In the present paper we realize the full potential of the method and significantly expand its power by introducing an arbitrary potential function that gives a family of bounds (linear and nonlinear) that takes into account high order interactions of various classes. We also propose the use of symbolic manipulation of multivariable polynomials (e.g., using Mathematica or Maple) for automatically deriving the constraints of the approximating spaces; these constraints are then fed to an LP solver that calculates the lower bound. In other words, the user provides the data (arrival and service rates, the topology of the network and the routing) and receives as output a lower bound on achievable performance.

The fourth methodological contribution is a general technique for generating nonlinear (convex) constraints. We show that optimization over this set of

constraints can be performed efficiently (in polynomial time) using cutting plane methods and techniques from semidefinite programming. Our ideas are influenced by the recent developments in deriving lower bounds for integer programming problems using semidefinite programming [Lovasz and Schrijver (1990), Alizadeh (1992)].

Literature review. With respect to characterizing the performance region of stochastic and dynamic systems, there have been some interesting developments in the last decade. Gelenbe and Mitrani (1980) first showed, using conservation laws, that the performance region of a multiclass queue can be described as a polyhedron. Federgruen and Groenevelt (1988) advanced the theory by observing that in certain special cases of multiclass queues the polyhedron has a very special structure (it is a polymatroid) that gives rise to very simple optimal policies (the $c\mu$ rule). Shanthikumar and Yao (1992) generalized the theory further by observing that if a system satisfies conservation laws, then the underlying performance space is necessarily a *polymatroid polytope* and the optimal policy is a strict priority rule. Their results partially extend to some rather restricted queueing networks, in which they assume that all the different classes of customers have the same routing probabilities and the same service requirements at each station of the network [see also Ross and Yao (1987)]. Tsoucas (1991) derives the achievable region for scheduling a multiclass nonpreemptive M/G/1 queue with Bernoulli feedback introduced by Klimov (1974). Finally, Bertsimas and Niño-Mora (1992) generalize the idea of conservation laws and show that for all systems that satisfy these generalized conservation laws, the underlying performance space is a polyhedron with very strong structural properties, called an *extended polymatroid* in Bhattacharya, Georgiadis and Tsoucas (1992). Optimization of a linear function over extended polymatroids can be achieved by an adaptive greedy algorithm [see Bhattacharya, Georgiadis and Tsoucas (1992) and Bertsimas and Niño-Mora (1992)]. The framework of Bertsimas and Niño-Mora (1992) includes all the cases we mentioned before, as well as the multiarmed bandit problem [Gittins (1989)], branching bandits [Weiss (1988)] and some deterministic scheduling problems.

Perhaps one of the most successful approaches for controlling multiclass queueing networks in heavy traffic, which offers valuable new insights, is to use *Brownian network models*, where the stochastic processes in the network are modeled as Brownian motions. Introduced by Harrison (1986) and further explored by Wein, this approach proposes heuristic policies that typically outperform more traditional ones. This approach has been more successful in closed networks [Harrison and Wein (1990)] and networks with controllable input [Wein (1990a, b)], but has not been as successful in scheduling open networks. In particular, Harrison and Wein (1989) show that a threshold policy is consistent with the optimality conditions for a Brownian two-station, three-class network that we also consider in this paper (Section 3). Wein (1990a, b) proposes priority rules and admission control policies in open networks where admission control is allowed. For a nice survey of the

heavy-traffic approach for optimization of multiclass networks, the reader is referred to Kelly and Laws (1992). For a thorough survey of the rather vast literature on routing in stochastic systems, see Walrand (1988).

In the only study that concerns lower bounds for general networks, Ou and Wein (1992) derive *pathwise* lower bounds for general open queueing networks with deterministic routing. They also calculate steady-state bounds by averaging over all sample paths. A distinct characteristic of their approach is that *simulation* is needed for the computation of the bounds, to be contrasted with our approach where bounds are calculated by solving a mathematical programming problem (linear or nonlinear) with all the parameters known in closed form from the data of the network.

Chen, Yang and Yao (1991) follow a *stochastic intensity control* approach for the specific network topology studied in Harrison and Wein (1989), which we also study in Section 3. They model the arrival and service processes as counting processes with controllable stochastic intensities, their objective being to minimize a discounted cost function over an infinite time horizon, and they establish a switching curve structure.

Structure of the paper. The rest of the paper is organized as follows. In Section 2, we formally define the sequencing problem for multiclass open networks and the class of policies that we will be considering. In Section 3, we start with a well-studied, simple, open network in order to illustrate the fundamental ideas in our approach without excessive notation. The particular structure of this network allows us to derive further bounds, which are based on different ideas. In Section 4, we introduce two variations of a method for obtaining polyhedral descriptions (first order methods) of a general open multiclass network with Poisson arrivals and exponentially distributed, class dependent service times with deterministic or probabilistic routing. In Section 5, we extend our methodology to include routing decisions and closed networks. In Section 6, we explain how the methodology can be extended to derive tighter approximations of the achievable region by taking into account higher order interactions and by introducing an additional family of nonlinear constraints. We also describe how ideas from semidefinite programming can be used to handle this family of nonlinear constraints. In Section 7, we prove that our method produces the exact characterization for an M/M/1 multiclass queue and for Klimov's problem with Poisson arrivals and exponentially distributed service times. In Section 8, we apply our first order methods to specific network examples and report numerical results. Finally, in Section 9, we include some concluding remarks.

2. Problem formulation. In this section, we define the class of queueing networks we will consider, the class of policies we allow and establish our notation.

Here, as well as in Sections 3 and 4, we will consider an open multiclass queueing network involving only sequencing decisions (routing is given) with N single server stations (nodes) and R different job classes. The class of a job

summarizes all relevant information on the current characteristics of a job, including the node at which it is waiting for service. In particular, jobs waiting at different nodes are by necessity of different classes and a job changes class whenever it moves from one node to another. Let $\sigma(r)$ be the node associated with class r and let C_i be the set of all classes r such that $\sigma(r) = i$. When a job of class r completes service at node i, it can become a job of class s, with probability p_{rs}, and move to server $\sigma(s)$. It can also exit the network, with probability $p_{r0} = 1 - \sum_{s=1}^{R} p_{rs}$. There are R independent Poisson arrival streams, one for each customer class. The arrival process for class r customers has rate λ_{0r} and these customers join station $\sigma(r)$. The service time of class r jobs is assumed to be exponentially distributed with rate μ_r. Note that jobs of different classes associated with the same node can have different service requirements. We assume that service times are independent of each other and independent of the arrival process.

Whenever there is one or more customer waiting for service at a node, we can choose which, if any, of these customers should be served next. (Notice that we are not restricting ourselves to work-conserving policies.) In addition, we allow for the possibility of preemption. A rule for making such decisions is called a *policy*. Let $n_r(t)$ be the number of class r customers present in the network at time t. The vector $\mathbf{n}(t) = (n_1(t), \ldots, n_R(t))$ will be called the *state* of the system at time t. A policy is called *Markovian* if each decision it makes is determined as a function of the current state. It is then clear that under a Markovian policy, the queueing network under study evolves as a continuous-time Markov chain.

For technical reasons, we will only study Markovian policies satisfying the following assumption:

ASSUMPTION A. (a) The Markov chain $\mathbf{n}(t)$ has a unique invariant distribution.

(b) For every class r, we have $E[n_r^2(t)] < \infty$, where the expectation is taken with respect to the invariant distribution.

Let n_r be the steady-state mean of $n_r(t)$ and let x_r be the mean response time (waiting plus service time) of class r customers. We are interested in determining a scheduling policy that minimizes a linear cost function of the form $\sum_{r=1}^{R} c_r x_r$. We approach this problem by trying to determine the set X of all vectors $(x_1, \ldots, x_R)$ that are obtained by considering different policies that satisfy Assumption A. By minimizing the cost function $\sum_{r=1}^{R} c_r x_r$ over the set X, we can then obtain the cost of an optimal policy.

The set of X is not necessarily convex and this leads us to considering its convex hull X'. Any vector $x \in X'$ is the performance vector associated with a generally non-Markovian policy obtained by "time sharing" or randomization of finitely many Markovian policies. Note also that if the minimum over X' of a linear function is attained, then it is attained at some element of X. We will refer to X' as the *region of achievable performance* or, simply, the *achievable region*.

Given that the exact characterization of the achievable region appears to be very difficult, in general, we provide methods that approximate the achievable region by a larger set. Minimizing $\sum_{r=1}^{R} c_r x_r$ over this larger set provides us with a lower bound on the cost of an optimal policy.

3. A simple two-station network. In this section, we use a simple example to illustrate the methodology that will be developed in its full generality in the next sections.

We consider the network, with two types (not classes) of customers, depicted in Figure 1. Type 1 customers visit stations 1 and 2, in that order, before exiting the network and type 2 customers visit only station 1 before exiting the network. We define *class* 1, 3 customers to be type 1 customers at stations 1, 2, respectively, and *class* 2 customers to be type 2 customers at station 1. Let λ_1 and λ_2 be the arrival rates for customers of class 1 and 2, respectively. Let μ_1 and μ_2 be the service rates at stations 1 and 2, respectively. (We assume that both customer types have the same service requirements at the first station.) In order to ensure that at least one stable policy exists, we assume that $\lambda_1 + \lambda_2 < \mu_1$ and $\lambda_1 < \mu_2$.

Let n_i and x_i be as defined in Section 2. We are interested in a scheduling policy that minimizes a linear cost function of the form $\sum_{i=1}^{3} c_i x_i$, where c_1, c_2, c_3 are given finite weights. Note that for this problem, a policy amounts to a rule according to which the first server can decide which customer class, if any, to serve.

