Optimization of Multiclass Queuing Networks: Polyhedral and Nonlinear Characterizations of Achievable Performance

Dimitris Bertsimas, Ioannis Ch. Paschalidis
and
John N. Tsitsiklis

WP# -3509-92 MSA December, 1992

Optimization of Multiclass Queueing Networks: Polyhedral and Nonlinear Characterizations of Achievable Performance [1]

Dimitris Bertsimas Ioannis Ch. Paschalidis
John N. Tsitsiklis

Laboratory for Information and Decision Systems
and
Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139

December 1992

[1] A preliminary version of this paper [BPT1] appeared in the proceedings of the "Workshop on hierarchical control for real-time scheduling of manufacturing systems", Lincoln, New Hampshire, October 16-18, 1992. The full paper was presented in the ORSA/TIMS conference [BPT2] on November 2, 1992.
Research supported by the National Science Foundation under Grant ECS-8552419, by a Presidential Young Investigator award DDM-9158118 with matching funds from Draper Laboratory, by the Leaders for Manufacturing Program at MIT, and by the ARO under grand DAAL03-92-G0309.

Abstract

We consider open and closed multiclass queueing networks with Poisson arrivals (in open networks), exponentially distributed class dependent service times, and with class dependent deterministic or probabilistic routing. For open networks, 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 variants of a method to derive polyhedral and nonlinear spaces which contain the entire set of achievable response times under stable and preemptive scheduling policies. By optimizing over these spaces, we obtain lower bounds on achievable performance. In particular, we obtain a sequence of progressively more complicated nonlinear approximations (relaxations) which are progressively closer to the exact achievable space. In the special case of single station networks (multiclass queues and Klimov's model) and homogenous multiclass networks, our characterization gives exactly the achievable region. Consequently, the proposed method can be viewed as the natural extension of conservation laws to multiclass queueing networks. For closed networks, the performance objective is to maximize throughput. We similarly find polyhedral and nonlinear spaces that include the performance space and by maximizing over these spaces we obtain an upper bound on the optimal throughput.

We check the tightness of our bounds by simulating heuristic scheduling policies for simple open networks 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 both the number of variables and constraints being polynomial (quadratic) in the number of classes in the network. The i-th order approximation involves solving a convex programming problem in dimension $O(R^{i+1})$, where R is the number of classes in the network, which can be solved efficiently using techniques from semi-definite programming.

1 Introduction

A *multiclass queueing network* is one that services multiple types of customers which may differ in their arrival processes, service requirements, routes through the network as well as costs per unit of waiting time. The fundamental optimization problem that arises in open networks is to determine an optimal policy for sequencing and routing customers in the network in order to minimize 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 the 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 which route each type of customer should follow to get through the network. In this paper we consider optimization problems involving both routing and sequencing decisions.

There are several important applications of the described problems: Packet-switching communication networks with different types of packets and different priorities between these packet-types, job shop manufacturing systems, scheduling control of a multi-processor and multi-programmed computer system, 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 in the case of Poisson arrivals, and *class dependent* exponential service times, the simplest possible policy, FCFS, does not lead to product form solutions and it is not known how to analyze FCFS analytically. 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. But even if simulation is used for a proposed heuristic policy, it is not clear 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 approximately within the same order of magnitude as "pathwise" bounds derived in [OuWe] with a technique that needs a simulation experiment for the calculation of the bound. 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 i-th 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 waiting 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 programming problem. This has serious advantages because one can use advanced algorithmic methods from a mature field, and also consider more general objective functions (in particular involving variances). With respect to this objective, we obtain a sequence of progressively more complicated nonlinear approximations (relaxations) which are progressively closer to the exact achievable space. We note, that except for a simple example in [GeMi], we do not know of any other example of a nonlinear characterization. In the first order approximation, where most of the emphasis is placed for tractability purposes, we find two polyhedra that contain the achievable region of expected waiting times for the different classes in open and closed multiclass networks.

In the case of simpler systems (a multiclass queue [GeMi, Klv2], a single server network [Klim, Tsou] and a homogeneous open network [RoYa]) our first order characterization is exact, i.e., it is identical to the characterization in [GeMi] and [RoYa] for the multiclass queue and homogeneous network respectively, and consistent with the characterization of Tsoucas [Tsou] derived using conservation laws. In all of these cases we also find a reformulation of the achievable space 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. Obviously, optimizing over these spaces we obtain bounds to the optimal value and in the case where the characterization is exact we find the exact value as well as the optimal policy.

The third methodological contribution of the 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 1960s has used potential function methods to prove stability. Regarding the use of potential function methods to bound performance in queueing systems, Kumar [Kuma] uses a method of Meyn and Down [MeDo] (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 (re-entrant 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 introduce the idea of choosing the best possible potential function to obtain the tightest possible bounds by allowing the flexibility of unknown coefficients. We also propose an algebraic way based on manipulation of multivariable polynomials for automatically deriving the constraints of the approximating spaces. One could imagine that this *automatic generation* could be combined with an algorithm that finds lower bounds on the achievable performance by progressively adding constraints to the problem. This is exactly how large scale combinatorial problems are solved to optimality using polyhedral methods.

The fourth methodological contribution of the paper is a separate general technique to generate nonlinear (convex) constraints. We show that optimization over this set of constraints can be achieved by cutting plane methods very efficiently (in polynomial time) using techniques from semi-definite programming. Our ideas are influenced by the recent developments in deriving lower bounds for integer programming problems using semi-definite programming (Lovasz and Schrijver [LoSc], Alizadeh [Al]).

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 [GeMi] first showed using conservation laws that the performance region of a multiclass queue can be described as a polyhedron. Federgruen and Groonvelt [FeGr] advanced the theory further 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). Shantikumar and Yao [ShYa] generalized the theory further by observing that if a system satisfies conservation laws, then the underlying performance space is necessarily a *polymatroid polytope*. They also prove that 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 [RoYa]). Tsoucas ([Tsou]) derives the achievable region for scheduling a multiclass non-preemptive M/G/1 queue with Bernoulli feedback introduced by Klimov ([Klim]). Finally, Bertsimas and Niño-Mora [BeNi] generalize the idea of conservation laws and show that for all systems that satisfy these generalized conservation laws, their underlying performance space is a polyhedron with very strong structural properties, called an *extended polymatroid* in [BGT]. Optimization of a linear function over extended polymatroids can be achieved by an adaptive greedy algorithm (see [BGT] and [BeNi]). The framework of [BeNi] includes all the cases we mentioned before, as well as the multi-armed bandit problem (Gittins [Gi]), branching bandits (Weiss [We]) and deterministic scheduling problems. In this way Klimov's algorithm and Gittins indices for the multi-armed bandit problem are special cases of the adaptive greedy algorithm for optimizing a linear function over an extended polymatroid.

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 ([Ha]) and further explored by Wein, this approach proposes heuristic policies which typically outperform more traditional ones. This approach has been more successful in closed networks ([HaWe2]) and networks with controllable input ([Wei1], [Wei2]), but has not been as successful in scheduling open networks. In particular, Harrison and Wein show in [HaWe1] that a threshold policy is consistent with the optimality conditions for a

Brownian two-station, three-class network which we also consider in this paper (Section 3). Wein [Wei1, Wei2] 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 [KeLa]. For a thorough survey of the rather vast literature on routing in stochastic systems see Walrand [Wa].

In the only study that concerns lower bounds for general networks, Ou and Wein [OuWe] 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 et al. [ChYY] follow a *stochastic intensity control* approach for the specific network topology studied in [HaWe1], 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 as well as the class of policies that we are 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 apply our methodology to obtain bounds for multiclass networks involving both routing and sequencing decisions. In Section 6, we extend one of the methods of Section 4 to closed networks. In Section 7, we explain how the methodology can be extended to derive tighter nonlinear approximations of the achievable region and to take into account higher order interactions. As an example, we derive the second order approximation (a

nonlinear characterization) of a general multiclass open network. We also describe how ideas from semi-definite programming can be used to tighten nonlinear approximations to the achievable region. In Section 8, we prove that we can get the exact characterization for an M/M/1 multiclass queue, for Klimov's problem with Poisson arrivals and exponentially distributed service times and for homogeneous networks. In Section 9, we apply our first order methods to three specific network examples considered in the literature and report numerical results. Finally, in Section 10, 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.

In this section, as well as in Sections 3 and 4, we will consider an open multiclass queueing network involving only sequencing decisions (the 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 customers 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*. Note that for the time being only sequencing decisions are involved; the routing probabilities p_{rs} are given. Let

$n_r(t)$ be the number of class r customers present in the network at time t. The vector $\vec{n}(t) = (n_1(t), \ldots, n_R(t))$ 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 policies satisfying the following assumption:

Assumption A *a) The Markov chain $\vec{n}(t)$ 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 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 *region of achievable performance*, that is, the set of all vectors $(x_1, \ldots, x_R)$ that are obtained by considering different policies. By minimizing the cost function $\sum_{r=1}^{R} c_r x_r$, over this region, we can then obtain the cost of an optimal policy. 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 customers* to be type 1 customers at station 1, *class 2 customers* to be type 2 customers at station 1 and *class 3 customers* to be type 1 customers at station 2. Let λ_1 and λ_2 be the arrival rates for customers of class 1 and 2, respectively. Let $\mu_{11}, \mu_{12}, \mu_3$ be the service rates for the different classes. We assume that $\mu_{11} = \mu_{12}$; that is, both customer types have the same service requirements at the first server. We will denote the common service rate at

the first server by μ_1. In order to ensure that at least one stable policy exists, we assume that $\lambda_1 + \lambda_2 < \mu_1$ and $\lambda_1 < \mu_3$.

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.

