

MIT Open Access Articles

*Solving convex optimization with side constraints
in a multi-class queue by adaptive $c\mu$ rule*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Li, Chih-ping, and Michael J. Neely. "Solving Convex Optimization with Side Constraints in a Multi-Class Queue by Adaptive $c\mu$ Rule." *Queueing Systems* 77.3 (2014): 331–372.

As Published: <http://dx.doi.org/10.1007/s11134-013-9377-3>

Publisher: Springer US

Persistent URL: <http://hdl.handle.net/1721.1/104074>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Solving convex optimization with side constraints in a multi-class queue by adaptive $c\mu$ rule

Chih-ping Li · Michael J. Neely

Received: date / Accepted: date

Abstract We study convex optimization problems with side constraints in a multi-class $M/G/1$ queue with controllable service rates. In the simplest problem of optimizing linear costs with fixed service rate, the $c\mu$ rule is known to be optimal. A natural question to ask is whether such simple policies exist for more complex control objectives. In this paper, combining the achievable region approach in queueing systems and the Lyapunov drift theory suitable to optimize renewal systems with time-average constraints, we show that convex optimization problems can be solved by variants of adaptive $c\mu$ rules. These policies greedily re-prioritize job classes at the end of busy periods in response to past observed delays in each job class. Our method transforms the original problems into a new set of queue stability problems, and the adaptive $c\mu$ rules are queue stable policies. An attractive feature of the adaptive $c\mu$ rules is that they use limited statistics of the queue, where no statistics are required for the problem of satisfying average queueing delay in each job class.

Keywords $c\mu$ -rule · Dynamic scheduling · Stochastic optimization · Lyapunov drift analysis · Strong conservation law · Achievable region method · Multi-class queueing systems · Polymatroid optimization

1 Introduction

We study the problems of serving jobs categorized into multiple classes in a queueing system, with the goals of optimizing a global objective and providing differenti-

This material is supported in part by: the Network Science Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory W911NF-09-2-0053, and the NSF Career grant CCF-0747525.

Chih-ping Li
Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139, USA,
E-mail: cpli@mit.edu

Michael J. Neely
Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA,
E-mail: mjneely@usc.edu

ated services. Such problems have attracted significant attention for decades due to their wide applications in computers, communication networks, and manufacturing systems. One useful solution method is the achievable region approach. That is, we first characterize the achievable region of a performance measure of interest, such as the set of all feasible delay vectors, then use optimization theory to develop optimal control policies (see [6, 7, 26, 37] for surveys). Many multi-class queueing systems, especially single-server queues, have performance regions of a special form: the base of a polymatroid [36]. This is a special polyhedron with the property that each of its vertices is the performance vector of a strict priority policy. One celebrated result is that minimizing linear costs, such as the average occupancy of all jobs in the queue, over a polymatroidal performance region is solved by the well known $c\mu$ rule [30]. A fundamental question we seek to answer is whether such simple policies exist for more complex control objectives, such as solving linear optimization with side constraints or convex optimization problems. In this paper, we show that constrained convex optimization in a multi-class $M/G/1$ queue with nonpreemptive service, whose performance region is the base of a polymatroid [11] (see details in Section 3), can be solved by adaptive online policies that employ a weighted $c\mu$ rule in every busy period.

We consider an $M/G/1$ queue serving N independent classes of Poisson arrivals. The controller serves jobs one at a time in a nonpreemptive fashion. After completing a job it makes a decision about which job class to serve next. We study four optimization problems. In the first two problems, we assume the queue has a fixed service rate, and consider:

1. Designing a policy that satisfies an average queueing delay constraint $\bar{W}_n \leq d_n$ for each job class $n \in \{1, \dots, N\}$, assuming all constraints are feasible.
2. Developing a policy that minimizes a separable convex function $\sum_{n=1}^N f_n(\bar{W}_n)$ of average queueing delays, subject to delay constraints $\bar{W}_n \leq d_n$ for all job classes.

In the third and fourth problem, we assume the queue has an adjustable service rate $\mu(P(t))$, incurring an instantaneous cost $P(t)$ at time t . We restrict attention to the simple rate control scheme that the service rate stays fixed in each busy period, but may be changed at the end of busy periods. Under this assumption, we consider two scheduling and service rate control problems:

3. Minimizing the average cost of service rate allocations subject to delay constraints $\bar{W}_n \leq d_n$ for all job classes.
4. Minimizing a separable convex delay function $\sum_{n=1}^N f_n(\bar{W}_n)$ subject to a constraint on the average cost of service rate allocations.