In the remainder of this section, we illustrate our methodology for deriving a lower bound on the optimal cost. To this effect, we use a systematic procedure for obtaining $2^3 - 1$ inequalities that must be satisfied by the vector (x_1, x_2, x_3). (Note that we have one inequality for each nonempty set of

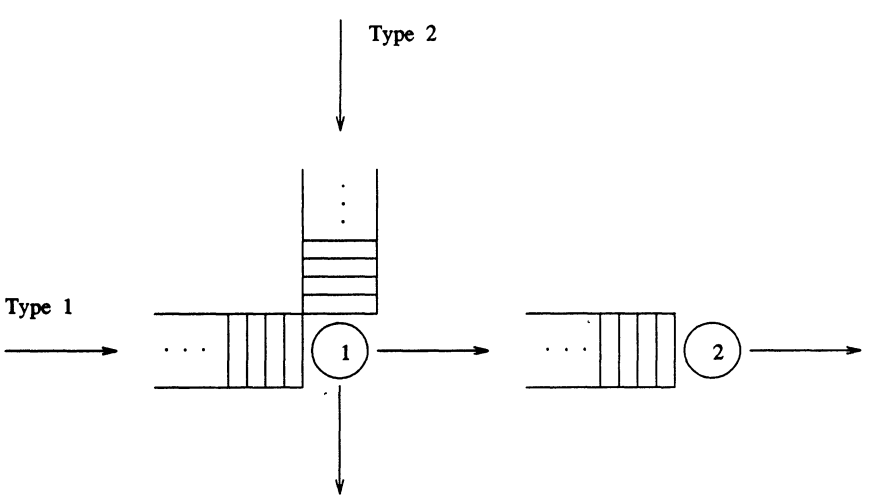

FIG. 1. *A simple two-station network.*

classes.) The derivation of these inequalities readily extends to more general networks (Section 4). We also obtain some additional inequalities by less systematic (but still generalizable) methods.

3.1. *The main inequalities.* The result that follows is derived by making use of potential functions $(R^S(t))^2$, where

$$R^S(t) = \sum_{i \in S} f_S(i) n_i(t). \tag{1}$$

S is a set of classes and the quantities $f_S(i)$ are positive constants, which we will call f-parameters.

THEOREM 3.1. *For the network defined in this section and for every policy satisfying Assumption A, the following inequalities hold*:

$$\lambda_1 x_1 + \lambda_2 x_2 \geq \frac{\lambda_1 + \lambda_2}{\mu_1 - \lambda_1 - \lambda_2}, \tag{2}$$

$$x_1 \geq \frac{1}{\mu_1 - \lambda_1}, \tag{3}$$

$$x_2 \geq \frac{1}{\mu_1 - \lambda_2}, \tag{4}$$

$$x_3 \geq \frac{1}{\mu_2}, \tag{5}$$

$$x_1 + x_3 \geq \frac{1}{\mu_2 - \lambda_1}, \tag{6}$$

$$\lambda_2 x_2 + \lambda_1 x_3 \geq \frac{\lambda_1 + \lambda_2}{\mu_1 + \mu_2 - \lambda_2}, \tag{7}$$

$$2\lambda_1 x_1 + \lambda_2 x_2 + \lambda_1 x_3 \geq \frac{3\lambda_1 + \lambda_2}{\mu_1 + \mu_2 - 2\lambda_1 - \lambda_2}. \tag{8}$$

PROOF. We will only prove (2). The other inequalities can be derived similarly. For the interested reader, the complete derivation can be found in Paschalidis (1992).

The analysis is much simplified by "uniformizing" the Markov chain under study, so that the total transition rate out of a state is the same for all states. To this effect, we visualize the process as if server 2 were always working on a class 3 customer. However, if $n_3(t) = 0$, we say that server 2 is working on a fictitious customer and a service completion does not lead to a new state. Similarly, we visualize the first server as if it were always working concurrently on a customer of class 1 and a customer of class 2, at a total rate of $2\mu_1$. A service completion at server 1 corresponding to a class 1 customer is a fictitious one that leaves the state unchanged, unless $n_1(t) \neq 0$ and the

scheduling policy had decided that a class 1 customer should be served. With the foregoing conventions, the transition rate is $\lambda_1 + \lambda_2 + 2\mu_1 + \mu_2$, which we assume, for convenience, to be equal to 1.

Let τ_k be the sequence of times at which a transition (due to a real or fictitious customer) occurs. We assume that the state vector $\mathbf{n}(t)$ is a right-continuous function of time so that $\mathbf{n}(\tau_k)$ refers to the state right after the kth transition. We will be using the notation $1\{\cdot\}$ to denote the indicator function; that is, $1\{A\} = 1$ if event A occurs and zero otherwise. Finally, by $B_r(t)$ we denote the event that node $\sigma(r)$ is busy with a class r customer at time t and by $\bar{B}_r(t)$, its complement.

The derivation of (2) uses the function $R(t) = f(1)n_1(t) + f(2)n_2(t)$. We have

$$\begin{aligned} E\big[R^2(\tau_{k+1})|\mathbf{n}(\tau_k)\big] &= \lambda_1\big(R(\tau_k) + f(1)\big)^2 + \lambda_2\big(R(\tau_k) + f(2)\big)^2 \\ &\quad + \mu_1 1\{B_1(\tau_k)\}\big(R(\tau_k) - f(1)\big)^2 + \mu_1 1\{\bar{B}_1(\tau_k)\}R^2(\tau_k) \\ &\quad + \mu_1 1\{B_2(\tau_k)\}\big(R(\tau_k) - f(2)\big)^2 + \mu_1 1\{\bar{B}_2(\tau_k)\}R^2(\tau_k) \\ &\quad + \mu_2 R^2(\tau_k). \end{aligned}$$

We expand the squared terms and observe that if we set $f(1) = f(2) = f$, the term

$$2\mu_1 1\{B_1(\tau_k)\}R(\tau_k)f(1) + 2\mu_1 1\{B_2(\tau_k)\}R(\tau_k)f(2)$$

is equal to $2\mu_1 1\{\text{server 1 busy at } \tau_k\}R(\tau_n)f$. Using the fact

$$1\{\text{server 1 busy at } \tau_k\} \le 1, \tag{9}$$

we obtain

$$\begin{aligned} E\big[R^2(\tau_{k+1})|\mathbf{n}(\tau_k)\big] &\ge R^2(\tau_k) + \lambda_1 f^2 + \lambda_2 f^2 \\ &\quad + \mu_1 1\{B_1(\tau_k)\}f^2 + \mu_1 1\{B_2(\tau_k)\}f^2 \\ &\quad - 2\mu_1 R(\tau_k)f + (2\lambda_1 f + 2\lambda_2 f)R(\tau_k). \end{aligned} \tag{10}$$

Recall that after uniformizing the Markov chain under consideration, the transition rate out of a state became the same for all states. Given this property, it is easily verified that the unique invariant distribution of the continuous-time Markov chain is the same as the (necessarily unique) invariant distribution of the embedded discrete-time Markov chain $\mathbf{n}(\tau_k)$. In particular, under the invariant distribution of the two chains, we have

$$E\big[R(\tau_{k+1})\big] = E\big[R(\tau_k)\big] = E\big[R(t)\big] \qquad \forall\, t, k \tag{11}$$

and

$$E\big[R^2(\tau_{k+1})\big] = E\big[R^2(\tau_k)\big] = E\big[R^2(t)\big] \qquad \forall\, t, k. \tag{12}$$

Furthermore, (12) and Assumption A imply that $E[R^2(\tau_k)]$ is finite.

We now consider the Markov chain $\mathbf{n}(\tau_k)$ under its invariant distribution and take expectations of both sides of (10). We use (11) to replace $E[R(\tau_k)]$ by $E[R(t)]$, (12) to cancel the R^2 terms and the relation $E[1\{B_j(\tau_k)\}] = \lambda_j/\mu_1$,

$j = 1, 2$. We then rearrange terms to obtain

$$E[R(\tau_k)] \geq \frac{(\lambda_1 + \lambda_2)f}{\mu_1 - \lambda_1 - \lambda_2}.$$

We finally use the relation $n_i = \lambda_i x_i$, $i = 1, 2$, to obtain (2). □

DISCUSSION. Note that (2) is the same as the conservation law for the multiclass M/M/1 queue [see Gelenbe and Mitrani (1980), Chapter 6], with an inequality sign instead of an equality. Within the class of policies we are considering the conservation law does not hold since we allow idling. If, however, we restrict ourselves to work-conserving policies, then it is possible to derive the conservation law using our approach. See Section 7 for more details on the application of our approach to the multiclass queue.

Note also, that (3) and (4) have a very intuitive explanation: They are the two inequalities that together with the conservation law define the achievable region for the multiclass queue at station 1. In Section 7 we prove, for the general case of multiclass queue and for work-conserving policies, that (3) and (4) hold with equality if we give preemptive priority to customers of class 1 and class 2, respectively.

3.1.1. ADDITIONAL INEQUALITIES. We note that (5) simply states that the mean response time of class 3 is no smaller than its mean service time $1/\mu_2$. In fact, the inequalities of Theorem 3.1 allow x_3 to be as small as $1/\mu_2$. This is reasonable because policies of the following type lead to zero waiting time for class 3 customers: serve class 1 customers only if server 2 is idle and has no customers in its queue. On the other hand, any such policy runs the risk of being unstable. To see this, suppose that $\lambda_2 = 0$. For the system to remain stable, there have to be $2\lambda_1$ service completions per unit time. Given that the preceding policies only allow one server to work at a time, such policies are unstable if $2\lambda_1 > \max(\mu_1, \mu_2)$. We conclude that x_3 must be strictly larger than $1/\mu_2$ if a policy is stable and $2\lambda_1 > \max(\mu_1, \mu_2)$. This argument can be carried out in more detail and leads to the following result; its proof is omitted and can be found in Paschalidis (1992).