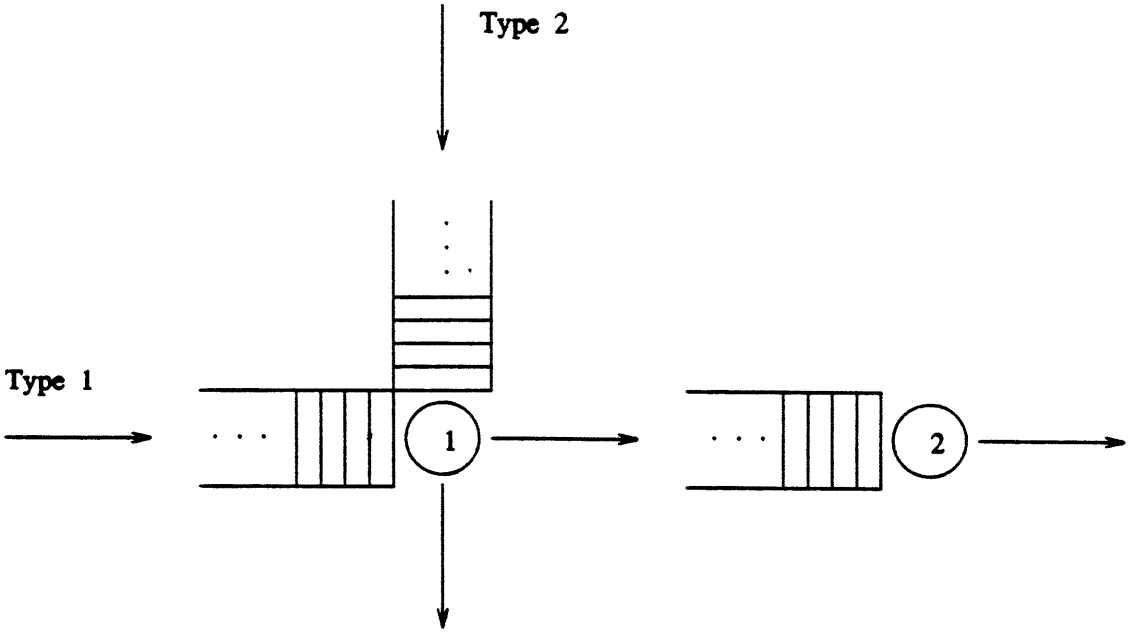

Figure 1: A simple two-station network

In the remainder of this section, we illustrate our methodology for deriving a lower bound on the optimal cost. To this effect, we show 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 non-empty set of 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 [Pasc].

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 $\mu_{11} + \mu_{12}$. 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 above 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 real or fictitious customer) occurs. We assume that the state vector $\vec{n}(t)$ is a right-continuous function of time so that $\vec{n}(\tau_k)$ 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 $\overline{B}_r(t)$ the event that node $\sigma(r)$ is not busy with a class r customer at time t.

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

$$\begin{aligned} E[R^2(\tau_{k+1}) \mid \vec{n}(\tau_k)] \;=\; & \lambda_1(R(\tau_k) + f(1))^2 + \lambda_2(R(\tau_k) + f(2))^2 + \\ & \mu_1 1\{B_1(\tau_k)\}(R(\tau_k) - f(1))^2 + \mu_1 1\{\overline{B}_1(\tau_k)\}R^2(\tau_k) + \\ & \mu_1 1\{B_2(\tau_k)\}(R(\tau_k) - f(2))^2 + \mu_1 1\{\overline{B}_2(\tau_k)\}R^2(\tau_k) + \\ & \mu_2 R^2(\tau_k) \end{aligned}$$

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

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

we obtain

$$\begin{aligned} E[R^2(\tau_{k+1}) \mid \vec{n}(\tau_k)] \;\geq\; & R^2(\tau_k) + \lambda_1 f^2 + \lambda_2 f^2 + \\ & \mu_1 1\{B_1(\tau_k)\}f^2 + \mu_1 1\{B_2(\tau_k)\}f^2 - \\ & 2\mu_1 R(\tau_k)f + (2\lambda_1 f + 2\lambda_2 f)R(\tau_k) \end{aligned} \tag{10}$$

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 $\vec{n}(\tau_k)$. In particular, under the invariant distribution of the two chains, we have

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

and

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

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

We now consider the Markov chain $\vec{n}(\tau_k)$ 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

$$E[1\{B_j(\tau_k)\}] = \frac{\lambda_j}{\mu_1} \qquad j = 1, 2.$$

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 equation (2) is the same as the conservation law for the multiclass M/M/1 queue (see [GeMi, Chap. 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 8 for more details on the application of our approach to the multiclass queue.

Note also, that equations (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 8 we prove for the general case of the multiclass queue and for work-conserving policies, that equations (3), (4) hold with equality if we give

preemptive priority to customers of class 1, 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 above described 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 $1/\mu_3$ if a policy is stable and $2\lambda_1 > \max(\mu_1, \mu_3)$. This argument can be carried out in more detail and leads to the following result; its proof is omitted and can be found in [Pasc].

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

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

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. But 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 were 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 8, 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 9.

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.

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 we have:

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 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

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

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 τ_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 by $\overline{B}_r(t)$ the event that node $\sigma(r)$ is not busy with a class r customer at time t. As in Theorem 3.1, we assume that the process $\vec{n}(t)$ is right-continuous.

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

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

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 assume that the *f-parameters* satisfy (17) because as we will see later in the proof this choice leads to tighter bounds. 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 above term, we use the fact

$$
1\{\text{server } i \text{ busy from some class } r \in S \cap C_i \text{ at } \tau_k\} \leq 1. \tag{19}
$$

It should now be apparent why we selected *f-parameters* satisfying (17). By doing so, we were able to aggregate certain indicator functions and use the inequality (19), instead of using the less tight inequalities $1\{B_r(\tau_k)\} \leq 1$.

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)

$$E[1\{B_r(\tau_k)\}] = \frac{\lambda_r}{\mu_r},$$

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. But, even if this is the case, it will 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 in [Kuma] (who attributes it to Meyn [MeDo]). He dealt with a network with deterministic routing and with special structure (re-entrant 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 which choice of the *f-parameters* satisfies (17). For every set S of

classes, (17) yields

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

which implies

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

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 $C_i \cap S$ non-empty. 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 $C_i \cap S$ non-empty. 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

$$f_S(r) = \frac{1}{\mu_r} + \sum_{r' \in S} p_{rr'} f_S(r'), \qquad r \in S. \tag{20}$$

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 8), 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

$$R(t) = \sum_{r=1}^{R} f(r) n_r(t) \tag{21}$$

where $f(r)$ are scalars that we call *f-parameters*. We let again τ_k be the sequence of transition times (due to real or fictitious customers) in the uniformized Markov chain. As in Section 4.1, we denote by $B_r(t)$ the event that node $\sigma(r)$ is busy with a class r customer at time t, and by $\overline{B}_r(t)$ the event that node $\sigma(r)$ is not busy with a class r customer at time t. We finally introduce $B_{0i}(t)$ to denote the event that node i is idle at time t. We then define

$$I_{rr'} = E[1\{B_r(\tau_k)\}n_{r'}(\tau_k)] \tag{22}$$

and

$$N_{ir'} = E[1\{B_{0i}(\tau_k)\}n_{r'}(\tau_k)], \tag{23}$$

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 equalities hold:*

$$2\mu_r I_{rr} - 2\sum_{r'=1}^{R} \mu_{r'} p_{r'r} I_{r'r} - 2\lambda_{0r}\lambda_r x_r = \lambda_{0r} + \lambda_r(1 - p_{rr}) + \sum_{r' \neq r} \lambda_{r'} p_{r'r} \qquad r = 1, \ldots, R \tag{24}$$

and

$$\mu_r I_{rr'} + \mu_{r'} I_{r'r} - \sum_{w=1}^{R} \mu_w p_{wr} I_{wr'} - \sum_{w=1}^{R} \mu_w p_{wr'} I_{wr} - \lambda_{0r}\lambda_{r'} x_{r'} - \lambda_{0r'}\lambda_r x_r = -\lambda_r p_{rr'} - \lambda_{r'} p_{r'r} \qquad \forall r, r' \text{ such that } r > r'. \tag{25}$$

Proof: We uniformize as in Theorem 4.1 and proceed similarly to obtain the recursion

$$\begin{aligned} E[R^2(\tau_{k+1}) \mid \vec{n}(\tau_k)] = \\ &\sum_{r=1}^{R} \lambda_{0r}(R(\tau_k) + f(r))^2 + \\ &\sum_{r=1}^{R} \mu_r 1\{B_r(\tau_k)\} \left[\sum_{r'=1}^{R} p_{rr'}(R(\tau_k) - f(r) + f(r'))^2 + p_{r0}(R(\tau_k) - f(r))^2 \right] + \\ &\sum_{r=1}^{R} \mu_r 1\{\overline{B}_r(\tau_k)\} R^2(\tau_k) \end{aligned}$$

Rearranging terms, taking expectations with respect to the invariant distribution, and using the fact that at steady-state we have

$$E[1\{B_r(\tau_k)\}] = \frac{\lambda_r}{\mu_r},$$

where λ_r is the solution of the traffic equations (15), we get:

$$2\sum_{r=1}^{R}\mu_r\left[\sum_{r'=1}^{R}p_{rr'}(f(r)-f(r'))+p_{r0}f(r)\right]E[1\{B_r(\tau_k)\}R(\tau_k)]-$$
$$2\sum_{r=1}^{R}\lambda_{0r}f(r)E[R(\tau_k)]=$$
$$\sum_{r=1}^{R}\lambda_{0r}f^2(r)+\sum_{r=1}^{R}\lambda_r\left[\sum_{r'=1}^{R}p_{rr'}(f(r)-f(r'))^2+p_{r0}f^2(r)\right] \tag{26}$$

Moreover, it is seen from (21) and (22) that

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

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 the equation below. (One easy way of obtaining that equation is to consider (26) for the special case where f is the rth unit vector.)

$$2\mu_r\left(p_{r0}+\sum_{r'\neq r}p_{rr'}\right)I_{rr}-2\sum_{r'\neq r}\mu_{r'}p_{r'r}I_{r'r}-2\lambda_{0r}\lambda_r x_r=$$
$$\lambda_{0r}+\lambda_r\left(p_{r0}+\sum_{r'\neq r}p_{rr'}\right)+\sum_{r'\neq r}\lambda_{r'}p_{r'r},$$

from which we easily obtain (24) since the transition probabilities add up to one.