The above four problems are presented in this paper with increasing complexity so that the readers can gradually familiarize themselves with the new methodology we use to solve them. The first problem seeks to provide average delay guarantees to each job class. A motivation for the second problem is to provide some notion of delay fairness across job classes; we formulate it as a convex optimization problem. In particular, in Section 5.1 we introduce the notion of *delay proportional fairness*, in the same spirit as the well-known rate proportional fairness [20] or utility proportional fairness [35], and show that the corresponding objective functions f_n are

quadratic (rather than logarithmic functions that correspond to rate proportional fairness in network utility maximization problems). The potential applications of the third and fourth problem are dynamic power allocations in computer systems. Modern CPUs have the capability of adjusting operation frequencies to conserve power when the loading is low [2], resulting in low electricity cost. In this context, the third problem investigates how to schedule jobs to minimize power cost while providing delay guarantees to different traffic streams. The fourth problem studies, under a budget on power cost, how to fairly allocate the resources to job classes.

To design simple control policies with near-optimal performance, we look at the special structure of the performance region of the $M/G/1$ queue. Since it is the base of a polymatroid with vertices achieved by strict priority policies, every feasible performance vector, including the solutions to the first two delay optimization problems, can be achieved by a randomized policy that updates priorities over busy periods according to a stationary distribution. We are thus motivated to find an optimal online policy that updates priorities over busy periods. In the last two problems with service rate control, we find the optimal online policies that update priorities and service rates over busy periods.¹

The main contribution of this paper is to develop online priority and service rate control policies that solve the four problems. Our construction of the policies is based on using virtual queues (or counters) to monitor, in each job class, the amount of past observed delays violating the delay constraints, stored as virtual queue backlogs. Then, at decision epochs (i.e., the end of busy periods), job classes with more severe delay violations are offered higher priorities until the next decision epoch, and so on. Technically, using Lyapunov drift analysis [15, 25], we show that policies stabilizing the virtual queues are online policies solving the optimization problems in the $M/G/1$ queue. These policies make a “max-weight” decision [32, 33] in every busy period, where the decisions turn out to be the $c\mu$ rule assigning priorities by sorting weighted virtual queue backlogs. In particular, for the last two problems with service rate control, we require a generalized “ratio max-weight” principle [22, 25] to find the optimal priority assignment in every busy period, because the size of a busy period is a function of the service rate. We show that the resulting dynamic $c\mu$ rules satisfy the average delay constraints in the first problem, and yield performance that is $O(1/V)$ away from optimal in the other three problems, where $V > 0$ is a control parameter that can be chosen sufficiently large for optimality, with a tradeoff in convergence time of the algorithms. Our policies are developed without pre-computing the offline solutions to the four optimization problems, and require limited statistical knowledge of the queue. Surprisingly, no queue statistics are required for the first problem of providing average delay guarantees. The use of Lyapunov drift in this context is novel, as conventional max-weight techniques cannot optimize network delay [25].

¹ In principle, we can solve the first two delay optimization problems offline to obtain the optimal delay vector, and then develop the corresponding randomized policy. When the number of job classes is large, however, this offline approach is prohibitive because finding the optimal mixing of strict priority policies requires solving a linear system of $N!$ variables (there are $N!$ strict priority policies). The last two problems we study are even more complicated due to adjustable service rates. In addition, finding the optimal randomized policies offline requires the statistical knowledge of traffic streams such as arrival rates and the first two moments of job sizes. The adaptive policies we develop in this paper minimize the use of such statistics, and therefore can tolerate inaccurate estimations of these statistics.

In the literature, the $c\mu$ rule is known to be an optimal scheduling policy in many contexts, e.g., [3, 9] and [34, Chapter 8]. Linear optimization with side constraints over the base of a polymatroid is studied in [27, 28]. Offline numerical methods that solve the minimization of a separable convex function over the base of a polymatroid are proposed in [10]. Work [8] studies convex optimization over a multi-class $M/G/1$ queue with Bernoulli feedback, and develops an adaptive priority policy. This policy, based on stochastic approximation, has design philosophy similar to ours. A related problem to those studied in this paper is that of minimizing convex holding costs in queueing systems (see [1, 16] for a restless bandit formulation and [17, 23, 24] for showing that a generalized $c\mu$ rule is asymptotically optimal under heavy traffic). In these studies, a convex penalty is taken as a function of instantaneous queue occupancy, where in this paper we consider a convex penalty as a function of average delay. Convex optimization with side constraints over the base of a polymatroid is also studied in the context of real-time scheduling in wireless networks [18, 19]. State-dependent allocation of service rates in a single-server queue is addressed in [14, 31] and references therein. The usual approach uses dynamic programming methods to show the monotonic structure of optimal policies.