THEOREM 3.2. *Suppose that* $2\lambda_1 > \max(\mu_1, \mu_2)$. *Then, for every policy satisfying Assumption A, we have*

$$x_3 \geq \frac{2\lambda_1 - \max(\mu_1, \mu_2)}{\mu_1 + \mu_2 - \max(\mu_1, \mu_2)} \frac{\mu_1}{\mu_2(\mu_1 + \mu_2)} + \frac{1}{\mu_2}. \tag{13}$$

Another bound is obtained as follows. If we set $c_1 = c_3 = 1$ and $c_2 = 0$, it is obvious that an optimal policy gives lowest priority to class 2 customers and processes customers of class 1 or 3 without any idling. However, then customers of class 1 and 3 evolve according to a tandem queue for which the

value of $x_1 + x_3$ is known to be equal to $1/(\mu_1 - \lambda_1) + 1/(\mu_2 - \lambda_1)$. For an arbitrary policy, the value of $x_1 + x_3$ is at least that large and we have

$$x_1 + x_3 \geq \frac{1}{\mu_1 - \lambda_1} + \frac{1}{\mu_2 - \lambda_1}. \tag{14}$$

We are able to derive the bound (14) because we could find a choice of the cost coefficients c_i for which an optimal policy and its cost is known. This suggests that we also consider the case where $c_3 = 0$. For this case, we are dealing with the problem of priority scheduling of a two-class queue. An optimal policy is given by the well-known $c\mu$ rule and its cost is also known. However, for reasons that will become clearer in Section 7, the bounds that are obtained via this approach do not provide any new information, but are subsumed by the bounds of Theorem 3.1.

As discussed in Section 2, we can use the bounds derived in this section to provide a lower bound on the cost of an optimal policy. This lower bound can be computed by minimizing $\sum_{i=1}^{3} c_i x_i$ subject to the constraints of Theorems 3.1 and 3.2, and the additional constraint (14). Some numerical results can be found in Section 8.

4. Sequencing of multiclass open networks: Approximate polyhedral characterization. In this section, we derive bounds on the achievable performance region for a general open multiclass queueing network when only sequencing decisions are involved. We will be using the model and the notation of Section 2. We first derive a set of inequalities by generalizing the method of the previous section. We then propose a nonparametric variation of the method that yields tighter and computationally more efficient bounds. The nonparametric variation has also been derived independently by Kumar and Kumar (1993). Our development of the general method using potential functions predates the work of Kumar and Kumar (1993) and was published in Bertsimas, Paschalidis and Tsitsiklis (1992).

4.1. *A parametric method.* The traffic equations for our network model take the form

$$\lambda_r = \lambda_{0r} + \sum_{r'=1}^{R} \lambda_{r'} p_{r'r}, \qquad r = 1, \ldots, R. \tag{15}$$

We assume that the inequality

$$\sum_{r \in C_i} \frac{\lambda_r}{\mu_r} < 1$$

holds for every node i. This ensures that there exists at least one policy under which the network is stable.

Let us consider a set S of classes. We consider a potential function of the form $(R^S(t))^2$, where

$$R^S(t) = \sum_{r \in S} f_S(r) n_r(t) \tag{16}$$

and where $f_S(r)$ are constants to be referred to as f-parameters. For any set S of classes, we will use a different set of f-parameters, but in order to avoid overburdening our notation, the dependence on S will not be shown explicitly.

We will impose the following condition on the f-parameters. Although it may appear unmotivated at this point, the proof of Theorem 4.1 suggests that this condition leads to tighter bounds. We assume that for each S the following statement holds. For any node i, the value of the expression

$$\mu_r\left[\sum_{r'\in S} p_{rr'}(f(r)-f(r'))+\sum_{r'\notin S} p_{rr'}f(r)\right] \tag{17}$$

is nonnegative and the same for all $r \in C_i \cap S$, and will be denoted by f_i. If $C_i \cap S$ is empty, we define f_i to be equal to zero.

We then have the following theorem.

THEOREM 4.1. *For any set S of classes, for any choice of the f-parameters satisfying the restriction* (17) *and for any policy satisfying Assumption A, the following inequality holds*:

$$\sum_{r\in S}\lambda_r f(r)x_r \geq \frac{N'(S)}{D'(S)}, \tag{18}$$

where

$$N'(S)=\sum_{r\in S}\lambda_{0r}f^2(r)+\sum_{r\notin S}\lambda_r\sum_{r'\in S}p_{rr'}f^2(r')$$
$$+\sum_{r\in S}\lambda_r\left[\sum_{r'\in S}p_{rr'}(f(r)-f(r'))^2+\sum_{r'\notin S}p_{rr'}f^2(r)\right],$$
$$D'(S)=2\left[\sum_{i=1}^{N}f_i-\sum_{r\in S}\lambda_{0r}f(r)\right],$$

S being a subset of the set of classes and x_r being the mean response time of class r.

PROOF. The steps are similar to the proof of Theorem 3.1. We first uniformize the Markov chain so that the transition rate at every state is equal to

$$\nu=\sum_r\lambda_{0r}+\sum_r\mu_r.$$

The idea is again to pretend that every class is being served with rate μ_r, but a service completion is a fictitious one unless a customer of class r is being served in actuality. Without loss of generality we scale time so that $\nu = 1$. Let $\tau_{k,}$ be the sequence of transition times for the uniformized chain. Again, by $B_r(t)$ we denote the event that node $\sigma(r)$ is busy with a class r customer at time t and denote by $\overline{B}_r(t)$ its complement. As in Theorem 3.1, we assume that the process $\mathbf{n}(t)$ is right-continuous.

We have the following recursion for $R(\tau_k)$:

$$
\begin{aligned}
&E\left[R^2(\tau_{k+1})|\mathbf{n}(\tau_k)\right] \\
&\quad = \sum_{r\in S} \lambda_{0r}(R(\tau_k) + f(r))^2 + \sum_{r\notin S} \lambda_{0r} R^2(\tau_k) \\
&\qquad + \sum_{r\in S} \mu_r 1\{B_r(\tau_k)\}\left[\sum_{r'\in S} p_{rr'}(R(\tau_k) - f(r) + f(r'))^2\right. \\
&\qquad\qquad \left. + \sum_{r'\notin S} p_{rr'}(R(\tau_k) - f(r))^2\right] \\
&\qquad + \sum_{r\in S} \mu_r 1\{\bar{B}_r(\tau_k)\} R^2(\tau_k) \\
&\qquad + \sum_{r\notin S} \mu_r 1\{B_r(\tau_k)\}\left[\sum_{r'\in S} p_{rr'}(R(\tau_k) + f(r'))^2 + \sum_{r'\notin S} p_{rr'} R^2(\tau_k)\right] \\
&\qquad + \sum_{r\notin S} \mu_r 1\{\bar{B}_r(\tau_k)\} R^2(\tau_k).
\end{aligned}
$$

In the above equation, we use the convention that the set of classes $r' \notin S$ also contains the case $r' = 0$, which corresponds to the external world of the network. (Recall that p_{r0} is the probability that a class r customer exits the network after completion of service.) We now use the assumption that the f-parameters satisfy (17). Then the term

$$
2\sum_{r\in S} \mu_r 1\{B_r(\tau_k)\}\left[\sum_{r'\in S} p_{rr'} R(\tau_k)(f(r) - f(r')) + \sum_{r'\notin S} p_{rr'} R(\tau_k) f(r)\right]
$$

can be written as

$$
\sum_{i=1}^{N} f_i R(\tau_k) 1\{\text{server } i \text{ busy from some class } r \in S \cap C_i \text{ at } \tau_k\}.
$$

(Recall that we defined $f_i = 0$ for those stations i having $C_i \cap S$ empty.) To bound the preceding term, we use the fact that the indicator is at most 1. It should now be apparent why we selected f-parameters satisfying (17). By doing so, we were able to aggregate certain indicator functions before bounding them by 1.

In addition, to bound the term

$$
\sum_{r\notin S} 2\mu_r 1\{B_r(\tau_k)\} \sum_{r'\in S} p_{rr'} R(\tau_k) f(r')
$$

we use the inequality $1\{B_r(\tau_k)\} \geq 0$.

We apply all of these bounds to our recursion for $R(\tau_k)$. We then take expectations of both sides. For the same reasons as in the proof of Theorem 3.1, we can take expectations with respect to the invariant distribution (these expectations are finite due to Assumption A) and we can replace $E[R(\tau_k)]$ by $E[R(t)]$. After some elementary algebra and rearrangements, using (17) and the relation (valid in steady state) $E[1\{B_r(\tau_k)\}] = \lambda_r/\mu_r$, we finally obtain (18). □

REMARKS. In order to apply Theorem 4.1, we must choose some f-parameters that satisfy (17). We do not know whether there always exists a choice of the f-parameters that provides dominant bounds. However, even if this were the case, it would probably be difficult to determine these "best" f-parameters. Later in this section, we show that finding the best f-parameters is not so important because there is a nonparametric variation of this bounding method that yields tighter bounds.

The proof method in Theorem 4.1 is similar to the one used by Kumar (1992) [who attributes it to Meyn and Down (1994)]. He dealt with a network with deterministic routing and with special structure (reentrant line), and only considered the case where the f-parameters were the "remaining number of stages" in order to obtain a single and fairly crude lower bound on the average number of customers in the system. The flexibility in the choice of the f-parameters that we have introduced, along with the aggregation of certain indicator functions, yields much tighter bounds.