Equality of the off-diagonal terms of Q and Q_0, similarly yields the next equation. (Equating the (r, r')th entries of Q and Q_0 is the same as considering (26) for the special case where f is a vector whose rth and r'th entries are 1 and all other entries are zero.) Due to symmetry, it suffices to consider $r > r'$. We have

$$\mu_r \left(p_{r0} + \sum_{r' \neq r} p_{rr'} \right) I_{rr'} - \mu_r p_{rr'} I_{rr} + \mu_{r'} \left(p_{r'0} + \sum_{w \neq r'} p_{r'w} \right) I_{r'r} - \mu_{r'} p_{r'r} I_{r'r'} - \sum_{w \neq r, r'} \mu_w p_{wr} I_{wr'} - \sum_{w \neq r, r'} \mu_w p_{wr'} I_{wr} - \lambda_{0r} \lambda_{r'} x_{r'} - \lambda_{0r'} \lambda_r x_r = -\lambda_r p_{rr'} - \lambda_{r'} p_{r'r}$$

which implies (25). □

In addition to the equalities in Theorem 4.2, some more equalities are provided by the result that follows.

Theorem 4.3 *For each node i of the network, each class r', and for every policy satisfying Assumption A, the following equality holds, in steady-state:*

$$\sum_{r \in C_i} I_{rr'} + N_{ir'} = \lambda_{r'} x_{r'} \tag{28}$$

Proof : Let us fix some node i. We note that the events

$$B_r(\tau_k) = \text{"server } i \text{ busy from class } r \text{ at } \tau_k\text{"}, \qquad r \in C_i,$$

and the event

$$B_{0i}(\tau_k) = \text{"server } i \text{ idle at } \tau_k\text{"}$$

are mutually exclusive and exhaustive. Thus:

$$E\left[n_{r'}(\tau_k) \left(\sum_{r \in C_i} 1\{B_r(\tau_k)\} + 1\{B_{0i}(\tau_k)\} \right) \right] = n_{r'} = \lambda_{r'} x_{r'}$$

Using the definitions (22), (23), we obtain (28). □

The equations provided by Theorems 4.2 and 4.3, together with the obvious inequalities $I_{rr'} \geq 0, N_{ir'} \geq 0$ and $x_i \geq 0$, define a polyhedron. This polyhedron contains as much information on the region of achievable performance as the polyhedron obtained by the

approach of Theorem 4.1. This is due to the fact that both polyhedra are derived using the same basic recursion for $R(\tau_k)$. Moreover in the nonparametric approach no inequalities are introduced, in contrast 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 proves formally such a relation between the two polyhedra and establishes that the nonparametric approach is at least as powerful as our first approach.

Theorem 4.4 *If the variables* $\{x_r,\ I_{rr'},\ N_{ir'};\ r,r'=1,\ldots,R,\ i=1,\ldots,N\}$ *are nonnegative and satisfy the equalities in Theorems 4.2 and 4.3, then the variables* $\{x_r,\ r=1,\ldots,R\}$ *satisfy the inequalities of Theorem 4.1.*

Proof : Let the variables $\{x_r,\ I_{rr'},\ N_{ir'};\ r,r'=1,\ldots,R,\ i=1,\ldots,N\}$ have the stated properties. Since equation (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 (28) 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

$$E[1\{\text{server } \sigma(r) \text{ busy from some class } r\in S\cap C_i \text{ at } \tau_k\}n_{r'}(\tau_k)] \le n_{r'} \qquad (29)$$

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

$$1\{\text{server } \sigma(r) \text{ busy from some class } r\in S\cap C_i \text{ at } \tau_k\} \le 1 \qquad (30)$$

and

$$1\{B_r(\tau_k)\} \ge 0 \qquad (31)$$

in order to get the bound (18). That is, we first wrote down the recursive equation, we then applied (30) and (31) 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 (29) 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 (28). □

Remarks : We can intuitively argue that (28) contains more information than the somewhat "crude" (29). Thus, we strongly believe that there are, in general, $\{x_r,\ r = 1,\ldots,R\}$ satisfying the inequalities of Theorem 4.1, such that there are no nonnegative values for the variables $I_{rr'},\ N_{ir'}$ with which

$$\{x_r,\ I_{rr'},\ N_{ir'};\ r,r' = 1,\ldots,R,\ i = 1,\ldots,N\}$$

would satisfy the equalities in Theorems 4.2 and 4.3.

We can now obtain a new bound on the optimal cost, by minimizing $\sum_{r=1}^{R} c_r x_r$ subject to the equality constraints of Theorems 4.2 and 4.3 and the nonnegativity constraints on all of the variables involved. As a consequence of Theorem 4.4, we see that this lower bound will be tighter (that is, at least as large) than the lower bound obtained using Theorem 4.1. In addition, the linear program that has to be solved in order to compute this lower bound 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.

5 Routing and Sequencing

In this section we relax the assumption that only sequencing decisions are involved. We extend our nonparametric variation of the 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. We let again τ_k be the sequence of transition times (due to real or fictitious customers) in the uniformized Markov chain. We also denote, as in Section 4.2, by $B_r(t)$ the event that node $\sigma(r)$ is busy with a class r customer at time t, by $\overline{B}_r(t)$ the event that node $\sigma(r)$ is not busy with a class r customer at time t and by $B_{0i}(t)$ the event that node i is idle at time t. Instead of the routing probabilities $p_{rr'}$ being given, we control whether class r becomes 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 to. If F_r is a singleton for all $r = 1, \ldots, R$ the problem is reduced to the class with no routing decisions allowed. By modifying the sets F_r we can adjust the level of routing control we can apply to the network.

We define the following variables:

$$u_r = E[1\{B_r(\tau_k)\}], \tag{32}$$

$$D_{rr'} = E[1\{B_r(\tau_k)\}p_{rr'}(\tau_k)], \tag{33}$$

$$G_{rr'j} = E[1\{B_r(\tau_k)\}p_{rr'}(\tau_k)n_j(\tau_k)], \tag{34}$$

$$I_{rj} = E[1\{B_r(\tau_k)\}n_j(\tau_k)], \tag{35}$$

where $1\{\cdot\}$ is the indicator function and the expectations are taken with respect to the invariant distribution. Notice that $1\{B_r(\tau_k)\}$, $p_{rr'}(\tau_k)$ express the sequencing and routing decisions respectively. Using the nonparametric method on the function $R(t) = \sum_{r=1}^{R} f(r)n_r(t)$ we can prove the following theorem.

Theorem 5.1 *For every scheduling policy satisfying Assumption A, the achievable space* $\{(n_r, u_r, D_{rr'}, G_{rr'j}, I_{rj})\}$ *is contained in the following polyhedron:*

$$\mu_r u_r - \sum_{l=1}^{R} \mu_l D_{lr} = \lambda_{0r} \qquad r = 1, \ldots, R \tag{36}$$

$$\begin{aligned} 2\mu_r I_{rr} - 2\sum_{l=1}^{R} \mu_l G_{lrr} - 2\lambda_{0r} n_r = \\ \lambda_{0r} + \mu_r(u_r - D_{rr}) + \sum_{l \neq r} \mu_l D_{lr} \qquad r = 1, \ldots, R \end{aligned} \tag{37}$$

and

$$\begin{aligned} \mu_r I_{rr'} + \mu_{r'} I_{r'r} - \sum_{w=1}^{R} \mu_w G_{wrr'} - \sum_{w=1}^{R} \mu_w G_{wr'r} - \lambda_{0r} n_{r'} - \lambda_{0r'} n_r = \\ -\mu_r u_r D_{rr'} - \mu_{r'} u_{r'} D_{r'r} \qquad \forall r, r' \text{ such that } r > r'. \end{aligned} \tag{38}$$

$$\sum_{r \in C_i} u_r \leq 1 \qquad i = 1, \ldots, N \tag{39}$$

$$\sum_{r \in C_i} I_{rj} \leq n_j \qquad r = 1, \ldots, R; \; i = 1, \ldots, N \tag{40}$$

$$u_r = \sum_{l=0}^{R} D_{rl} \qquad r = 1, \ldots, R \tag{41}$$

$$I_{rj} = \sum_{l=0}^{R} G_{rlj} \qquad r, j = 1, \ldots, R \tag{42}$$

$$D_{rr'} = G_{rr'j} = 0 \quad if \quad r' \notin F_r, \; j = 1, \ldots, R$$

$$n_r, \; u_r, \; D_{rr'}, \; G_{rr'j}, \; I_{rj} \geq 0.$$

Proof: We uniformize as in Theorem 4.1 and proceed similarly to obtain the recursion

$$\begin{aligned} &E[R(\tau_{k+1}) \mid \vec{n}(\tau_k)] = \\ &\quad \sum_{r=1}^{R} \lambda_{0r}(R(\tau_k) + f(r)) + \\ &\quad \sum_{r=1}^{R} \mu_r 1\{B_r(\tau_k)\} \left[\sum_{r'=1}^{R} p_{rr'}(\tau_k)(R(\tau_k) - f(r) + f(r')) + p_{r0}(\tau_k)(R(\tau_k) - f(r)) \right] + \\ &\quad \sum_{r=1}^{R} \mu_r 1\{\overline{B}_r(\tau_k)\} R(\tau_k) \end{aligned}$$

Rearranging, taking expectations and demanding that the above relations should hold for all *f-parameters*, (36) follows, which is the usual traffic equation involving routing decisions. Applying the methodology to $E[R^2(\tau_{k+1})]$ we establish (37) and (38). Finally (39), (40), (41), (42) are obvious from the definition of the variables. □

6 Closed Networks: Approximate Characterization

In this section we briefly outline how the nonparametric variation of the bounding method is applied to closed queueing 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 F customers always present in the closed network. We use exactly the same notation as in the open case with the only difference being 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. How to incorporate routing

decisions should be obvious by now.