The outline of this paper is as follows. Section 2 describes the detailed queueing model. Section 3 summarizes useful properties of a multi-class $M/G/1$ queue that will be used in this paper. The four optimization problems are solved in Sections 4-7, followed by simulation results in Section 8. Section 9 provides proofs of main results in the paper.

2 Queueing model

We only consider queueing delay, not system delay (queueing plus service), in this paper. System delay can be easily incorporated because, in a nonpreemptive queue, average queueing and system delay differ by the mean service time. We use *delay* and *queueing delay* interchangeably in the rest of the paper.

Consider an $M/G/1$ queue serving N classes of jobs. Jobs in a class $n \in \{1, \dots, N\}$ arrive as an independent Poisson process with rate $\lambda_n > 0$. All job sizes are independent across classes, and are independent and identically distributed (i.i.d.) within each class. Let the random variable S_n denote the size of a class n job, with mean job size $\mathbb{E}[S_n]$. As a technical detail, we assume the first four moments of S_n are finite for all classes n ; the distribution of S_n is otherwise arbitrary. When a job arrives, we only know its class but not its actual size. The server has instantaneous service rate $\mu(P(t))$, where $P(t)$ is the service cost at time t . Assume $\mu(\cdot)$ is an increasing function with $\mu(0) = 0$. We regard the queue as a frame-based system, where each frame consists of an idle period and the following busy period. For $k \in \{0, 1, 2, \dots\}$, let t_k be the start of the k th frame, and the k th frame is $[t_k, t_{k+1})$. Define $t_0 = 0$ and the queue is initially empty. Let $T_k \triangleq t_{k+1} - t_k$ be the size of frame k . Let $A_{n,k}$ be the set of class n arrivals in frame k . For each job $i \in A_{n,k}$, let $W_{n,k}^{(i)}$ be its queueing delay.

We consider scheduling policies that are work conserving, non-anticipative, non-preemptive, and independent of actual job sizes (recall that job sizes are unknown upon arrival). Jobs in each class are served according to first-in-first-out (FIFO).

Scheduling policies that satisfy these properties are referred to as *admissible* policies. When the service rate is controllable, we focus on control decisions that allocate a fixed service rate $\mu(P_k)$ in the k th busy period, with an instantaneous cost $P_k \in [P_{\min}, P_{\max}]$; the decisions are possibly random. Zero service rates are allocated in idle periods with zero cost. Suppose the maximum cost P_{\max} is finite, but sufficiently large to ensure feasibility of the desired delay constraints. The minimum cost P_{\min} is chosen to be large enough so that the queue is stable even if P_{\min} is used for all time. That is, for queue stability we need $\sum_{n=1}^N \lambda_n \frac{\mathbb{E}[S_n]}{\mu(P_{\min})} < 1$, i.e., $\mu(P_{\min}) > \sum_{n=1}^N \lambda_n \mathbb{E}[S_n]$.

The average delay under our policies may not have well-defined limits. For this, we define the average queuing delay of job class n as

$$\bar{W}_n \triangleq \limsup_{K \rightarrow \infty} \frac{\mathbb{E} \left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} \right]}{\mathbb{E} \left[\sum_{k=0}^{K-1} |A_{n,k}| \right]}, \quad (1)$$

where $|A_{n,k}|$ is the number of class n arrivals in frame k . We only consider average delays sampled at frame boundaries for simplicity. To verify (1), the running average delay of class n jobs at time t_K is equal to

$$\frac{\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}}{\sum_{k=0}^{K-1} |A_{n,k}|} = \frac{\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}}{\frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}|}.$$

We define two averages

$$\bar{w}_n \triangleq \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}, \quad \bar{a}_n \triangleq \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}|.$$

If both limits \bar{w}_n and \bar{a}_n exist with probability 1, then the ratio \bar{w}_n/\bar{a}_n is the limiting average delay for class n . In this case, we get

$$\begin{aligned} \bar{W}_n &= \frac{\lim_{K \rightarrow \infty} \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} \right]}{\lim_{K \rightarrow \infty} \mathbb{E} \left[\frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}| \right]} \\ &= \frac{\mathbb{E} \left[\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} \right]}{\mathbb{E} \left[\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}| \right]} = \frac{\bar{w}_n}{\bar{a}_n}, \end{aligned} \quad (2)$$