Let us now specify one choice of the f-parameters that satisfies (17). For a set S of classes, (17) yields

$$f_i = \mu_r f(r) \sum_{r' \in S} p_{rr'} - \mu_r \sum_{r' \in S} p_{rr'} f(r') + \mu_r f(r) \sum_{r' \notin S} p_{rr'} \qquad \forall\, r \in C_i \cap S,$$

which implies

$$\frac{f_i}{\mu_r} = f(r) - \sum_{r' \in S} p_{rr'} f(r') \qquad \forall\, r \in C_i \cap S.$$

Thus, due to (17), in order to explicitly determine the f-parameters, it suffices to select nonnegative constants f_i, for each station i with $C_i \cap S$ nonempty. One natural choice of these f_i's that appears to provide fairly tight bounds is to let $f_i = 1$ for all stations i with $C_i \cap S$ nonempty. This leads to $f_S(r)$ being equal to the expected remaining processing time until a job of class r exits the set of classes S. With this choice, the parameters $f_S(r)$ can be determined by solving the system of equations

$$f_S(r) = \frac{1}{\mu_r} + \sum_{r' \in S} p_{rr'} f_S(r'), \qquad r \in S. \tag{19}$$

Moreover, this choice of the f-parameters causes the denominator of (18) to be of form $1 - \sum_{r \in S} \lambda_r / \mu_r$, which is the natural heavy traffic term; this is a key reason why we believe that it leads to tight bounds. Our claim is also supported by the fact that in Klimov's problem (see Section 7), this choice of the f-parameters yields an exact characterization.

Based on Theorem 4.1, a lower bound on the optimal cost can be found as follows. For every nonempty set of classes S, choose some f-parameters that satisfy the assumptions of Theorem 4.1 and obtain a linear inequality on the vector $(x_1, \ldots, x_R)$. Then, a lower bound is obtained by minimizing $\sum_{r=1}^{R} c_r x_r$ subject to these $2^R - 1$ inequalities. Note that this is a linear programming problem.

4.2. *A nonparametric bounding method.* In this subsection, we present a *nonparametric method* for deriving constraints on the achievable performance region. We use again a function of the form

$$R(t) = \sum_{r=1}^{R} f(r) n_r(t), \tag{20}$$

where $f(r)$ are scalars that we call f-parameters. We use the same notation as in Section 4.1 and we also introduce $B_{0i}(t)$ to denote the event that node i is idle at time t. We then define

$$I_{rr'} = E\big[1\{B_r(\tau_k)\} n_{r'}(\tau_k)\big] \tag{21}$$

and

$$N_{ir'} = E\big[1\{B_{0i}(\tau_k)\} n_{r'}(\tau_k)\big], \tag{22}$$

where $1\{\cdot\}$ is the indicator function and the expectations are taken with respect to the invariant distribution.

THEOREM 4.2. *For every scheduling policy satisfying Assumption A, the following relations hold*:

$$\begin{aligned} &2\mu_r I_{rr} - 2\sum_{r'=1}^{R} \mu_{r'} p_{r'r} I_{r'r} - 2\lambda_{0r}\lambda_r x_r \\ &\quad = \lambda_{0r} + \lambda_r(1 - p_{rr}) + \sum_{r' \neq r} \lambda_{r'} p_{r'r}, \qquad r = 1, \ldots, R, \end{aligned} \tag{23}$$

and

$$\begin{aligned} &\mu_r I_{rr'} + \mu_{r'} I_{r'r} - \sum_{w=1}^{R} \mu_w p_{wr} I_{wr'} - \sum_{w=1}^{R} \mu_w p_{wr'} I_{wr} - \lambda_{0r}\lambda_{r'} x_{r'} - \lambda_{0r'}\lambda_r x_r \\ &\quad = -\lambda_r p_{rr'} - \lambda_{r'} p_{r'r} \qquad \forall\, r, r' \text{ such that } r > r', \end{aligned} \tag{24}$$

$$\sum_{r \in C_i} I_{rr'} + N_{ir'} = \lambda_{r'} x_{r'}, \qquad I_{rr'} \geq 0, \qquad N_{ir'} \geq 0, \qquad x_i \geq 0. \tag{25}$$

PROOF. We uniformize as in Theorem 4.1 and proceed similarly to obtain the recursion

$$\begin{aligned} &E\big[R^2(\tau_{k+1}) | \mathbf{n}(\tau_k)\big] \\ &\quad = \sum_{r=1}^{R} \lambda_{0r}\big(R(\tau_k) + f(r)\big)^2 \\ &\qquad + \sum_{r=1}^{R} \mu_r 1\{B_r(\tau_k)\} \Bigg[\sum_{r'=1}^{R} p_{rr'}\big(R(\tau_k) - f(r) + f(r')\big)^2 \\ &\qquad\qquad + p_{r0}\big(R(\tau_k) - f(r)\big)^2 \Bigg] \\ &\qquad + \sum_{r=1}^{R} \mu_r 1\{\bar{B}_r(\tau_k)\} R^2(\tau_k). \end{aligned}$$

Rearranging terms and taking expectations with respect to the invariant distribution, we obtain

$$2\sum_{r=1}^{R}\mu_r\left[\sum_{r'=1}^{R}p_{rr'}(f(r)-f(r'))+p_{r0}f(r)\right]E\left[1\{B_r(\tau_k)\}R(\tau_k)\right]$$

$$-2\sum_{r=1}^{R}\lambda_{0r}f(r)E\left[R(\tau_k)\right] \tag{26}$$

$$=\sum_{r=1}^{R}\lambda_{0r}f^2(r)+\sum_{r=1}^{R}\lambda_r\left[\sum_{r'=1}^{R}p_{rr'}(f(r)-f(r'))^2+p_{r0}f^2(r)\right].$$

Moreover, it is seen from (20) and (21) that

$$E\left[1\{B_r(\tau_k)\}R(\tau_k)\right]=\sum_{r'=1}^{R}f(r')I_{rr'}.$$

Let us define the vector $f=(f(1),\ldots,f(R))$. We note that both sides of (26) are quadratic functions of f. In particular, (26) can be written in the form

$$f^TQf=f^TQ_0f \tag{27}$$

for some symmetric matrices Q, Q_0. Since (27) is valid for all choices of f, we must have $Q=Q_0$. It only remains to carry out the algebra needed in order to determine the entries of the matrices Q and Q_0. From (26), equality of the rth diagonal entries of Q and Q_0 yields (23), and equality of the off-diagonal terms yields (24). Due to symmetry, it suffices to consider $r>r'$. Finally, since the events $B_r(\tau_k)=$"server i busy from class r at τ_k," $r\in C_i$, and $B_{0i}(\tau_k)=$"server i idle at τ_k" are mutually exclusive and exhaustive, we obtain (25). □

REMARK 1. An alternative derivation of the equalities of Theorem 4.2 is as follows: We consider a test function g and write

$$E\{E[g(\mathbf{n}(\tau_{k+1}))|\mathbf{n}(\tau_k)]\}=E[g(\mathbf{n}(\tau_k))]$$

as long as the indicated expectations exist. By rewriting the previous equality in terms of the instantaneous transition rate matrix for the Markov chain $\mathbf{n}(t)$ and by introducing some new variables, we obtain certain relations between the variables. In particular, (23) can be derived by considering test functions of the form $g(\mathbf{n}(t))=n_r^2(t)$, while (24) can be derived by considering the test functions of the form $g(\mathbf{n}(t))=n_r(t)n_{r'}(t)$. Intuitively, we expect that these quadratic test functions capture some of the interaction among different customer classes.

Although the two methods of derivation [using the potential function $R(t)$ or the test functions $g(\cdot)$] are entirely equivalent, we chose to present the method using potential functions for two reasons:

(a) It makes the connection with conservation laws more apparent (see Section 7).

(b) More importantly, it permits *the automatic* derivation of constraints for arbitrary networks. The different constraints are generated by collecting terms in a multivariable polynomial, which is easily done using a symbolic manipulation program such as Mathematica or Maple, and then can be fed to an LP solver. We thus obtain a software tool that receives a description of the queueing network and its parameters and outputs a lower bound on the achievable performance.

REMARK 2. The polyhedron defined in Theorem 4.2 contains as much information on the region of achievable performance as the polyhedron defined in Theorem 4.1. Both polyhedra are derived using the same basic recursion for $R(\tau_k)$, but in the nonparametric approach no inequalities are introduced, in constrast to the approach of Theorem 4.1, where certain inequalities were used to bound certain terms, leading to possible loss of tightness. Our next theorem formally proves such a relation between the two polyhedra and establishes that the nonparametric approach is at least as powerful as our first approach.

THEOREM 4.3. *If the variables* $\{x_r, I_{rr'}, N_{ir'};\ r, r' = 1,\dots,R,\ i = 1,\dots,N\}$ *satisfy the constraints in Theorem* 4.2, *then the variables* $\{x_r,\ r = 1,\dots,R\}$ *satisfy the constraints in Theorem* 4.1.

PROOF. Let the variables $\{x_r, I_{rr'}, N_{ir'};\ r, r' = 1,\dots,R,\ i = 1,\dots,N\}$ satisfy the constraints in Theorem 4.2. Since (27) holds for every choice of the f-parameters, it is seen that we can write down an equality for every nonempty set of classes S if we set to zero the f-parameters corresponding to classes outside S. For any such S, it is apparent from (25) that

$$\sum_{r\in S\cap C_i} I_{rr'} + \sum_{r\notin S\cap C_i} I_{rr'} + N_{ir'} = \lambda_{r'} x_{r'},$$

which implies that

$$\sum_{r\in S\cap C_i} I_{rr'} \le n_{r'}$$

and

$$E\big[1\{\text{server } \sigma(r) \text{ busy from some class } r\in S\cap C_i \text{ at } \tau_k\} n_{r'}(\tau_k)\big] \le n_{r'}. \tag{28}$$

Now recall that in the proof of Theorem 4.1 we used that

$$1\{\text{server } \sigma(r) \text{ busy from some class } r\in S\cap C_i \text{ at } \tau_k\} \le 1 \tag{29}$$

and

$$1\{B_r(\tau_k)\} \ge 0 \tag{30}$$

in order to get the bound (18). That is, we first wrote down the recursive equation, we then applied (29) and (30) and we finally took expectations to get (18). It can be seen that exactly the same bound is obtained by first

writing down the recursive equation, then taking expectations and finally using (28) along with the positivity constraint for the variables $I_{rr'}$. Thus, from the equality in (27) corresponding to the subset S, the inequality (18) is derived by using (25). □

By minimizing $\sum_{r=1}^{R} c_r x_r$ subject to the constraints of Theorem 4.2 we obtain a lower bound that is no smaller than the one obtained using Theorem 4.1. In addition, the linear program in Theorm 4.1 only involves $O(R^2)$ variables and constraints. This should be contrasted with the linear program associated to our nonparametric variation of the method, which involved R variables, but $O(2^R)$ constraints. Although in most cases the nonparametric method is preferable, in certain special cases (Section 7) the polyhedron defined in Theorem 4.1 has special structure (it is an extended polymatroid), which leads to an efficient one-pass greedy algorithm for finding an optimal solution.