The goal in closed networks is to maximize throughput or equivalently maximize the percentage of time servers are busy. As in the open case we only consider sequencing decisions and policies satisfying Assumption A(a) (Assumption A(b) is automatically satisfied). We use the notation of Section 2 and also the function

$$R(t) = \sum_{r=1}^{R} f(r) n_r(t).$$

In addition, we will use the definitions (22) and (32). In a closed network we are interested in maximizing the weighted throughput

$$\sum_{r=1}^{R} c_r \mu_r E[1\{B_r(\tau_k)\}],$$

where the maximization is over all policies satisfying Assumption A(a), and c_r the benefit from maximizing the throughput of class r. The following theorem provides a polyhedron that contains the achievable space of $\{(n_r, I_{rr'}, u_r)\}$.

Theorem 6.1 *For every scheduling policy satisfying Assumption A(a) the achievable space is contained in the following polyhedron:*

$$\mu_r u_r - \sum_{l=1}^{R} \mu_l u_l p_{lr} = 0 \qquad r = 1, \ldots, R \tag{43}$$

$$2\mu_r I_{rr} - 2\sum_{l=1}^{R} \mu_l p_{lr} I_{lr} = u_r \mu_r (1 - p_{rr}) + \sum_{l \neq r} \mu_l u_l p_{lr} \qquad r = 1, \ldots, R \tag{44}$$

and

$$\mu_r I_{rr'} + \mu_{r'} I_{r'r} - \sum_{w=1}^{R} \mu_w p_{wr} I_{wr'} - \sum_{w=1}^{R} \mu_w p_{wr'} I_{wr} = -\mu_r u_r p_{rr'} - \mu_{r'} u_{r'} p_{r'r} \qquad \forall r, r' \text{ such that } r > r'. \tag{45}$$

$$\sum_{l \in C_i} I_{lr} \leq n_r \qquad r = 1, \ldots, R, \; i = 1, \ldots, N \tag{46}$$

$$\sum_{l=1}^{R} n_r = F \tag{47}$$

$$n_r, I_{rr'}, u_r \geq 0.$$

Proof: We initially apply our bounding technique to the function $R(t)$. We uniformize as in Theorem 4.1 and proceed similarly to obtain the recursion

$$E[R(\tau_{k+1}) \mid \vec{n}(\tau_k)] = \sum_{r=1}^{R} \mu_r 1\{B_r(\tau_k)\} \left[\sum_{r'=1}^{R} p_{rr'}(R(\tau_k) - f(r) + f(r'))\right] + \sum_{r=1}^{R} \mu_r 1\{\overline{B}_r(\tau_k)\} R(\tau_k)$$

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

$$\sum_{r=1}^{R} f(r)\mu_r u_r - \sum_{l=1}^{R} \mu_l u_l \sum_{r'=1}^{R} p_{lr'} f(r') = 0.$$

Since this equality holds for all *f-parameters* we obtain (43).

Applying the methodology to the potential function $R^2(t)$ we obtain

$$E[R^2(\tau_{k+1}) \mid \vec{n}(\tau_k)] = \sum_{r=1}^{R} \mu_r 1\{B_r(\tau_k)\} \left[\sum_{r'=1}^{R} p_{rr'}(R(\tau_k) - f(r) + f(r'))^2\right] + \sum_{r=1}^{R} \mu_r 1\{\overline{B}_r(\tau_k)\} R^2(\tau_k)$$

As in the proof of Theorem 4.2, we define a vector $f = (f(1), \ldots, f(R))$. Rearranging terms, taking expectations with respect to the invariant distribution we obtain that

$$f^T Q f = f^T Q_0 f,$$

for some symmetric matrices Q, Q_0, and from that $Q = Q_0$. From this (44), (45) follow, expressing the equality of the diagonal and off-diagonal terms of Q and Q_0 respectively. Finally, (46) is obvious while (47) expresses the fact that there F customers in the closed network. □

Obviously, maximizing the linear function $\sum_{r=1}^{R} c_r \mu_r u_r$ over the polytope of the previous theorem yields an upper bound on the optimal weighted throughput.

7 Higher Order Interactions and Nonlinear Characterizations

Until now the methodology we have developed offers *polyhedral* spaces that contain the achievable region and takes into account *pairwise* interactions among classes in the network. In this section we significantly 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 space in a systematic way by using ideas from the powerful methodology of semi-definite programming.

In this way, we obtain a sequence of progressively more complicated nonlinear approximations (relaxations) which 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 [GeMi].

7.1 Higher Order Interactions

Let us reflect on the methodology used so far. We use the function $R(t) = \sum_{r=1}^{R} f(r)n_r(t)$. The dynamics of the system are then expressed in terms of the recursion

$$E[R(\tau_{k+1}) \mid \vec{n}(\tau_k)] = G(R(\tau_k), \vec{n}(\tau_k), \vec{1}\{A(\tau_k)\}, \Lambda),$$

where $G(\cdot)$ is a function that expresses the dynamics, $\vec{1}\{A(\tau_k)\}$ is the vector of the possible events and decisions that can take place in the system and Λ is the set of parameters, which is known as data and fully describe the system. Demanding that the recursion $E[R(\tau_{k+1}) \mid \vec{n}(\tau_k)]$ holds for all *f-parameters*, we obtain the traffic flow equations. From the recursion $E[R^2(\tau_{k+1}) \mid \vec{n}(\tau_k)]$ and by selecting proper *f-parameters* (parametric method), or by demanding that the recursion holds for all *f-parameters* (nonparametric method) we obtain linear inequalities (parametric method) or a set of $R(R+1)/2$ linear equalities in terms of new variables (nonparametric method). In this respect the nonparametric method is more powerful, because it is independent of the choice of parameters and leads to sharper characterizations as we have proved in Section 4. By its nature, the methodology will

only take into account pairwise interactions among the various classes, which are present if one expands the square terms in the recursion. For example the nonparametric method introduces variables $E[1\{B_r(\tau_k)\}n_j(\tau_k)]$, taking into account the interactions of classes r and j.

These observations naturally lead us to the following generalizations of the methodology, which we apply to a multiclass open queueing network, where only sequencing decisions are involved. Using the notation of Section 2, we apply the nonparametric method to $E[R^3(\tau_{k+1}) \mid \vec{n}(\tau_k)]$, and define, in addition to $I_{rj} = E[1\{B_r(\tau_k)\}n_j(\tau_k)]$, new variables

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

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

$$J_{rjk} = E[n_r(\tau_k)n_j(\tau_k)n_k(\tau_k)]$$

and obtain a set of linear equalities in the set of the new variables as follows. We modify Assumption A(b) and assume that $E[n_r^3(t)] < \infty$.

Theorem 7.1 *In a multiclass open queueing network, where only sequencing decisions are involved, and for every scheduling policy satisfying the modified Assumption A, the achievable space* $\{(n_r, I_{rj}, H_{rjk}, M_{jk}\ J_{rjk})\}$ *is contained in the following space:*

$$-\lambda_r + \sum_{r=1}^{R} \lambda_{r'} p_{r'r} + 3\mu_r(1-p_{rr})I_{rr} + 3\sum_{r' \neq r} \mu_{r'} p_{r'r} I_{r'r} - 3\mu_r H_{rrr} +$$
$$3\sum_{r'=1}^{R} \mu_{r'} p_{r'r} H_{r'rr} + \lambda_{0r} + 3\lambda_{0r} n_r + 3\lambda_{0r} M_{rr} = 0 \qquad r = 1, \ldots, R \tag{48}$$

$$3\lambda_{0r'} M_{rr} + 3\lambda_{0r} n_{r'} + 6\lambda_{0r} M_{rr'} + 3\lambda_r p_{rr'} - 3\lambda_{r'} p_{r'r} + 3\mu_r(1-p_{rr})I_{rr'} -$$
$$6\mu_r p_{rr'} I_{rr} - 6\mu_{r'} p_{r'r} I_{r'r} + 3\sum_{w \neq r} \mu_w p_{wr} I_{wr'} + 3\sum_{l=1}^{R} \mu_l p_{lr'} H_{lrr} -$$
$$3\mu_{r'} H_{r'rr} - 6\mu_r H_{rrr'} + 6\sum_{w=1}^{R} \mu_w p_{wr} H_{wrr'} \qquad \forall r, r' \text{ such that } r > r'. \tag{49}$$

$$\lambda_{0k} M_{rj} + \lambda_{0r} M_{jk} + \lambda_{0j} M_{rk} - \mu_r(1-p_{rr})H_{rjk} - \mu_j(1-p_{jj})H_{jrk} - \mu_k(1-p_{kk})H_{krj} +$$

$$\sum_{l\neq r,j,k} \mu_l p_{lr} H_{ljk} + \sum_{l\neq r,j,k} \mu_l p_{lj} H_{lrk} + \sum_{l\neq r,j,k} \mu_l p_{lk} H_{lrj} - \mu_r p_{rj} I_{rk} + \mu_r p_{rj} H_{rrk} -$$
$$\mu_r p_{rk} I_{rj} + \mu_r p_{rk} H_{rrj} - \mu_j p_{jk} I_{jk} + \mu_j p_{jr} H_{jjk} - \mu_j p_{jk} I_{jr} + \mu_j p_{jk} H_{jjr} -$$
$$\mu_k p_{kr} I_{kj} + \mu_k p_{kr} H_{kkj} - \mu_k p_{kj} I_{kr} + \mu_k p_{kj} H_{kkr} = 0 \qquad \forall r,j,k \text{ such that } r<j<k. \quad (50)$$

$$\sum_{l\in C_i} I_{lr} \leq n_r \qquad r=1,\ldots,R,\ i=1,\ldots,N \quad (51)$$

$$\sum_{l\in C_i} H_{ljk} \leq M_{jk} \qquad j,k=1,\ldots,R,\ i=1,\ldots,N \quad (52)$$

$$n_r,\ I_{rj},\ H_{rjk},\ M_{jk}\ J_{rjk} \geq 0. \quad (53)$$

Proof: We uniformize as in Theorem 4.1 and proceed similarly to obtain the recursion for $R^3(t)$. We express the recursion as a third degree multivariable polynomial of $\vec{f}$ which should be identically equal to zero for all *f-parameters*. Equating the R coefficients of the monomials $f(r)^3$, the $R(R-1)$ coefficients of the monomials $f(r)^2 f(r')$ and the $\sum_{k=1}^{R}(k-1)(k-2)/2$ coefficients of the monomials $f(r)f(j)f(k)$ for $r<j<k$, we obtain (48), (49), (50), respectively, after some algebra. (51), (52) and (53) are obvious. □

The new variables we introduced take into account interactions among three classes in the system and as such we expect that they lead to more powerful characterizations. Another advantage of the methodology is that now one can 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 one can continue with this idea further by applying the nonparametric method to $E[R^i(\tau_{k+1}) \mid \vec{n}(\tau_k)]$ 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 with P_i the space obtained by applying the nonparametric method to $E[R^i(\tau_{k+1}) \mid \vec{n}(\tau_k)]$, the approximating space up to ith order interactions is $\cap_{l=1}^{i} P_l$. The dimension of the space and the number of constraints is $O(R^i)$, which even for moderate i is quite expensive.