which shows that \bar{W}_n defined by (1) is indeed the limiting average delay.²

² The second equality in (2), where we pass limits into expectations, can be proved by a generalized Lebesgue's dominated convergence theorem [12, Exercise 20 in Sec. 2.3] stated as follows. Let $\{X_n\}_{n=1}^{\infty}$ and $\{Y_n\}_{n=1}^{\infty}$ be two sequences of random variables such that: (1) $0 \leq |X_n| \leq Y_n$ with probability 1 for all n ; (2) for some random variables X and Y , $X_n \rightarrow X$ and $Y_n \rightarrow Y$ with probability 1; (3) $\lim_{n \rightarrow \infty} \mathbb{E}[Y_n] = \mathbb{E}[Y] < \infty$. Then $\mathbb{E}[X]$ is finite and $\lim_{n \rightarrow \infty} \mathbb{E}[X_n] = \mathbb{E}[X]$. The details are omitted for brevity.

3 Preliminaries

We present useful properties of a multi-class $M/G/1$ queue with nonpreemptive service. Here, we assume a constant service rate $\mu(P)$ with a fixed cost P (this is extended in Section 6). Let $X_n \triangleq S_n/\mu(P)$ be the service time of a class n job. Define $\rho_n \triangleq \lambda_n \mathbb{E}[X_n]$. Fix an arrival rate vector $(\lambda_n)_{n=1}^N$ satisfying $\sum_{n=1}^N \rho_n < 1$; the rate vector $(\lambda_n)_{n=1}^N$ is supportable in the queue.

Let I_k and B_k , $k \in \{0, 1, 2, \dots\}$, be the k th idle and busy period, respectively. The size of the k th frame is $T_k = I_k + B_k$. The distributions of B_k and T_k are fixed under any work-conserving policy when the service rate is fixed. This is because the sample path of unfinished work in the system always decreases at the processing rate of the server and has a jump when a job arrives, regardless of the order jobs are served. Since Poisson arrivals are memoryless, we have $\mathbb{E}[I_k] = 1/(\sum_{n=1}^N \lambda_n)$ for all k . For the same reason, the $M/G/1$ queue renews itself at the start of every frame (i.e., at the start of every idle period). Consequently, the frame size T_k , busy period B_k , and the number $|A_{n,k}|$ of class n arrivals in a frame are all i.i.d. over k . Using renewal reward theory [29] with renewal epochs defined at frame boundaries $\{t_k\}_{k=0}^\infty$, we have for all $k \in \{0, 1, 2, \dots\}$:

$$\mathbb{E}[T_k] = \frac{\mathbb{E}[I_k]}{1 - \sum_{n=1}^N \rho_n} = \frac{1}{(1 - \sum_{n=1}^N \rho_n) \sum_{n=1}^N \lambda_n}, \quad (3)$$

$$\mathbb{E}[|A_{n,k}|] = \lambda_n \mathbb{E}[T_k], \quad n \in \{1, \dots, N\}. \quad (4)$$

We define the delay performance region \mathscr{W} of the $M/G/1$ queue as the set of average queueing delay vectors under a collection Π of admissible scheduling policies whose decisions are based on the history of the system since the last time the queue is empty (in other words, scheduling decisions are stationary and independent across busy periods). The delay region \mathscr{W} is studied in [13, Theorems 8.3 and 8.5] and [11, Theorem 2] and presented next.

Lemma 1 (Theorem 2 in [11]) *Let \bar{W}_n^π be the average queueing delay of class n jobs under a policy $\pi \in \Pi$. Define the delay performance region $\mathscr{W} = \{(\bar{W}_1^\pi, \dots, \bar{W}_N^\pi) \mid \pi \in \Pi\}$. Let $x_n \triangleq \rho_n \bar{W}_n$ be the average unfinished work in the queue for class n , and define $\Omega = \{(x_n)_{n=1}^N \mid (\bar{W}_n)_{n=1}^N \in \mathscr{W}\}$. Then*

$$\Omega = \left\{ (x_n)_{n=1}^N \mid \sum_{n=1}^N x_n = \frac{R\rho}{1-\rho}, \quad \sum_{n \in A} x_n \geq \frac{R\rho_A}{1-\rho_A} \quad \forall A \subset \{1, \dots, N\} \right\}, \quad (5)$$

where $R = \frac{1}{2} \sum_{n=1}^N \lambda_n \mathbb{E}[X_n^2]$, $\rho = \sum_{n=1}^N \rho_n$, and $\rho_A = \sum_{n \in A} \rho_n$. The set Ω is the base of a polymatroid, which has the properties: (1) every vertex of Ω is the performance vector of a strict priority policy; (2) the performance vector of each strict priority policy is a vertex of Ω .