5. Extensions: Routing and closed networks. In this section we briefly describe several generalizations of the methods introduced in the previous section. In particular, we treat networks where routing is subject to optimization and closed networks. We will only outline our methodology. The interested reader can find the details in Bertsimas, Paschalidis and Tsitsiklis (1992a).

5.1. *Routing and sequencing.* We extend our nonparametric method to multiclass open queueing networks that allow both routing and sequencing decisions. The framework and the notation is exactly the same as in Section 4. Instead of the routing probabilities $p_{rr'}$ being given, we control whether a customer of class r becomes a customer of class r'. For this reason, we introduce $p_{rr'}(\tau_k)$ to denote the probability (which is under our control) that class r becomes r' at time τ_{k+1}, given that we had a class r service completion at time τ_{k+1}. For each class r, we are given a set F_r of classes to which a class r customer can be routed. (If F_r is a singleton for every r, the problem is reduced to the class with no routing decisions allowed.)

The procedure for obtaining the approximate achievable region is similar to the proof of Theorem 4.2 except that the constants $p_{rr'}$ are replaced by the random variables $p_{rr'}(\tau_k)$ in the main recursion. We also need to define some additional variables. As in (21) and (22) in Section 4, these variables will be expectations of certain products of certain random variables; the routing random variables $p_{rr'}(\tau_k)$ will also appear in such products.

An important difference from Section 4 is that the application of the nonparametric method to $R(t)$ also yields the traffic equations for the network. These traffic equations are now part of the characterization of the achievable region because they involve expectations of the decision variables $p_{rr'}(\tau_k)$, whereas in Section 4 they involved the constants $p_{rr'}$. Application of the method to $R^2(t)$ provides more equations that, with some definitional relations between variables, similar to (25), complete the characterization.

Rather than providing the full details here, we refer the reader to the example in Section 8.2.

5.2. *Closed networks.* The methodology is very similar to the one in open networks, although there are some differences. Consider a closed multiclass queueing network with N single server stations (nodes) and R different job classes. There are K customers always present in the closed network. We use the same notation as for open networks except that there are no external arrivals ($\lambda_{0r} = 0$) and the probability that a customer exits the network is equal to zero ($p_{r0} = 0$). We do not allow routing decisions, although routing decisions can be included in a manner similar to the case of open networks.

As in open networks, we only consider sequencing decisions and policies satisfying Assumption A(a); Assumption A(b) is automatically satisfied. We seek to maximize the weighted throughput

$$\sum_{r=1}^{R} c_r \lambda_r,$$

where $\lambda_r = \mu_r E[1\{B_r(\tau_k)\}]$ is the throughput of class r.

The procedure for obtaining the approximate achievable region is again similar to the proof of Theorem 4.2. We substitute $\lambda_{0r} = 0$ and $p_{r0} = 0$ in the main recursion to obtain constraints similar to (23), (24) and (25). As in Section 5.1, the application of the nonparametric method to $R(t)$ yields the traffic equations that are part of the characterization of the achievable region. In addition, for every class r we have the constraints $\sum_{j=1}^{R} I_{rj} = K\lambda_r/\mu_r$ and $\sum_{j \in C_r} (\lambda_j/\mu_j) \leq 1$, along with $\sum_{j=1}^{R} n_j = K$ and the positivity constraints of the variables involved.

5.3. *Other extensions.* Although we presented the method for systems with Poisson arrivals and exponential service time distributions, the method can be easily extended to systems with phase-type distributions by introducing additional variables. Moreover, one can use the method to derive bounds on the performance of particular policies. This is done by introducing additional constraints that capture as many features of a particular policy as possible.

6. Higher order interactions and nonlinear characterizations. The methodology we have developed so far leads to a *polyhedral* set that contains the achievable region and takes into account *pairwise* interactions among classes in the network. In this section, we extend the methodology and its power as follows:

1. We take into account *higher order interactions* among various classes by extending the potential function technique developed thus far.
2. We obtain *nonlinear* characterizations of the achievable region in a systematic way by using ideas from the powerful methodology of semidefinite programming.

In particular, we show how to construct a sequence of progressively more complicated nonlinear approximations (relaxations) that are progressively closer to the exact achievable space. We note that there are no examples of nonlinear characterizations of the achievable region in the literature with the exception of a simple example in Gelenbe and Mitrani (1980).

6.1. *Higher order interactions.* Our results so far have made use of the function $R(t) = \sum_{r=1}^{R} f(r)n_r(t)$ and were based essentially on the equation

$$E\big[E\big[R^2(\tau_{k+1})|\mathbf{n}(\tau_k)\big]\big] = E\big[R^2(\tau_k)\big].$$

By its nature, this method takes into account only pairwise interactions among the various classes. For example, the nonparametric method introduces variables $E[1\{B_r(\tau_k)\}n_j(\tau_k)]$, taking into account the interaction of classes r and j.

We now describe a generalization that aims at capturing higher order interactions. Consider again an open queueing network of the form described in Section 2, where there are no routing decisions to be made. We apply the nonparametric method by deriving an expression for $E[R^3(\tau_{k+1})|\mathbf{n}(\tau_k)]$ and then collecting terms. Alternatively, we can use test functions $g(\mathbf{n}(t)) = n_r(t)n_j(t)n_k(t)$. [We need to modify Assumption A(b) and assume that $E[n_r^3(t)] < \infty$.] In addition to the variables $I_{rj} = E[1\{B_r(\tau_k)\}n_j(\tau_k)]$, we introduce some new variables, namely,

$$H_{rjk} = E\big[1\{B_r(\tau_k)\}n_j(\tau_k)n_k(\tau_k)\big] \tag{31}$$

and

$$M_{jk} = E\big[n_j(\tau_k)n_k(\tau_k)\big].$$

The recursion for $E[R^3(\tau_{k+1})|\mathbf{n}(\tau_k)]$ leads to a set of linear constraints involving the variables $\{(n_r, I_{rj}, H_{rjk}, M_{jk})\}$.

The new variables we introduced take into account interactions among three customer classes and we expect that they lead to tighter constraints. Another advantage of this methodology is that we can now obtain lower bounds for more general objective functions involving the *variances* of the number of customers of class r, since the variables $M_{jj} = E[n_j^2(\tau_k)]$ are now in the augmented space.

Naturally, we can continue with this idea and apply the nonparametric method to $E[R^i(\tau_{k+1})|\mathbf{n}(\tau_k)]$ for $i \geq 4$. In this way, we take into account interactions among i classes in the system. There is an obvious trade-off between accuracy and tractability in this approach. If we denote by P_i the set obtained by applying the nonparametric method to $E[R^i(\tau_{k+1})|\mathbf{n}(\tau_k)]$, the approximate performance region that takes into account interactions of up to order i is $\bigcap_{l=1}^{i} P_l$. The dimension of this set and the number of constraints is $O(R^i)$, which even for moderate i can be prohibitively large.

The explicit derivation of the resulting constraints is conceptually very simple, but is algebraically involved and does not yield any additional insights. In fact, this derivation is not hard to automate: A symbolic manipula-

tion program, like Mathematica or Maple, can be used to write down the recursion for $E[R^i(\tau_{k+1})|\mathbf{n}(\tau_k)]$ and generate equality constraints by collecting the coefficients of each monomial in the f-parameters. We have indeed developed this software package, which automatically generates linear constraints and finds lower bounds using an LP solver.

6.2. *Nonlinear interactions.* In several examples (see Section 8), we found that although the method provides relatively tight bounds, it does not exactly characterize the achievable region and there is a gap between the lower bound and the performance of an optimal policy. We believe that the reason is that the achievable region is not always a polyhedron. We will therefore discuss how to extend our methods so as to allow the possibility of nonlinear characterizations.

Let $\mathbf{Y}$ be a vector of random variables and let Q be a symmetric positive semidefinite matrix. Clearly,

$$E\big[(\mathbf{Y} - E[\mathbf{Y}])^T Q(\mathbf{Y} - E[\mathbf{Y}])\big] \geq 0,$$

which implies that

$$E[\mathbf{Y}^T Q\mathbf{Y}] \geq E[\mathbf{Y}^T]QE[\mathbf{Y}], \tag{32}$$

which is Jensen's inequality applied to the convex function $x^T Qx$. Notice that (32) holds for every symmetric semidefinite matrix Q. By selecting particular values for matrices Q, one obtains a family of inequalities. For example, consider the model of Section 6.1 and fix some r. Let $\mathbf{Y}$ be the vector with components $1\{B_r(\tau_k)\}n_j(\tau_k)$, $j = 1, \ldots, R$, and use the identity $1\{B_r(\tau_k)\} = (1\{B_r(\tau_k)\})^2$ to obtain the quadratic inequalities

$$\sum_{i,j} H_{rij}Q_{ij} \geq \sum_{i,j} Q_{ij}I_{ri}I_{rj}, \qquad r = 1, \ldots, R, \tag{33}$$

where I_{ri} and H_{rij} have been defined in (21) and (31).