The explicit derivation of the equalities of these spaces is algebraically involved but conceptually very simple. Since the only operations involved in the derivation is manipulation of multivariable polynomials we used the software program Mathematica to derive the equations in Theorem 7.1. Since it is rather routine for Mathematica to automatically gen-

erate constraints, one could imagine that this *automatic generation* could be combined with an algorithm to find lower bounds for the achievable performance that would progressively add constraints in the problem. This is exactly how large scale combinatorial problems are solved to optimality using polyhedral methods.

7.2 Nonlinear Interactions

We briefly outline in this section how ideas from semi-definite programming can be used to obtain nonlinear constraints on the achievable space.

Let $\vec{Y}$ be a vector of random variables. Let Q be a symmetric positive semi-definite matrix. Clearly,

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

which implies that

$$E[\vec{Y}^T Q \vec{Y}] \geq E[\vec{Y}^T] Q E[\vec{Y}], \tag{54}$$

which is Jensen inequality applied to the convex function $x^T Q x$. Notice that (54) should hold for every symmetric semi-definite matrix Q. By selecting particular values for matrices Q, one obtains a family of inequalities.

This methodology can be used to generate families of quadratic inequalities for the model of the previous subsection as follows. As an example, for a fixed $r = 1, \ldots, R$ by selecting as the random vector $\vec{Y}$ the vector $(1\{B_r(\tau_k)\} n_j(\tau_k))$, $j = 1, \ldots, R$ and using the identity $1\{B_r(\tau_k)\} = (1\{B_r(\tau_k)\})^2$, we obtain the quadratic inequalities

$$\sum_{i,j} H_{rij} Q_{ij} \geq \sum_{i,j} Q_{ij} I_{ri} I_{rj}, \; r = 1, \ldots, R. \tag{55}$$

Choosing specific Q values we could generate families of quadratic inequalities. Instead we will impose the constraints of the form (54) for all choices of Q. Let Z be the polyhedral space of Theorem 7.1. A lower bound on the optimal solution value has the form:

$$z_{LB} = \min \sum_{r=1}^{R} c_r n_r$$

subject to:

$$(n_r, I_{rj}, H_{rjk}, M_{jk}, J_{rjk}) \in Z$$

$$\sum_{i,j} H_{rij} Q_{ij} \geq \sum_{i,j} Q_{ij} I_{ri} I_{rj}, \; r = 1, \ldots, R, \forall Q \geq 0. \tag{56}$$

For every fixed $Q \geq 0$ (semi-definite) the constraint (56) is convex in the variables I_{ri}. As a result given a subfamily of constraints of the form (56) we have a convex programming problem. On the other hand, the optimal solution value z_{LB} can be obtained by a cutting plane type algorithm which solves at each stage the following separation problems:
SEPARATION: Given a $(n_r, I_{rj}, H_{rjk}, M_{jk}, J_{rjk}) \in Z$ solve for each r:

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

subject to:

$$Q \geq 0$$

If $z_{SEP} \geq 0$ for $r = 1, \ldots, R$, then the current vector satisfies all constraints of the form (56) and it is optimal. If not, then a semi-definite matrix Q has been found for which the corresponding constraint is violated by the current vector. We can then add this constraint to the current subfamily of constraints, resolve the convex programming problem, and continue similarly.

We note that the separation problem is an instance of a semi-definite programming problem which can be solved efficiently by a simplex type and interior point methods (see [Al]). Therefore, the overall algorithm would be very efficient. From a complexity point of view the overall algorithm would run in polynomial time if one uses the ellipsoid algorithm or a variant like Vaidya's algorithm, since the separation problem is solvable in polynomial time by an interior point algorithm.

In order to add higher order nonlinearities we can also add some more general nonlinear constraints involving polynomials of degree $i - 1$ of the type

$$E[1\{B_r(\tau_k)\} n_j^h(\tau_k)] \geq E[n_j(\tau_k)\}]^h, \; h = 1, \ldots i - 1,$$

which hold because of Jensen inequality, since the variables involved are nonnegative and hence the functions x^h are convex. In this way we obtain a sequence of progressively more complicated nonlinear spaces that approximate the achievable region.

8 Single Station Networks and Homogeneous Open Networks: Complete Characterizations

In this section we investigate the connections of our methodology with mostly known (although we do obtain a new result) exact characterizations of the achievable region in simpler systems based on conservation laws. Our overall goal in this section is to show that our first order methods are equally powerful with conservation laws and lead to explicit characterizations. Moreover, the nonparametric method offers a reformulation of the achievable space, with a polynomial number of variables and constraints, which is interesting both from a probabilistic, but also from a combinatorial point of view.

We show that our bounds give an exact characterization of the achievable region for a) single station systems and b) open networks in which all classes are stochastically the same as soon as the classes enter the network, i.e., $p_{r,r'} = p_{i,i'}$ for all r, r' such that: $\sigma(r) = i$ and $\sigma(r') = i'$ and $\mu_r = \mu_i$ for all r with $\sigma(r) = i$. Regarding multiclass single stations, we examine a multiclass M/M/1 queue with (Klimov's problem [Klim]) and without Bernoulli feedback, where each class can have distinct service requirements under work-conserving, preemptive policies. The achievable region for the multiclass queue is derived in [GeMi] based on conservation laws. The achievable region for Klimov's problem with preemption is, to the best of our knowledge, not known. Tsoucas [Tsou] derives the form of the achievable region for Klimov's problem under non-preemptive policies and general service requirements. We address the preemptive case for exponential service requirements. Our results in this case are explicit, since we compute all the parameters in closed form, while Tsoucas does not give explicit formulae for some of the parameters in his characterization. Regarding homogeneous open networks, we remark that our methods reproduce the exact characterization obtained in Ross and Yao [RoYa] using conservation laws. We do not reproduce the results here since they are identical with those obtained in [RoYa] and the methodology to obtain them is the same as in the multiclass queue. We introduce conservation laws and their connections with polyhedral performance regions.

8.1 Strong conservation laws and extended polymatroids

In this section we summarize briefly some material from Shantikumar and Yao [ShYa] and Bertsimas and Niño-Mora [BeNi].

Let $E = \{1, \ldots, n\}$ be a finite set. Let x denote an n-vector, with components x_i, for $i \in E$. For $S \subseteq N$, let us write $S^c = E \setminus S$. Let 2^E denote the class of all subsets of E. Let $b: 2^E \to \Re_+$ be a set function, that satisfies $b(\emptyset) = 0$. Let $f = (f_i^S)_{i \in E,\ S \subseteq E}$ be a matrix that satisfies

$$f_i^S > 0, \quad \text{for} \quad i \in S \qquad \text{and} \qquad f_i^S = 0, \quad \text{for} \quad i \in S^c. \tag{57}$$

Let $\pi = (\pi_1, \ldots, \pi_n)$ be a permutation of E. Let $v(\pi)$ be the unique solution of the linear system

$$\sum_{i=1}^{j} f_{\pi_i}^{\{\pi_1, \ldots, \pi_j\}} x_{\pi_i} = b(\{\pi_1, \ldots, \pi_i\}), \quad \text{for} \quad i = 1, \ldots, n. \tag{58}$$

Let

$$\mathcal{P}(f, b) = \{ x \in \Re_+^n : \sum_{i \in S} f_i^S x_i \geq b(S), \quad \text{for } S \subseteq E \} \tag{59}$$

and

$$\mathcal{B}(f, b) = \{ x \in \Re_+^n : \sum_{i \in S} f_i^S x_i \geq b(S), \quad \text{for } S \subset E \quad \text{and} \quad \sum_{i \in E} f_i^E x_i = b(E) \}. \tag{60}$$

The following definition is due to Bhattacharya *et al.* [BGT].

Definition 8.1 (Extended Polymatroid) We say that the polyhedron $\mathcal{P}(f, b)$ is an *extended polymatroid* with base set E if for every permutation π of E, $v(\pi) \in \mathcal{P}(f, b)$. In this case we say that the polytope $\mathcal{B}(f, b)$ is the *base* of the extended polymatroid $\mathcal{P}(f, b)$.

Notice that if $f_i^S = 1$ then the polyhedron $\mathcal{P}(f, b)$ is a classical polymatroid.

Shantikumar and Yao [ShYa] formalized a definition of *strong conservation laws* for performance measures in general multiclass queues, that implies a polymatroidal structure in the performance space. Bertsimas and Niño [BeNi] present a more general definition of *strong conservation laws* that implies an extended polymatroidal structure in the performance space that has several interesting and important implications. Consider a general multiclass queueing system and let $\mathcal{U}$ be the class of all nonidling and nonanticipative scheduling policies (see Gelenbe and Mitrani [GeMi]). We consider $\mathcal{U}$ to be the class of *admissible* policies. Let x_i^u be a performance measure of class i jobs, $i \in E = \{1, \ldots, n\}$ under policy $u \in \mathcal{U}$. We assume that x_i^u is an expectation. Let x^u be the corresponding

performance vector. Let x^π denote the performance vector under an absolute priority rule that assigns priorities to jobs according to the permutation $\pi = (\pi_1, \ldots, \pi_n)$ of E, where jobs of class π_n have maximum priority.