Note that there is a one-to-one mapping between the set Ω and the delay region \mathscr{W} via a simple scaling. Lemma 1 says that there is a one-to-one correspondence between the vertices of Ω and the set of strict priority policies. Thus, every feasible vector

$(x_n)_{n=1}^N \in \Omega$ (i.e., every feasible delay vector $(\bar{W}_n)_{n=1}^N \in \mathcal{W}$) is attained by a randomization of strict priority policies. Such randomization can be implemented across busy periods because the $M/G/1$ queue renews itself at the end of busy periods.

Optimizing a linear cost function over the base of a polymatroid is useful in later analysis; the solution is the well known $c\mu$ rule [37, Theorem 3].

Lemma 2 (The $c\mu$ rule, Corollary 1 in [11]) *In a multi-class $M/G/1$ queue with nonpreemptive service, consider the linear program*

$$\text{minimize } \sum_{n=1}^N c_n x_n \quad (6)$$

$$\text{subject to } (x_n)_{n=1}^N \in \Omega \quad (7)$$

where Ω is defined in (5) and $c_n \geq 0$ are constants. Assume $\sum_{n=1}^N \rho_n < 1$ for stability and that the service time of a class n job has a finite second moment $\mathbb{E}[X_n^2] < \infty$ for all n . The optimal solution to (6)-(7) is achieved by a strict priority policy that prioritizes job classes in the decreasing order of c_n . That is, if $c_1 \geq c_2 \geq \dots \geq c_N$, then class 1 gets the highest priority, class 2 gets the second highest priority, and so on. In this case, the optimal average queueing delay \bar{W}_n^* of job class n is [5, Section 3.5.3]

$$\bar{W}_n^* = \frac{R}{(1 - \sum_{k=0}^{n-1} \rho_k)(1 - \sum_{k=0}^n \rho_k)}, \quad (8)$$

where $\rho_0 \triangleq 0$ and $R \triangleq \frac{1}{2} \sum_{n=1}^N \lambda_n \mathbb{E}[X_n^2]$.

Lemma 2 shows that minimizing the weighted sum of average queue occupancy of all job classes, i.e., minimizing $\sum_{n=1}^N c_n \lambda_n \bar{W}_n = \sum_{n=1}^N (c_n \mu_n) x_n$ where $\mu_n = 1/\mathbb{E}[X_n]$ and $x_n = \rho_n \bar{W}_n$, is achieved by prioritizing job classes in the decreasing order of $c_n \mu_n$. Hence Lemma 2 is called the $c\mu$ rule.

4 First problem: Achieving per-class average delay

We design a dynamic scheduling policy that yields average queueing delays satisfying $\bar{W}_n \leq d_n$ for all classes n . In this problem, we assume the queue has a fixed service rate and that all delay constraints are feasible.

Our main idea is to track the running delay performance in each job class, and use this information to re-prioritize job classes at the end of busy periods. For this, we define a discrete-time *virtual delay queue* $\{Z_{n,k}\}_{k=0}^{\infty}$ for each class $n \in \{1, \dots, N\}$, where $Z_{n,k}$ is the virtual backlog at time t_k (i.e., at the start of the k th frame). Assume initially $Z_{n,0} = 0$ for all classes n . Each queue $Z_{n,k}$ is updated at time instants $\{t_k, k \in \mathbb{Z}^+\}$ according to

$$Z_{n,k+1} = \max \left[Z_{n,k} + \sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n), 0 \right], \quad (9)$$

where $W_{n,k}^{(i)}$ is the queueing delay of the i th class n job served in the k th frame $[t_k, t_{k+1})$. The value of $Z_{n,k}$ tracks the amount of observed queueing delays of class n jobs

exceeding the desired delay bound d_n . Thus, $Z_{n,k}$ can be viewed as the *debt* the queue controller owes to job class n to satisfy its delay requirement.

In (9), we can view $W_{n,k}^{(i)}$ and d_n as arrivals and service opportunities of the virtual queue $Z_{n,k}$, respectively. The next lemma shows that the stability of the queue $Z_{n,k}$ is a sufficient condition to satisfy the constraint $\bar{W}_n \leq d_n$.

Definition 1 Queue $Z_{n,k}$ is called *mean rate stable* if $\lim_{K \rightarrow \infty} \mathbb{E}[Z_{n,K}]/K = 0$.

Lemma 3 If queue $Z_{n,k}$ is mean rate stable, then $\bar{W}_n \leq d_n$.