Any choice of Q leads to a new set of quadratic inequalities. We will actually impose the constraints of the form (33) for all choices of Q. Let Z be the polyhedron obtained by using the ideas in Section 6.1. We then obtain a lower bound by solving the following optimization problem, which we call P_{NLP}:

$$\begin{aligned} &\text{minimize } \sum_{r=1}^{R} c_r x_r \\ &\text{subject to } (x_r, I_{rj}, H_{rjk}, M_{jk}) \in Z, \\ &\sum_{i,j} H_{rij}Q_{ij} \geq \sum_{i,j} Q_{ij}I_{ri}I_{rj}, \qquad r = 1, \ldots, R, \forall\, Q \geq 0. \end{aligned} \tag{34}$$

Although this lower bound involves an optimization problem with infinitely many constraints, it can be efficiently solve, as we now explain. We first note that for any fixed positive semidefinite matrix $Q \geq 0$, the constraint (34) is convex in the variables I_{ri} and H_{rij}. Let us start by imposing the constraint

(34) only for Q equal to the identity matrix and solve the resulting optimization problem, which is a relaxation of P_{NLP}. We then check whether the find solution violates any of the constraints (34) for some $Q \geq 0$. This can be done by solving the following separation problem:

SEPARATION PROBLEM. Given some $(n_r, I_{rj}, H_{rjk}, M_{jk}) \in Z$, minimize, for each r, the objective function

$$\sum_{i,j} H_{rij} Q_{ij} - \sum_{i,j} Q_{ij} I_{ri} I_{rj}$$

over all positive semidefinite matrices Q.

If the optimal value in the separation problem is nonnegative for every r, then the current vector $(n_r, I_{rj}, H_{rjk}, M_{jk})$ satisfies all constraints of the form (34) and is an optimal solution for problem P_{NLP}. If not, then a semidefinite matrix Q has been found for which the corresponding constraint is violated by the current vector. We can then add this constraint explicitly, solve the resulting relaxation of P_{NLP} and continue similarly.

We note that the separation problem is a semidefinite programming problem that can be solved efficiently by simplex type or interior point methods [see Alizadeh (1992)]. The overall algorithm would run in polynomial time if we use the ellipsoid algorithm or a variant like Vaidya's algorithm to solve the relaxations of P_{NLP} and interior point methods for the semidefinite programming problem.

Higher order nonlinear constraints also can be obtained by using inequalities such as

$$E\left[1\{B_r(\tau_k)\} n_j^h(\tau_k)\right] \geq E\left[1\{B_r(\tau_k)\} n_j(\tau_k)\right]^h, \qquad h = 1, 2, \ldots,$$

which again follow from Jensen's inequality. We thus obtain a sequence of progressively more complicated convex sets that approximate the achievable region.

7. Single station networks: Complete characterization. In this section, we demonstrate that our characterizations of the achievable region are exact in the case of single station systems. More specifically, the contributions of this section are the following:

1. We consider a multiclass M/M/1 queue with Bernoulli feedback [Klimov's (1974) problem], where each class can have distinct service requirements under work-conserving *preemptive* policies. We show that our parametric method leads to a complete and explicit characterization. This result is similar to the results of Tsoucas (1991), who derives the form of the achievable region in Klimov's problem under *non-preemptive* policies and general service requirements. In contrast to the characterization in Tsoucas (1991), all of the parameters in our characterization are given in closed form. Bertsimas and Niño-Mora (1992) also derive the performance space in this case using different methods.

2. Using the nonparametric method, we obtain an exact characterization of the achievable region for single station systems that involves only a polynomial number of variables and constraints. This is, in our opinion, interesting not only from a probabilistic, but also from a combinatorial point of view. The parametric method gives rise to a polyhedron (an extended polymatroid) in R variables and $2^R - 1$ constraints for which the optimization problem can be solved by means of a one-pass polynomial time algorithm. It has been conjectured that whenever problems of this type are solvable in polynomial time, they have LP formulations with a polynomial number of variables and constraints. The nonparametric method verifies this conjecture for this special case.
3. In multiclass queues without feedback and homogeneous open networks, the performance space was derived in Gelenbe and Mitrani (1980) and Ross and Yao (1987), respectively, using conservation laws. Since our parametric method is exact in these cases, we see that our parametric method is as powerful as conservation laws. Moreover, as the method generalizes to arbitrary networks, the parametric method can be viewed as generalization of conservation laws to queueing networks.

We next introduce conservation laws and their connections with polyhedral performance regions. For a more comprehensive discussion, see Bertsimas and Niño-Mora (1992) and Shanthikumar and Yao (1992).

7.1. *Strong conservation laws.* Consider a multiclass queueing network of the type described in Section 2. Let $E = \{1, \ldots, R\}$ be the set of all classes and let 2^E be the set of all subets of E. Let $\mathscr{U}$ be the set of all work-conserving, possibly preemptive, Markovian policies that satisfy Assumption A. We say that a policy is a priority rule if it assigns priorities to jobs according to some particular order. Finally, we say that a priority rule gives preemptive priority to a subset S of the set of classes if all classes in S have higher priority than the classes outside S. For any policy $u \in \mathscr{U}$ and any class i, we use n_i^u to denote the mean number of class i customers in the system under that policy.

DEFINITION 7.1 (Strong conservation laws). The vector $\mathbf{n}$ is said to satisfy *strong conservation laws* if there exists a function b: $2^E \to \Re_+$ such that $b(\varnothing) = 0$ and a set of coefficients $f = (f_S(i))_{i \in E, S \subseteq E}$ satisfying

$$f_S(i) > 0, \quad \text{for } i \in S \quad \text{and} \quad f_S(i) = 0, \quad \text{for } i \notin S, \tag{35}$$

such that:

(a) for any $S \subseteq E$ and any priority rule π that gives priority to S,

$$\sum_{i \in S} f_S(i) n_i^\pi = b(S); \tag{36}$$

(b) for any policy $u \in \mathscr{U}$,

$$\sum_{i \in S} f_S(i) n_i^u \geq b(S) \quad \text{for all } S \subset E \quad \text{and} \quad \sum_{i \in E} f_E(i) n_i^u = b(E). \tag{37}$$

Consider now the inequalities in Theorem 4.1. If we can show that for each set S of classes, the corresponding inequality holds with equality when priority is given to classes in S, then the system satisfies strong conservation laws.

Our interest in strong conservation laws stems from the following result.

THEOREM 7.1 [Bertsimas and Niño-Mora (1992)]. *Assume that the vector* $\mathbf{n}$ *satisfies strong conservation laws. Define the polyhedron* $\mathscr{B}(f, b)$ *as the set of all vectors that satisfy the linear constraints in* (37). *Then*:

(a) *The vertices of* $\mathscr{B}(f, b)$ *are the performance vectors* n^{π} *of priority rules.*

(b) $\mathscr{B}(f, b)$ *is the achievable region when we restrict to work-conserving policies.*

When strong conservtion laws are satisfied, the achievable region $\mathscr{B}(f, b)$ is an *extended polymatroid* [see Bertsimas and Niño-Mora (1992) and Bhattacharya, Georgiadis and Tsoucas (1992) for the definition]. The fundamental structural property of an extended polymatroid is that minimizing a linear function $\sum_{i \in E} c_i n_i$ over $\mathscr{B}(f, b)$ can be achieved by a one-pass (in particular, polynomial time) algorithm [see Bertsimas and Niño-Mora (1992)]. In addition, the vertices of the extended polymatroid can be very easily calculated by solving a triangular system of equations and therefore, according to Theorem 7.1, the performance vectors of the absolute priority rules are readily available.

7.2. *Klimov's problem.* Consider the single-server station of Figure 2. Customers of class $i \in E = \{1, 2, \ldots, R\}$ arrive in the system according to independent Poisson processes with rate λ_{0i} and have an exponentially distributed service time with mean $1/\mu_i$. Upon service completion, a class i customer is fed back into the system as a class j customer with probability p_{ij}, while with probability p_{i0} it leaves the system. Let n_i be the expected number of customers of class i in steady state.

The server is using a preemptive, work-conserving discipline satisfying Assumption A. We show that for this problem the polyhedron obtained from our method in Section 4 is equal to the achievable region.

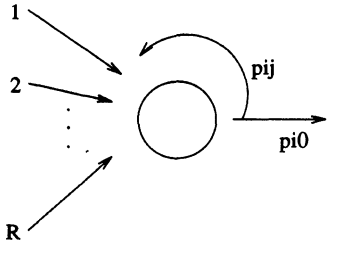

FIG. 2. *Klimov's problem.*

The traffic equations for the foregoing system are

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{R} \lambda_j p_{ji}. \tag{38}$$

We assume that $\lambda_i < \mu_i$ for all i.

Our characterization is as follows:

THEOREM 7.2. *The achievable region for the Klimov problem under work-conserving policies satisfying Assumption A is the polyhedron P defined by the constraints*

$$\sum_{i \in S} f_S(i) n_i \geq \frac{N'(S)}{D'(S)} \qquad \forall\, S \subset E, \tag{39}$$

$$\sum_{i \in E} f_S(i) n_i = \frac{N'(E)}{D'(E)}, \qquad n_i \geq 0, \qquad i = 1, \ldots, R, \tag{40}$$

where

$$N'(S) = \sum_{i \in S} \lambda_{0i} f_S^2(i) + \sum_{i \in S} \lambda_i \left[\sum_{j \in S} p_{ij} (f_S(i) - f_S(j))^2 + \sum_{j \notin S} p_{ij} f_S^2(i) \right] + \sum_{i \notin S} \lambda_i \sum_{j \in S} p_{ij} f_S^2(j),$$

$$D'(S) = 2\left[1 - \sum_{i \in S} \lambda_{0i} f_S(i)\right]$$

and where the coefficients $f_S(i)$ satisfy the system of equations

$$f_S(i) = \frac{1}{\mu_i} + \sum_{j \in S} p_{ij} f_S(j). \tag{41}$$

PROOF. We apply Theorem 4.1 with the f-parameters chosen as in (41). This shows that the inequalities (39) are necessary and the polyhedron P contains the achievable region.