Definition 8.2 (Strong Conservation Laws) The performance vector x is said to satisfy *strong conservation laws* if there exist a function $b : 2^E \to \Re_+$ such that $b(\emptyset) = 0$ and a matrix $f = (f_i^S)_{i \in E, S \subseteq E}$ satisfying (57) such that:

$$(a) \quad b(S) = \sum_{i \in S} f_i^S x_i^\pi, \quad \text{for all } \pi : \{\pi_1, \ldots, \pi_{|S|}\} = S \quad \text{and } S \subseteq E; \tag{61}$$

$$(b) \quad \sum_{i \in S} f_i^S x_i^u \geq b(S), \quad \text{for all } S \subset E \qquad \text{and} \qquad \sum_{i \in E} f_i^E x_i^u = b(E), \quad \text{for all } u \in \mathcal{U}. \tag{62}$$

The connection of conservation laws and extended polymatroids is reflected in the following theorem.

Theorem 8.1 *(Bertsimas and Niño [BeNi]) Assume that the performance vector x satisfies strong conservation laws* (61) *and* (62). *Then*
(a) The vertices of $\mathcal{B}(f, b)$ are the performance vectors of the absolute priority rules, and $x^\pi = v(\pi)$, for every permutation π of E.
(b) The extended polymatroid base $\mathcal{B}(f, b)$ is the performance space.

The previous theorem makes it relatively easy to check whether the performance space has an extended polymatroid structure.

The fundamental structural property of an extended polymatroid is that minimizing a linear function $\sum_{i \in E} c_i n_i$ over $\mathcal{B}(f, b)$ can be achieved by the following adaptive greedy algorithm originally proposed by Klimov [Klim] and proven using dynamic programming. For a proof of its optimality using linear programming duality (the variables y_S defined in the course of the algorithm are the optimal dual variables corresponding to the LP: $\min c\, n$, $n \in \mathcal{B}(f, b)$ see [BeNi], where c, n, are the vectors of c_i's, n_i's, respectively).

Theorem 8.2 *The performance vector corresponding to the optimal priority rule*

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

for the problem min $c\,n$, $n \in \mathcal{B}(f,b)$ *is the solution of the system of equations:*

$$\sum_{i=1}^{k} f_{\pi_i}^{\{\pi_1,\pi_2,\ldots,\pi_i\}} n_{\pi_i} = b(\{\pi_1,\pi_2,\ldots,\pi_k\}) \quad k = 1,2,\ldots,n \tag{63}$$

where the optimal ordering $\pi_1,\pi_2,\ldots,\pi_n$ *is given by the algorithm:*

Step 1:

$$E^0 \leftarrow E$$

$$y_{E^0} = \min_{i \in E^0} \left\{ \frac{c_i'}{f_{E^0}(i)} \right\}$$

$$\pi_n = \arg\min \left\{ \frac{c_i'}{f_{E^0}(i)} \right\}$$

Step 2: *For* $k = 1,2,\ldots,n-1$

$$E^k \leftarrow N^{k-1} \setminus \{\pi_{n-k+1}\}$$

$$y_{E^k} = \min_{i \in E^k} \left\{ \frac{c_i' - \sum_{j=0}^{k-1} f_{Ej}(i) y_{Ej}}{f_{E^k}(i)} \right\}$$

$$\pi_{n-k} = \arg\min \left\{ \frac{c_i' - \sum_{j=0}^{k-1} f_{Ej}(i) y_{Ej}}{f_{E^k}(i)} \right\}$$

8.2 A Multiclass Queue

We consider a multiclass queue with n customer classes (Figure 2). Customers of class i enter the station in a Poisson stream of rate λ_i. The station has a single server and each class of customers requires service time exponentially distributed with rate μ_i. Let $n_i(t)$ be the number of class i customers present in the system at time t. Let x_i, n_i be the expected response time and the expected number of class i customers in steady state, respectively.

Let E be set of classes. Let $\rho_i = \lambda_i/\mu_i$ be the traffic intensity of class i customers.

Theorem 8.3 *The polyhedron:*

$$\textbf{P1:} \qquad \sum_{i \in S} \frac{n_i}{\mu_i} \geq \frac{\sum_{i \in S}(\rho_i/\mu_i)}{1 - \sum_{i \in S} \rho_i} \quad \forall S \subset E \tag{64}$$

$$\sum_{i \in N} \frac{n_i}{\mu_i} = \frac{\sum_{i \in N}(\rho_i/\mu_i)}{1 - \sum_{i \in N} \rho_i} \tag{65}$$

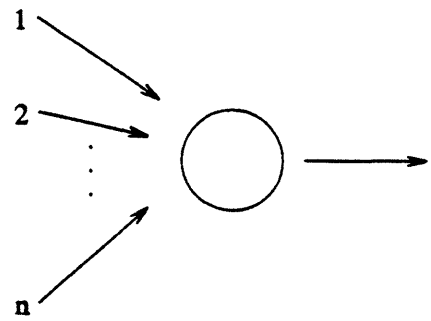

Figure 2: A multiclass queue

$$n_i \in \Re_+$$

is the achievable space for the multiclass queue for work-conserving policies satisfying Assumption A.

Proof : We uniformize the underlying chain and define the uniform rate to be:

$$\nu = \sum_{i=1}^{n} (\lambda_i + \mu_i)$$

Without loss of generality we assume that $\nu = 1$. Following the steps of our method for a subset S of $E = \{1, 2, \ldots, n\}$ we define the potential function $(R^S(t))^2$ where

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

Dropping S from $R^S(t)$ and $f_S(i)$ we get:

$$\begin{aligned} E[R^2(\tau_{k+1}) \mid \vec{n}(\tau_k)] \;=\; & \sum_{i \in S} \lambda_i (R(\tau_k) + f(i))^2 + \sum_{i \notin S} \lambda_i R^2(\tau_k) + \\ & \sum_{i \in S} \mu_i \, 1\{B_i(\tau_k)\} (R(\tau_k) - f(i))^2 + \\ & \sum_{i \in S} \mu_i \, 1\{\overline{B}_i(\tau_k)\} R^2(\tau_k) + \\ & \sum_{i \notin S} \mu_i R^2(\tau_k). \end{aligned}$$

We choose

$$f(i) = \frac{1}{\mu_i}, \; \forall i \in S,$$

and using:

$$1\{\text{server busy from some class } i \in S \text{ at } \tau_k\} \leq 1 \tag{66}$$

we obtain (64).

Therefore, $(P1)$ includes the achievable region. We next observe that inequality (64) holds with equality for work-conserving policies, when preemptive priority is given to the classes in the subset S. If preemptive priority is given to the classes in S we have:

$$R(\tau_k)1\{\text{server busy from some class } i \in S \text{ at } \tau_k\} = R(\tau_k)$$

because when $R(\tau_k) \neq 0$ (that is 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 definitely be working on a customer of classes $i \in S$. Otherwise, when $R(\tau_k) = 0$ the above equation holds trivially. Therefore, (64) holds with equality in this case. In particular, for $S \equiv E$, then equation (64) holds with equality for work-conserving policies and thus for work conserving policies we obtain (65).

In order to show that $(\dot{P}1)$ is exactly the performance space we observe that the performance vector (n_i/μ_i) satisfies strong conservation laws in the sense of Definition 8.2. Applying Theorem 8.1 we establish that $(P1)$ is a polymatroid having $n!$ extreme points corresponding to the $n!$ preemptive priorities rules and the performance vector of each priority rule is achievable. Thus, since every point in the polyhedron can be written as a convex combination of its extreme points $z_1, \ldots, z_{n!}$ with coefficients $a_1, \ldots, a_{n!}$, there exists a randomized policy that uses the priority rule corresponding to z_i with probability a_i that achieves the performance at this point. Therefore, (P1) is exactly the performance space of the multiclass queue. □

Remarks : Polyhedron $(P1)$ is a polymatroid and therefore, the greedy algorithm of Theorem 8.2 solves the problem of minimizing a linear function over $(P1)$ giving rise to the $c\mu$ rule.

We now apply the nonparametric method to find an alternative characterization of the achievable region that has a polynomial number of variables and constraints. Let $B_0(t)$ be the event that the single server is idle at time t.

Let us first define in analogy with (22) and (23):

$$I_{ij} = E[1\{B_i(\tau_k)\}n_j(\tau_k)] \tag{67}$$

$$N_j = E[1\{B_0(\tau_k)\}n_j(\tau_k)]. \tag{68}$$

Note that for work conserving policies $N_j = 0$.

Theorem 8.4 *For the multiclass queue and for work-conserving policies satisfying the Assumption A the following polyhedron P2:*

$$\mathbf{P2:} \qquad \mu_i I_{ii} - \lambda_i n_i = \lambda_i \quad i = 1, \ldots, n \tag{69}$$

$$\mu_i I_{ij} + \mu_j I_{ji} - \lambda_j n_i - \lambda_i n_j = 0 \quad \forall i, j, \ i \neq j \tag{70}$$

$$\sum_{i \in N} I_{ij} = n_j \quad j = 1, \ldots, n \tag{71}$$

$$n_i, I_{ij} \in \Re^+$$

projected in the $n_i, i = 1, 2, \ldots, n$ space yields $P1$.

Proof : Applying Theorem 4.2 for the multiclass queue, we immediately obtain that the nonparametric method yields polyhedron P_2. Let $P2'$ the projection of $P2$ in the space of n_i's. We want to show that $P2' \equiv P1$. In Theorem 4.4 we have shown that $P2' \subseteq P1$. Since $P1$ is exactly the achievable space it follows that $P1 \subseteq P2'$, from which the result $P2' \equiv P1$ follows. For a purely combinatorial derivation see [Pasc]. □

The previous theorem is an interesting reformulation of conservation laws. It states that a polymatroid polytope which is defined by $2^n - 1$ constraints can be transformed to a polytope defined in an augmented space of dimension $O(n^2)$ that has $O(n^2)$ constraints. It has been conjectured in the combinatorial optimization community that problems solvable in polynomial time have polynomial formulations. The previous theorem shows that this conjecture is indeed correct for the special case of the polymatroid polyhedron in a multiclass queue.