Proof (Lemma 3) From (9), we get

$$Z_{n,k+1} \geq Z_{n,k} - d_n |A_{n,k}| + \sum_{i \in A_{n,k}} W_{n,k}^{(i)}.$$

Summing the above over $k \in \{0, \dots, K-1\}$ for some integer K , using $Z_{n,0} = 0$, and taking expectation, we have

$$\mathbb{E}[Z_{n,K}] \geq -d_n \mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right] + \mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right].$$

Dividing the above by $\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]$ yields

$$\frac{\mathbb{E}[Z_{n,K}]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]} \geq \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]} - d_n.$$

Taking a lim sup as $K \rightarrow \infty$ and using the delay definition (1), we have

$$\bar{W}_n \leq d_n + \limsup_{K \rightarrow \infty} \frac{\mathbb{E}[Z_{n,K}]}{K} \frac{K}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]}.$$

Using the inequality $\mathbb{E}[|A_{n,k}|] = \lambda_n \mathbb{E}[T_k] \geq \lambda_n \mathbb{E}[I_k] = \lambda_n \mathbb{E}[I_0]$ and mean rate stability of queue $Z_{n,k}$, we obtain

$$\bar{W}_n \leq d_n + \frac{1}{\lambda_n \mathbb{E}[I_0]} \lim_{K \rightarrow \infty} \frac{\mathbb{E}[Z_{n,K}]}{K} = d_n. \quad \square$$

By Lemma 3, we transform the first delay optimization problem into a queue stability problem over virtual queues $(Z_{1,k}, \dots, Z_{N,k})$.

4.1 The control policy

The following policy stabilizes all virtual queues $(Z_{1,k}, \dots, Z_{N,k})$ and satisfies all delay constraints $\bar{W}_n \leq d_n$ by Lemma 3.

Delay Feasible Policy (DelayFeas):

- In the k th busy period, serve jobs by prioritizing job classes in the decreasing order of $Z_{n,k}$ (i.e., the job class with the larger $Z_{n,k}$ has the higher priority); ties are broken arbitrarily.
- Update $Z_{n,k}$ according to (9) at the end of busy periods.

As explained earlier, $Z_{n,k}$ represents the *delay debt* owed to class n jobs. The DelayFeas policy gives higher priority to job classes with larger debts in every busy period. Notice that this policy requires no statistical knowledge of the queue.

Theorem 1 (Proof in Section 9.2) *If the delay requirements $\{d_1, \dots, d_N\}$ are feasible, then the DelayFeas policy yields average delays satisfying $\bar{W}_n \leq d_n$ for all job classes $n \in \{1, \dots, N\}$.*

The convergence time of the DelayFeas policy reflects how soon the running average delay in each job class n is below the desired value d_n . By Lemma 3, the speed of the ratio $\mathbb{E}[Z_{n,K}]/K$ approaching zero gives us a good intuition. According to (74) in the proof of Theorem 1, we expect the DelayFeas policy to converge with speed $O(1/\sqrt{K})$, where K is the number of passed busy periods. The control policies developed for the next three problems have similar convergence time.

4.2 Construction of the DelayFeas policy

The construction of the DelayFeas policy follows a Lyapunov drift argument on the virtual queues $(Z_{1,k}, \dots, Z_{N,k})$. Define the weighted quadratic Lyapunov function

$$L(\mathbf{Z}_k) \triangleq \frac{1}{2} \sum_{n=1}^N \mathbb{E}[X_n] (Z_{n,k})^2$$

as a scalar measure of the virtual backlog vector $\mathbf{Z}_k \triangleq (Z_{n,k})_{n=1}^N$, where $\mathbb{E}[X_n]$ is the mean service time of class n jobs. Define the one-frame Lyapunov drift

$$\Delta(\mathbf{Z}_k) \triangleq \mathbb{E}[L(\mathbf{Z}_{k+1}) - L(\mathbf{Z}_k) \mid \mathbf{Z}_k]$$

as the conditional expected growth of the measure $L(\mathbf{Z}_k)$ over the k th frame. We show that the DelayFeas policy minimizes the Lyapunov drift $\Delta(\mathbf{Z}_k)$ in every frame. This will be useful for proving the DelayFeas policy stabilizes all virtual queues $(Z_{1,k}, \dots, Z_{N,k})$. We square (9) and use $(\max[a, 0])^2 \leq a^2$ to yield

$$(Z_{n,k+1})^2 \leq \left[Z_{n,k} + \sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \right]^2. \quad (10)$$