We will show next that the vector $\mathbf{n}$ satisfies strong conservation laws. We note that in the proof of Theorem 4.1 we used the fact

$$R^S(\tau_k) 1\{\text{server busy from some class } i \in S \text{ at } \tau_k\} \leq R^S(\tau_k). \tag{42}$$

The preceding inequality holds with equality when preemptive priority is given to the classes $i \in S$. To see this, observe that when $R^S(\tau_k) \neq 0$ (that is, if a customer of some class $i \in S$ is present) and preemptive priority is given to the classes $i \in S$, then the server should be working on a customer of some class in S. Otherwise, when $R(\tau_k) = 0$, (42) holds with equality, trivially. In addition, we also used in the proof of Theorem 4.1 the fact that

$$R^S(\tau_k) 1\{B_r(\tau_k)\} \geq 0, \qquad r \notin S.$$

This also holds with equality when preemptive priority is given to the classes $i \in S$. Hence the rhs of (39) is achieved by this specific priority policy. Moreover, since we have restricted ourselves to work-conserving policies, when $S \equiv E$, (40) holds.

Therefore, the performance vector $\mathbf{n}$ satisfies strong conservation laws. Applying Theorem 7.1, we conclude that P is equal to the achievable region. □

REMARK 1. Having derived the achievable region, the optimal policy and the optimal performance vector can be found by applying a one-pass algorithm [see Bertsimas and Niño-Mora (1992)]. In particular, although the characterization of the achievable region has an exponential number of inequalities, this computation runs in polynomial time (in fact, linear time).

REMARK 2. If we apply the nonparametric method (Theorem 4.2), we find an alternative polyhedral characterization of the achievable region, which we call Q, that has $O(R^2)$ variables and constraints in the enlarged space of $\{(n_i, I_{ij}, N_{ij})\}$. Due to Theorem 4.3 and the fact that P is the exact achievable region, Q is also an exact characterization of the achievable region.

REMARK 3. If we specialize Theorem 7.2 to a multiclass queue without feedback, we find exactly the performance space first derived in Gelenbe and Mitrani (1980). Once more, Theorem 4.2 leads to an alternative polynomial characterization. Finally, our approach can be shown to yield the exact achievable region for homogeneous networks [Ross and Yao (1987)].

8. Numerical results. In this section, we provide some numerical results in order to evaluate the performance of our bounding techniques for open and closed networks with sequencing or routing control.

8.1. *Sequencing for open networks.* In this subsection we provide two network examples where sequencing decisions are involved. For each of these examples and for various traffic conditions, we calculate the following items:

1. The lower bound on achievable performance obtained from the parametric method of Section 4.1.
2. The lower bound on achievable performance obtained from the nonparametric method of Section 4.2.
3. The performance of the FCFS policy.
4. The performance of the best policy we were able to find, serves as an upper bound.

Since the optimal cost is not known, we cannot calculate the closeness of our lower bound to the optimal cost. Instead, we will calculate its closeness to the upper bound, which is, of course, an overestimate. In particular, we will

calculate the *efficiency* of the bound, which we define as

$$\text{efficiency} \equiv \frac{\text{best lower bound}}{\text{best upper bound}} 100\%.$$

8.1.1. *The simple two-station network revisited.* Consider the two-station network example studied in Section 3 and depicted in Figure 1. Table 1 compares our lower bounds on attainable performance with FCFS and the following threshold policy:

> Let B be some constant. Give priority to type 1 customers at station 1 when there are B or fewer customers at station 2. Otherwise give priority to type 2 customers. Never idle.

This policy was proposed in Harrison and Wein (1989), where the Brownian network model approach was used.

"Lower bound 1" and "Lower bound 2" in the table correspond to the bounds developed in Sections 4.1 and Section 4.2, respectively. The objective function is the total expected number of customers in the network, that is, $c_1 = c_3 = \lambda_1$, $c_2 = \lambda_2$. Note that the performance reported in the table for the threshold policy corresponds to the optimal value of the threshold B, which was found for each case by doing several simulation runs. Table 2 contains the data used for each case reported in Table 1, and ρ_A and ρ_B are the total traffic intensities at stations 1 and 2, respectively.

It is interesting that the efficiency of our lower bound is comparable to the efficiency of the "pathwise bound" derived in Ou and Wein (1992), which is based on simulation. Note also that the threshold policy clearly outperforms FCFS. From Table 1 it is apparent that as $\rho \to 1$, the efficiency of the bound increases for both balanced and imbalanced traffic conditions. We believe that this behavior is mainly due to the fact that the threshold policy behaves better as the traffic gets heavier [see Harrison and Wein (1989)]. Moreover, the efficiency of the bounds is better in imbalanced traffic conditions.

TABLE 1
Numerical results for the network of Figure 1

Load node 1–node 2	Lower bound 1	Lower bound 2	FCFS	Thresh. policy	Efficiency (%)
Heavy–heavy	14.15	14.15	19.43	16.98	83
Heavier–heavier	19.9	19.9	28	23.76	84
Very heavy–very heavy	49.96	49.96	73	57.38	87
Medium–heavy	9.99	9.99	10.5	10.44	96
Light–medium	2.04	2.04	2.17	2.16	94
Heavy–medium	9.6	9.6	10.5	9.98	96
Medium–light	1.9	1.9	2.17	2.14	89

TABLE 2
Data for the experiments of Table 1

Load	ρ_A	ρ_B	λ_1	λ_2	μ_1	μ_2
Heavy–heavy	0.93	0.86	0.86	1	2	1
Heavier–heavier	0.95	0.90	0.90	1	2	1
Very heavy–very heavy	0.98	0.96	0.96	1	2	1
Medium–heavy	0.6	0.9	0.9	0.3	2	1
Light–medium	0.4	0.6	0.6	0.2	2	1
Heavy–medium	0.9	0.6	0.6	1.2	2	1
Medium–light	0.6	0.4	0.4	0.8	2	1

8.1.2. *A six-class network example.* Consider the network depicted in Figure 3. Customers of type 1 enter the network in a Poisson stream of rate λ_1 and they visit stations 1, 2, 1, 2 in that order, before exiting the network, forming classes 1, 2, 3, 4, respectively. Customers of type 2 enter the network in a Poisson stream of rate λ_2 and they visit stations 1, 2 before exiting the network, forming classes 5, 6, respectively. The single servers at stations 1 and 2 have service times exponentially distributed with rates μ_1 and μ_2, respectively.

Table 3 compares our lower bounds on attainable performance with FCFS and the best policy found for various load conditions, and evaluates the efficiency of the bound. "Lower bound 1" and "Lower bound 2" in the table correspond to the bounds developed in Sections 4.1 and 4.2, respectively. The costs for all of the experiments reported in the table were chosen to be $c_1 = 1.5$, $c_2 = 1.3$, $c_3 = 1.2$, $c_4 = 1$, $c_5 = 1.1$ and $c_6 = 1.1$.

For each load condition that we considered, the best policy we were able to find was a strict priority rule, not necessarily the same rule for different cases. (We only considered non-preemptive policies.) Table 4 contains the data used for each case reported in Table 3. Once more, ρ_A and ρ_B denote the total traffic intensities at stations 1 and 2, respectively.

8.2. *Routing.* In this subsection we treat the network of Figure 4. Customers arrive according to a Poisson process with rate λ and have to be routed to one of two service stations where they wait to be served. Service

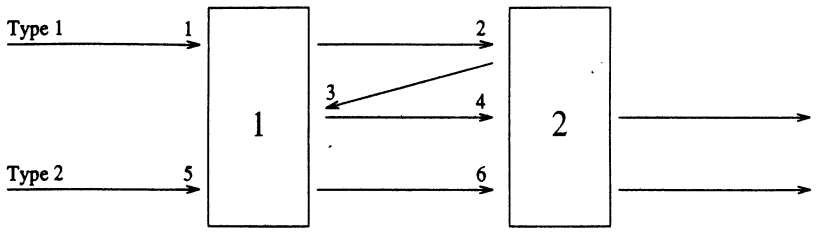

FIG. 3. *A six-class network example.*

TABLE 3
Numerical results for the network of Figure 3

Load node 1–node 2	Lower bound 1	Lower bound 2	FCFS	Best policy	Efficiency (%)
Heavy–heavy	15.72	16.67	30.56	26.89	62
Medium–medium	5.83	6.17	9.86	9.25	67
Medium–heavy	15.77	15.85	21.26	18.20	87
Heavy–medium	18.77	18.79	23.00	19.80	95

times are exponentially distributed with rate μ at either station. We are interested in a routing policy that minimizes the expected number of customers in the system. Let n_r, $r = 1, 2$, be the steady-state mean of the number of customers in stations 1 and 2, respectively.

We use the extension of the nonparametric method discussed in Section 5.1 and obtain the following set of equations, which are true for work-conserving policies satisfying Assumption A:

$$\begin{aligned} \lambda p_1 + \lambda K_{11} &= \mu_1 n_1, \\ \lambda(1 - p_1) + \lambda(n_2 - K_{12}) &= \mu_2 n_2, \\ \lambda K_{12} + \lambda(n_1 - K_{11}) &= \mu_1 I_{12} + \mu_2 I_{21}, \\ I_{12} \le n_2, \quad I_{21} \le n_1, \quad K_{11} \le n_1, \quad K_{12} \le n_2, \quad p_1 &\le 1. \end{aligned}$$

Besides n_1 and n_2, the variables in this characterization are as follows: $p_1 = E[\, p_1(\tau_n)]$ is the steady-state probability of routing customers to the first station; I_{ij}'s have been defined in (21) and $K_{ij} = E[\, p_i(\tau_n) n_j(\tau_n)]$ take into account the interaction between the routing decision and the number of customers at station j.