8.3 Klimov's Problem

Consider the single-server station of Fig. 3. Customers of class $i \in E = \{1, 2, \ldots, n\}$ arrive in the system according to a Poisson process of rate λ_i and have an exponentially distributed service time with mean $1/\mu_i$. Upon service completion a class i customer is fedback in the system as a class j customer with probability p_{ij}, while with probability p_{i0} he leaves the system. Let n_i be the expected number of customers of class i in steady state.

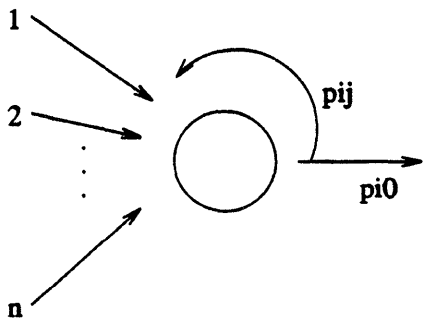

Figure 3: Klimov's problem.

The server is using a preemptive, work conserving discipline satisfying Assumption A. We show that for this problem also the polyhedron obtained from our method in Section 4 characterizes the achievable region exactly. The derived polyhedron has exactly the same structure as the polyhedron derived in [Tsou] for the M/G/1 case, under non-preemptive policies. In fact, the explicit form of the polytope is not given in [Tsou]; the rhs of the inequalities that define the polytope is an unknown function satisfying some properties. In contrast, we will explicitly define the polyhedron.

The traffic equations for the above system are:

$$\hat{\lambda}_i = \lambda_i + \sum_{j=1}^{n} \hat{\lambda}_j p_{ji}. \tag{72}$$

Our characterization is as follows:

Theorem 8.5 *For every work-conserving policy satisfying Assumption A, the achievable space for the Klimov problem is given by the polyhedron:*

$$\textbf{P3:} \qquad \sum_{i \in S} f_S(i) n_i \geq \frac{N'(S)}{D'(S)} \quad \forall S \subset E \tag{73}$$

$$\sum_{i \in E} f_S(i) n_i = \frac{N'(E)}{D'(E)} \tag{74}$$

$$n_i \in \Re_+$$

where:

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

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

and

$$f_S(i) = \frac{1}{\mu_i} + \sum_{j \in S} p_{ij} f_S(j). \tag{75}$$

Proof : In order to show that inequalities (73) are necessary we apply Theorem 4.1 directly for this case, where the *f-parameters* are chosen in (75). As in Theorem 8.3 we observe, using the method of deriving these inequalities, that inequality (73) holds with equality, for work-conserving policies when preemptive priority is given to the set S of classes. In particular, for $S = E$ (73) holds with equality, and therefore (74) is necessary. Therefore, the performance vector $\vec{n}$ satisfies strong conservation laws. Applying Theorem 8.1 we establish that $(P3)$ is a polymatroid having $n!$ extreme points corresponding to the $n!$ preemptive priorities rules and the performance vector of each priority rule is achievable. Thus, since every point in the polyhedron can be written as a convex combination of its extreme points $z_1, \ldots, z_{n!}$ with coefficients $a_1, \ldots, a_{n!}$, there exists a randomized policy that uses the priority rule corresponding to z_i with probability a_i that achieves the performance at this point. Therefore, $(P3)$ is exactly the achievable space. □

9 Numerical Results for Open Networks

In this section we provide some numerical results in order to evaluate the performance of our bounding techniques for open networks where only sequencing decisions are involved. In particular, we provide three network examples and for each of these examples and for various traffic conditions we calculate:

1. The lower bound on achievable performance according to the approach developed in Section 4.1.

2. The lower bound on achievable performance according to the nonparametric variation of the method developed in Section 4.2.

3. The performance of the FCFS policy.

4. The performance of the best policy we were able to found which serves as an upper bound.

In this way, we are able to evaluate the tightness of our lower bound. In fact, since the optimal is not known for each case, we cannot calculate the closeness of our lower bound to the optimal policy. Instead, we will calculate its closeness to the upper bound which of course is an overestimate. In particular, we will calculate the *efficiency* of the bound which we define as:

$$\text{efficiency} = \frac{\text{Best Lower Bound}}{\text{Best Upper Bound}} 100\%$$

9.1 A 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:

> **Policy 1** : 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.

An alternative policy is:

> **Policy 2** : 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. Idle at station 1 when there are B or more customers at station 2 and no type 2 customer is present at station 1.

The threshold B in both policies is constant and its optimal value was calculated via simulation. Policy 1 was proposed in [HaWe1] where the Brownian network model approach

was used. Intuition seems to suggest that when c_1 and c_3 are comparable, policy 1, which is work-conserving is preferable. But when $c_3 \gg c_1$ then policy 2 should be closer to optimal.

"Lower Bnd. 1" and "Lower Bnd. 2" in the table correspond to the bound developed in Section 4.1 and Section 4.2, respectively. Costs were chosen in order to have as objective function the total expected number of customers in the network, i.e., $(c_1 = c_3 = \lambda_1, c_2 = \lambda_2)$. For this is the reason, threshold policy 1 was simulated and not the threshold policy 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. Finally, note that by ρ_A, ρ_B we denote the total traffic intensities at station 1 and station 2, respectively.

Load Node 1-Node 2	Lower Bnd. 1	Lower Bnd. 2	FCFS	Thresh. Policy	Effic.
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	9.18	9.18	10.5	10.44	88%
LIGHT-MEDIUM	1.61	1.61	2.17	2.16	75%
HEAVY-MEDIUM	9.6	9.6	10.5	9.98	96%
MEDIUM-LIGHT	1.9	1.9	2.17	2.14	89%

Table 1: Numerical results for the network of Figure 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

Table 2: Data for the experiments of Table 1.

It is interesting that the efficiency of our lower bound is of approximately the same order of magnitude as the efficiency of the "pathwise bound" derived in [OuWe], 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 behaviour is mainly due to the fact that the threshold policy behaves better as the traffic gets heavier (see [HaWe1]). Moreover, the efficiency of the bounds is better in imbalanced traffic conditions.

9.2 A Four-Class Network Example

Consider the network of Figure 4. Customers enter the network in a Poisson stream of rate λ and they visit stations 1,2,1,2, in that order before exiting the network, forming classes 1,2,3,4 respectively. The single servers at stations 1,2 has service times exponentially distributed with rates μ_1,μ_2 respectively.

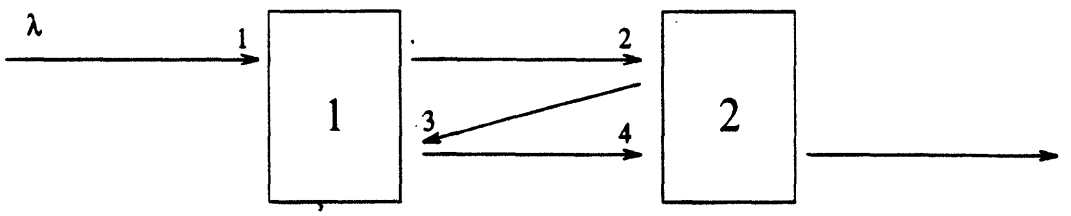

Figure 4: A Four-Class Network Example.

Table 3 compares our lower bounds on attainable performance with FCFS and the best found policy for various load conditions, providing also the efficiency of the bound. "Lower Bnd. 1" and "Lower Bnd. 2" in the table correspond to the bound developed in Section 4.1 and Section 4.2, respectively. Costs throughout the experiments reported in the table were chosen to be:

$$c_1 = 1.5,\ c_2 = 1.3,\ c_3 = 1.2,\ c_4 = 1.$$

In this specific example the best policy we were able to find, for each load condition we considered, happens to be a strict priority one. Note that we only considered non-preemptive policies. It is interesting that not a single policy was optimal for every case we considered. More precisely the following two policies were competing:

Policy 1: Give at station 1 highest priority to class 3 and lowest to class 1 ($3 \to 1$) and give at station 2 highest priority to class 4 and lowest to class 2 ($4 \to 2$).

Policy 2: Give at station 1 highest priority to class 3 and lowest to class 1

$(3 \rightarrow 1)$ and give at station 2 highest priority to class 2 and lowest to class 4 $(2 \rightarrow 4)$.

with the one outperforming the other in some cases and vice versa. In the table, next to the performance of the best policy for each case, we are giving in parenthesis the policy identifier, denoting by p1 and p2, policy 1 and policy 2, respectively. Table 4 contains the data used for each case reported in Table 3. Note that by ρ_A, ρ_B we denote the total traffic intensities at station 1 and station 2, respectively.

Load Node 1-Node 2	Lower Bnd. 1	Lower Bnd. 2	FCFS	Best Policy	Effic.
HEAVY-HEAVY	42.24	45.36	70.55	65.58 (p2)	69%
MEDIUM-MEDIUM	16.07	20.07	28.83	27.88 (p1)	72%
MEDIUM-HEAVY	17.06	17.35	23.2	20.55 (p1)	85%
LIGHT-MEDIUM	3.44	3.69	5.23	5.00 (p1)	74%
HEAVY-MEDIUM	20.08	20.55	25.93	22.00 (p2)	94%
MEDIUM-LIGHT	4.25	4.56	5.56	5.29 (p1)	86%

Table 3: Numerical results for the network of Figure 4.

Load	ρ_A	ρ_B	λ	μ_1	μ_2
HEAVY-HEAVY	0.85	0.80	0.17	0.40	0.43
MEDIUM-MEDIUM	0.57	0.63	0.13	0.46	0.41
MEDIUM-HEAVY	0.6	0.9	0.5	1.67	1.12
LIGHT-MEDIUM	0.4	0.6	0.5	2.5	1.67
HEAVY-MEDIUM	0.9	0.6	0.5	1.12	1.67
MEDIUM-LIGHT	0.6	0.4	0.5	1.67	2.5

Table 4: Data for the experiments of Table 3.