Multiplying (10) by $\mathbb{E}[X_n]/2$, summing over $n \in \{1, \dots, N\}$, and taking conditional expectation on \mathbf{Z}_k , we get

$$\begin{aligned} \Delta(\mathbf{Z}_k) &\leq \frac{1}{2} \sum_{n=1}^N \mathbb{E}[X_n] \mathbb{E} \left[\left(\sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \right)^2 \mid \mathbf{Z}_k \right] \\ &\quad + \sum_{n=1}^N \mathbb{E}[X_n] Z_{n,k} \mathbb{E} \left[\sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \mid \mathbf{Z}_k \right]. \end{aligned} \quad (11)$$

Lemma 7 in Section 9.1 shows that the second term in (11) is bounded by a finite constant $C > 0$. As a result, we have

$$\begin{aligned} \Delta(\mathbf{Z}_k) &\leq C + \sum_{n=1}^N \mathbb{E}[X_n] Z_{n,k} \mathbb{E} \left[\sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \mid \mathbf{Z}_k \right] \\ &= (C - \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \rho_n d_n) + \sum_{n=1}^N \mathbb{E}[X_n] Z_{n,k} \mathbb{E} \left[\sum_{i \in A_{n,k}} W_{n,k}^{(i)} \mid \mathbf{Z}_k \right], \end{aligned} \quad (12)$$

where the equality uses (4). We are interested in the admissible scheduling policy that observes the value of \mathbf{Z}_k at the beginning of the k th frame, and minimizes the right-hand side of (12) over that frame, for all $k \in \mathbb{Z}^+$. Since the service rate is fixed, $\mathbb{E}[T_k]$ is fixed and our desired policy minimizes the last sum of (12). (We remark that, given the value of \mathbf{Z}_k , minimizing the last sum of (12) does not depend on the system history prior to the k th frame.)

We show that the DelayFeas policy minimizes the last sum of (12). Let $Q_n(t)$ be the number of class n jobs in the queue (not including that in the server) at time t . Using a sample-path argument (e.g., [5, Figure 3.1]), it is easy to see

$$\sum_{i \in A_{n,k}} W_{n,k}^{(i)} = \int_{t_k}^{t_{k+1}} Q_n(t) dt, \quad (13)$$

where we recall that $W_{n,k}^{(i)}$ is only the queueing delay (not including service time). We define $\bar{Q}_{n,k}$ as the average occupancy of class n jobs in the queue *if the scheduling decisions made in the k th frame are used independently in all frames*. Let $\bar{W}_{n,k}$ be the associated average queueing delay for class n , satisfying $\bar{Q}_{n,k} = \lambda_n \bar{W}_{n,k}$ by Little's Theorem. We have

$$\frac{\mathbb{E} \left[\sum_{i \in A_{n,k}} W_{n,k}^{(i)} \mid \mathbf{Z}_k \right]}{\mathbb{E}[T_k]} = \frac{\mathbb{E} \left[\int_{t_k}^{t_{k+1}} Q_n(t) dt \mid \mathbf{Z}_k \right]}{\mathbb{E}[T_k]} = \bar{Q}_{n,k} = \lambda_n \bar{W}_{n,k}, \quad n \in \{1, \dots, N\}, \quad (14)$$

where the first equality uses (13), and the second equality uses renewal reward theory [29, Theorem 3.6.1]. From (14), the last sum of (12) is

$$\sum_{n=1}^N \mathbb{E}[X_n] Z_{n,k} \mathbb{E} \left[\sum_{i \in A_{n,k}} W_{n,k}^{(i)} \mid \mathbf{Z}_k \right] = \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \rho_n \bar{W}_{n,k}. \quad (15)$$

Now, minimizing the left-hand side of (15) over all feasible scheduling decisions in the k th frame is equivalent to minimizing the right-hand side of (15) over the set Π of admissible scheduling policies that make independent scheduling decisions over frames, i.e., over the delay performance region \mathscr{W} defined in Lemma 1. Since $\mathbb{E}[T_k]$ is fixed, this is equivalent to solving

$$\text{minimize } \sum_{n=1}^N Z_{n,k} x_n, \quad \text{subject to } (x_1, \dots, x_N) \in \Omega, \quad (16)$$

where $x_n \triangleq \rho_n \bar{W}_{n,k}$ and Ω is given in (5). By Lemma 2, (16) is solved by the $c\mu$ rule that assigns strict priorities to job classes in the decreasing order of $Z_{n,k}$ in the k th frame. This is the DelayFeas policy.