A lower bound is obtained by minimizing $n_1 + n_2$ subject to the preceding constraints along with the nonnegativity constraints of the variables involved. If $\lambda \ge \mu$ (i.e, for medium to heavy load), this minimization can be carried out analytically and the lower bound on the achievable performance is

$$z_{LB} = \frac{\lambda}{2\mu - \lambda}. \tag{43}$$

TABLE 4
Data for the experiments of Table 3

Load	ρ_A	ρ_B	λ_1	λ_2	μ_1	μ_2
Heavy–heavy	0.85	0.90	0.5	0.7	2	1.89
Medium–medium	0.7	0.7	0.5	0.7	2.43	2.43
Medium–heavy	0.6	0.9	0.5	0.7	2.83	1.89
Heavy–medium	0.9	0.6	0.5	0.7	1.89	2.83

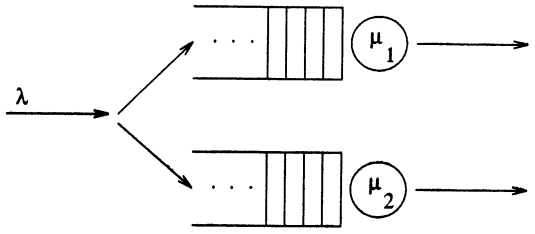

FIG. 4.

For this problem, the policy SQ that lets each arriving customer join the shortest queue is known to be optimal [Walrand (1988)]. In addition, Foschini and Salz (1978) have shown that this policy asymptotically achieves (43) as $\rho = \lambda/(2\mu \to 1)$. Hence, our lower bound is asymptotically exact. Table 5 compares our lower bound with the performance of the shortest queue policy (SQ) and provides its efficiency. In Table 6 we report the data used for the experiments of Table 5.

8.3. *Closed networks.* In this subsection we treat a closed network. Consider the network of Figure 3. We require a 50–50 product mix for types 1 and 2. Thus, customers of class 4, upon exiting station 2, are equally likely to become customers of class 1 and of class 5. Similarly, customers of class 6, upon exiting station 2, are equally likely to become customers of class 1 and of class 5. Service times are exponentially distributed with means 8, 5, 2, 7, 4, 1 for classes 1 to 6, respectively. Due to the 50–50 product mix, $\lambda \equiv \lambda_1 = \lambda_2$. The objective is to maximize the total throughput 2λ.

We use the extension of the nonparametric method proposed in Section 5.2 to derive a polyhedral approximate performance region for this problem under work-conserving policies satisfying Assumption A(a). Maximizing the objective function over this region, we obtain an upper bound on the optimal throughput. This network was treated in Harrison and Wein (1990), where a particular strict priority policy was proposed based on heavy-traffic considerations. Table 7 compares our upper bound with the policy of Harrison and Wein (1990) and calculates the efficiency of the bound.

TABLE 5
Numerical results for the routing example

Load	Lower bound	SQ policy	Efficiency (%)
Heavy	9.00	9.85	92
Medium	2.33	2.95	79
Light	1.22	1.69	72

TABLE 6
Data for the experiments of Table 5

Load	ρ	λ	μ
Heavy	0.9	2.7	1.5
Medium	0.7	2.1	1.5
Light	0.55	1.65	1.5

8.4. *Summary.* Our computational results suggest the following conclusions:

1. The lower bound obtained by the nonparametric variation of the method is at least as good as the lower bound obtained by the parametric method as expected from Theorem 4.3. In the more complicated example with six classes, it was strictly better. The reason is that the nonparametric method better takes into account the interactions among various classes.
2. The efficiency of our lower bounds is comparable to the efficiency of the "pathwise bound" derived in Ou and Wein (1992).
3. The bounds are very efficient in imbalanced traffic conditions and the efficiency of the bounds increases with the traffic intensity. A plausible explanation is that in imbalanced conditions the behavior of the system is dominated by a single bottleneck station, and for single station systems we know our bounds to be exact.
4. In balanced traffic conditions, the bounds also behave well, especially when the traffic intensity is not very close to 1. However, even for heavy-balanced traffic conditions and for the examples that we studied, the efficiency did not get worse than 62%.
5. In the routing example, the method is asymptotically exact, which is very encouraging.
6. In the closed network example (as well as other examples that we ran), the bounds were extremely tight.

9. Reflections. In this paper we proposed new techniques for describing the region of achievable performance for multiclass open and closed queueing networks, with Poisson arrivals (in open networks) and exponentially dis-

TABLE 7
Numerical results for the closed network example

Population	Upper bound	Heavy-traffic policy	Efficiency (%)
7	0.131	0.127 ± 0.0009	97
10	0.135	0.133 ± 0.0009	99
20	0.139	0.138 ± 0.0009	99

tributed service times. Our techniques use linear and nonlinear potential function methods. We introduced an arbitrary potential function that gives a family of bounds (linear and nonlinear) that take into account high order interactions of various classes. We also introduced the idea of choosing the best possible potential function to obtain the tightest possible bounds by allowing the flexibility of unknown coefficients.

We believe that the power of the method stems from the fact that it takes into account higher order interactions among various classes. Our first order method is as powerful as conservation laws since it leads to exact characterizations (single station network, homogeneous networks). As such, this approach can be seen as the natural extension of conservation laws. It is desirable to check the tightness of the various bounds derived in the paper in actual applications. The numerical results we report are encouraging, but certainly more work is needed, especially to illustrate the power of the higher order formulations.

REFERENCES

ALIZADEH, F. (1992). Combinatorial optimization with semi-definite matrices. In *Proceedings of 2nd Conference in Integer Programming* 385–405. Carnegie Mellon Univ.

BERTSIMAS, D., NIÑO-MORA, J. (1992). Conservation laws, extended polymatroids and the multi-armed bandit problem: a unified polyhedral approach. Working paper, Operations Research Center, MIT.

BERTSIMAS, D., PASCHALIDIS, I. CH. and TSITSIKLIS, J. N. (1992a). Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance. Technical Report LIDS-P-2155, Laboratory for Information and Decision Systems, MIT.

BERTSIMAS, D., PASCHALIDIS, I. CH. and TSITSIKLIS, J. N. (1992b). Scheduling of multiclass queueing networks: Bounds on achievable performance. In *Proceedings of Workshop on Hierarchical Control for Real Time Scheduling of Manufacturing Systems, Lincoln, New Hampshire, October 16–18*. Extended abstract.

BHATTACHARYA, P. P., GEORGIADIS, L. and TSOUCAS, P. (1992). Extended polymatroids: Properties and optimization. Research Report, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY.

CHEN, H., YANG, P. and YAO, D. D. (1991). Control and scheduling in a two-station queueing network: Optimal policies and heuristics. Preprint.

FEDERGRUEN, A. and GROENEVELT, H. (1988). Characterization and optimization of achievable performance in queueing systems. *Oper. Res.* **36** 733–741.

FOSCHINI, G. J. and SALZ, J. (1978). A basic dynamic routing problem and diffusion. *IEEE Trans. Commun.* **26** 320–327.

GELENBE, E. and MITRANI, L. (1980). *Analysis and Synthesis of Computer Systems*. Academic, London.

GITTINS, J. C. (1989). *Bandit Processes and Dynamic Allocation Indices*. Wiley, New York.

HARRISON, J. M. (1986). Brownian models of queueing networks with heterogeneous customers. Presented at the IMA Workshop on Stochastic Differential Systems.

HARRISON, J. M. and WEIN, L. M. (1989). Scheduling networks of queues: Heavy traffic analysis of a simple open network. *Queueing Systems Theory Appl.* **5** 265–280.

HARRISON, J. M. and WEIN, L. M. (1990). Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Oper. Res.* **38** 1052–1064.

KELLY, F. P. and LAWS, C. N. (1992). Dynamic routing in open queueing networks. Preprint.

KLEINROCK, L. (1976). *Queueing Systems* **2**: *Computer Applications*. Wiley, New York.

KLIMOV, G. P. (1974). Time-sharing service systems I. *Theory Probab. Appl.* **19**.

KUMAR, P. R. (1992). Re-entrant lines. In *Proceedings of Workshop on Hierarchical Control for Real Time Scheduling of Manufacturing Systems, Lincoln, New Hampshire, October 16–18.*

KUMAR, S. and KUMAR, P. R. (1993). Performance bounds for queueing networks and scheduling policies. Preprint.

LOVASZ, L. and SCHRIJVER, A. (1991). Cones of matrices and set functions, and 0-1 optimization. *SIAM J. Optim.*, **1**, 166–190.

MEYN, S. P. and DOWN, D. (1994). Stability of generalized Jackson networks. *Ann. Appl. Probab.* **4** 124–148.

OU, J. and WEIN, L. M. (1992). Performance bounds for scheduling queueing networks. *Ann. Appl. Probab.* **2** 460–480.

PASCHALIDIS, I. CH. (1992). Scheduling of multiclass queueing networks: Bounds on achievable performance. M.S. thesis, MIT.

ROSS, K. and YAO, D. (1987). Optimal dynamic scheduling in Jackson networks. Preprint.

SHANTHIKUMAR, J. G. and YAO, D. D. (1992). Multiclass queueing systems: Polymatroid structure and optimal scheduling control. *Oper. Res.* **40** 293–299.

TSOUCAS, P. (1991). The region of achievable performance in a model of Klimov. Research Report, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY.

WALRAND, J. (1988). *An Introduction to Queueing Networks.* Prentice-Hall, Englewood Cliffs, NJ.

WEIN, L. M. (1990a). Optimal control of a two-station brownian network. *Math. Oper. Res.* **15** 215–242.

WEIN, L. M. (1990b). Scheduling networks of queues; heavy traffic analysis of a two station network with controllable inputs. Preprint.

WEISS, G. (1988). Branching bandit processes. *Probab. Engng. Inform. Sci.* **2** 269–278.

SCHOOL OF MANAGEMENT
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE, ROOM E53-359
CAMBRIDGE, MASSACHUSETTS 02139

LABORATORY FOR INFORMATION AND
DECISION SCIENCES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139-4307