The efficiency of our lower bound is again of approximately the same order of magnitude as the efficiency of the "pathwise bound" derived in [OuWe]. As we argued in the beginning of this section the efficiency of the bounds depends both on the their closeness to optimality and on the suboptimality of the upper bound. In order to understand which factor is more important we calculated the performance of the optimal policy for one specific case via dynamic programming. In particular, we applied the value iteration algorithm for the MEDIUM-MEDIUM traffic case. The dynamic programming algorithm yielded an optimal for the objective function of 27.7 proving policy p1 almost optimal.

9.3 A Six-Class Network Example

Consider the network depicted in figure 5. 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 1,2 have service times exponentially distributed with rates μ_1,μ_2 respectively.

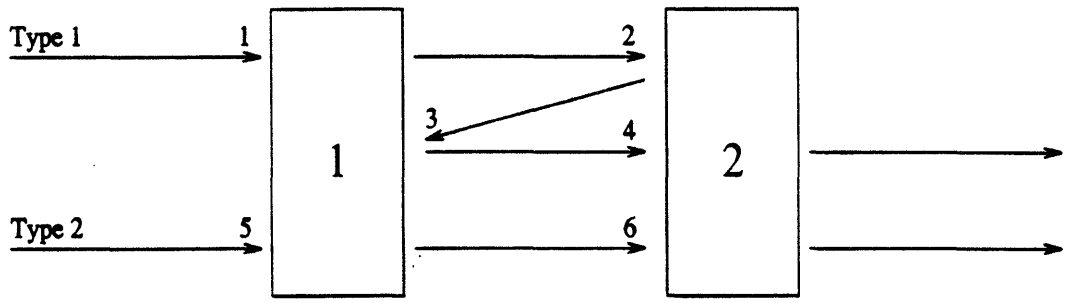

Figure 5: A Six-Class Network Example.

Table 5 compares our lower bounds on attainable performance with FCFS and the best found policy [1] for various load conditions, providing also the efficiency of the bound. "Lower Bnd. 1" and "Lower Bnd. 2" in the table correspond to the bound developed in Section 4.1 and Section 4.2, respectively. Costs throughout the experiments reported in the table were chosen to be:

$$c_1 = 1.5,\ c_2 = 1.3,\ c_3 = 1.2,\ c_4 = 1\ c_5 = 1.1,\ c_6 = 1.1.$$

In this specific example, also, the best policy we were able to find, for each load condition we considered, happens to be a strict priority one. Note that we only considered non-preemptive policies. It is interesting that not a single policy was optimal for every case we considered. More precisely the following two policies were competing:

Policy 1: Give at station 1 highest priority to class 3 and lowest to class 5 ($3 \rightarrow 1 \rightarrow 5$) and give at station 2 highest priority to class 6 and lowest to class

[1] we only considered nonpreemptive policies

2 ($6 \rightarrow 4 \rightarrow 2$).

Policy 2: Give at station 1 highest priority to class 1 and lowest to class 5 ($1 \rightarrow 3 \rightarrow 5$) and give at station 2 highest priority to class 2 and lowest to class 6 ($2 \rightarrow 4 \rightarrow 6$).

with the one outperforming the other in some cases and vice versa. In the table, next to the performance of the best policy for each case, we are giving in parenthesis the policy identifier, denoting by p1 and p2, policy 1 and policy 2, respectively. Table 6 contains the data used for each case reported in Table 5. Recall that by ρ_A, ρ_B we denote the total traffic intensities at station 1 and station 2, respectively.

Load Node 1-Node 2	Lower Bnd. 1	Lower Bnd. 2	FCFS	Best Policy	Effic.
HEAVY-HEAVY	15.72	16.67	30.56	26.89 (p2)	62%
MEDIUM-MEDIUM	5.83	6.17	9.86	9.25 (p2)	67%
MEDIUM-HEAVY	15.77	15.85	21.26	18.20 (p1)	87%
HEAVY-MEDIUM	18.77	18.79	23.00	19.80 (p1)	95%

Table 5: Numerical results for the network of Figure 5.

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

Table 6: Data for the experiments of Table 3.

9.4 Summary

Our computations results suggest:

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 4.4. In the more complicated examples with four and six classes it was

strictly better. The reason is that the nonparametric method takes more into account the interactions among various classes.

2. The efficiency of our lower bounds is approximately the same order of magnitude as the efficiency of the "pathwise bound" derived in [OuWe].

3. The bounds are very efficient in imbalanced traffic conditions. In these conditions the efficiency of the bounds increases with the traffic intensity. We believe that the reason for this is that in imbalanced conditions there is only one bottleneck, so the behavior of the system is dominated by only one station. But in single station systems our bounds are exact, which explains the tightness of our bounds.

4. In balanced traffic conditions, the bounds also behave well especially when the traffic intensity is not very close to one. But, even in these heavy-balanced traffic conditions, in the examples that we studied the efficiency does not get worse than 62%.

10 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 distributed 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 to illustrate especially the power of the higher order formulations.

References

[Al] Alizadeh F., (1992), "Combinatorial Optimization with Semi-Definite Matrices", 2nd conference in integer programming, Carnegie Mellon University, Proceedings, 385-405.

[BeNi] Bertsimas, D. and Niño-Mora J., (1992), "Conservation laws, extended polymatroids and the multi-armed bandit problem: a unified polyhedral approach", Operations Research Center, MIT, working paper.

[BPT1] Bertsimas, D., Paschalidis, I.Ch. and Tsitsiklis, J.N., "Scheduling of multiclass queueing networks: Bounds on achievable performance ", extended abstract, Proceedings of "Workshop on hierarchical control for real time scheduling of manufacturing systems, Lincoln, New Hampshire, October 16-18, 1992.

[BPT2] Bertsimas, D., Paschalidis, I.Ch. and Tsitsiklis, J.N., "Scheduling of multiclass queueing networks: Bounds on achievable performance", talk given at the ORSA/TIMS meeting conference, San Franscisco, November 2, 1992.

[BGT] Bhattacharya, P.P, Georgiadis, L., Tsoucas, P. (1992), "Extended Polymatroids: Properties and Optimization", *IBM Research Division*, Research Report, T.J Watson Research Center, Yorktown Heights, New York.

[ChYY] Chen, H., Yang, P. and Yao, D.D., (1991), "Control and Scheduling in a Two-Station Queueing Network: Optimal Policies and Heuristics", Preprint.

[FeGr] Federgruen, A. and Groenevelt, H., (1988), "Characterization and optimization of achievable performance in queueing systems", *Operations Research*, 36, 733-741.

[GeMi] Gelenbe, E. and Mitrani, I., (1980), *Analysis and Synthesis of Computer Systems*, Academic Press, London.

[Gi] Gittins, J.C., (1989), *Bandit Processes and Dynamic Allocation Indices*, John Wiley.

[Ha] Harrison, J.M., (1986), "Brownian Models of Queueing Networks with Heterogeneous Customers", Presented at the *IMA Workshop on Stochastic Differential Systems*, Mineapolis, MI, June 9–19.

[HaWe1] Harrison, J.M. and Wein, L.M., (1989), "Scheduling Networks of Queues: Heavy Traffic Analysis of a simple Open Network", *Queueing Systems Theory and Applications*, 5, 265–280.

[HaWe2] Harrison, J.M. and Wein, L.M., (1990), "Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network", *Operations Research*, 6, 1052-1064.

[KeLa] Kelly, F.P. and Laws, C.N., "Dynamic Routing in Open Queueing Networks", Preprint.

[Klv1] Kleinrock, L., (1975), *Queueing Systems, Vol. 1: Theory*, Wiley, New York.

[Klv2] Kleinrock, L., (1976), *Queueing Systems, Vol. 2: Computer Applications*, Wiley, New York.

[Klim] Klimov, G.P., (1974), "Time-Sharing Service Systems. I", *Theory of Probability and its Applications, Vol. XIX*, No 3.

[Kuma] Kumar, P.R., "Re-Entrant Lines", Proceedings of "Workshop on hierarchical control for real time scheduling of manufacturing systems, Lincoln, New Hampshire, October 16-18, 1992.

[LoSc] Lovasz L. and Schrijver A., (1991), "Cones of Matrices and Setfunctions, and 0-1 Optimization", *SIAM J. Optimization*, 1 (2).

[MeDo] Meyn, S.P and Down, D., (1991), "Stability of Generalized Jackson Networks", submitted for publication.

[NeWo] Nemhauser, G.L. and Wolsey, L.A. (1988), *Integer and Combinatorial Optimization*, Wiley, New York.

[OuWe] Ou, J. and Wein, L.M. (1992), "Performance Bounds for Scheduling Queueing Networks", *The Annals of Applied Probability*, Vol. 2, No. 2, 460–480.

[Pasc] Paschalidis, I.Ch. (1992), "Scheduling of Multiclass Queueing Networks: Bounds on Achievable Performance", Master of Science in Electrical Engineering and Computer Science Thesis, *Massachusetts Institute of Technology*.

[RoYa] Ross K. and Yao D. (1987), "Optimal dynamic scheduling in Jackson Networks", Preprint.

[ShYa] Shantikumar, J.G. and Yao, D.D., (1992), "Multiclass Queueing Systems: Polymatroid Structure and Optimal Scheduling Control", *Operations Research*, Vol. 40, No. 2, 293–299.

[Tsou] Tsoucas P., (1991), "The Region of Achievable Performance in a Model of Klimov", *IBM Research Division*, Research Report, T.J Watson Research Center, Yorktown Heights, New York.

[Wa] Walrand J., (1988) *An Introduction to Queueing Networks*, Prentice Hall.

[Wei1] Wein, L.M. (1990), "Optimal Control of a Two-Station Brownian Network", *Mathematics of Operations Research*, 15, 215-242.

[Wei2] Wein, L.M. (1990), "Scheduling Networks of Queues; Heavy traffic analysis of a two station network with controllable inputs", 38, 1065-1078.

[We] G. Weiss, (1988), "Branching bandit processes", *Probability in the Engineering and Information Sciences*, 2, 269-278.