We remark that the value of mean service time $\mathbb{E}[X_n]$ is only used in the analysis constructing the DelayFeas policy, and the policy itself does not need it. Using (15), we can re-write (12) as

$$\Delta(\mathbf{Z}_k) \leq \left(C - \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \rho_n d_n \right) + \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \rho_n \bar{W}_{n,k}, \quad (17)$$

which is useful in later performance analysis. For readers who are familiar with the use of Lyapunov drift analysis to design queue-stable policies in stochastic networks, the DelayFeas policy is the *max-weight* scheduling policy in this context, and our analysis in this section is in the similar spirit to those in [32, 33].

5 Second problem: Optimizing convex functions with side constraints

Consider the convex optimization problem with side constraints:

$$\text{minimize } \sum_{n=1}^N f_n(\bar{W}_n) \quad (18)$$

$$\text{subject to } \bar{W}_n \leq d_n, \quad n \in \{1, \dots, N\}, \quad (19)$$

$$(\bar{W}_1, \dots, \bar{W}_N) \in \mathscr{W}, \quad (20)$$

where \mathscr{W} is the delay performance region defined in Lemma 1. The penalty functions f_n are assumed to be continuous, convex, nondecreasing, and nonnegative for all job classes n . In this problem, we assume the queue has a fixed service rate and that the delay constraints (19) are feasible. We aim to design a control policy that solves (18)-(20).

5.1 Delay proportional fairness

One delay penalty function f_n of interest is the one that attains *proportional fairness*. A delay vector $(\bar{W}_n^*)_{n=1}^N$ is called *delay proportional fair* over the delay performance

region \mathcal{W} if it is the optimal solution under quadratic penalty functions $f_n(\bar{W}_n) = \frac{1}{2} c_n (\bar{W}_n)^2$ for all job classes n , where $c_n > 0$ are coefficients. In other words,

$$(\bar{W}_n^*)_{n=1}^N \in \operatorname{argmin}_{(\bar{W}_n)_{n=1}^N \in \mathcal{W}} \frac{1}{2} \sum_{n=1}^N c_n (\bar{W}_n)^2.$$

In this case, any feasible delay vector $(\bar{W}_n)_{n=1}^N \in \mathcal{W}$ satisfies the first-order optimality condition [4, Proposition 2.1.2]

$$\sum_{n=1}^N f'_n(\bar{W}_n^*) (\bar{W}_n - \bar{W}_n^*) = \sum_{n=1}^N c_n (\bar{W}_n - \bar{W}_n^*) \bar{W}_n^* \geq 0, \quad (21)$$

which is in the same spirit as the *rate (throughput) proportional fair* [20] criterion

$$\sum_{n=1}^N c_n \frac{x_n - x_n^*}{x_n^*} \leq 0 \quad (22)$$

in network utility maximization problems, where $(x_n)_{n=1}^N$ is any feasible throughput vector of the network users, and $(x_n^*)_{n=1}^N$ is the optimal throughput vector.

Intuitively, delay proportional fairness is associated with the product form criterion (21) instead of the ratio form (22) because we desire large throughput but favor small delay. To further clarify, we give a two-user example showing the two criteria (21) and (22) provide the same proportional tradeoff. Let $c_1 = c_2 = 1$. In a network utility maximization problem with the goal of providing fair throughput to the users, we suppose $(x_1^*, x_2^*) = (20, 2)$ is the rate-proportional-fair throughput vector. The performance of user 1 is $20/2 = 10$ times better than that of user 2. Consider any feasible deviation from the fair point (x_1^*, x_2^*) , say we increase Δy units of throughput for user 1. The criterion (22) shows that such deviation would incur more than $\Delta y/10$ units of throughput loss for user 2—this is considered unfair because the proportional performance loss of user 2 is larger than the proportional gain of user 1. In delay minimization problems, let $(\bar{W}_1^*, \bar{W}_2^*) = (3, 30)$ be the optimal delay vector that achieves delay proportional fairness. Again the performance of user 1 is 10 times better than that of user 2. According to (21), improving user 1 delay by Δy units would incur more than $\Delta y/10$ units of delay increase for user 2, which is proportionally unfair in the same spirit.

5.2 The control policy

Directly optimizing the penalty function $\sum_{n=1}^N f_n(\bar{W}_n)$ is difficult. We bypass this difficulty by formulating an equivalent optimization problem that uses auxiliary control variables (r_1, \dots, r_N) :

$$\text{minimize} \quad \sum_{n=1}^N f_n(r_n) \quad (23)$$

$$\text{subject to} \quad \bar{W}_n \leq d_n, \quad n \in \{1, \dots, N\} \quad (24)$$

$$\bar{W}_n \leq r_n, \quad r_n \in [0, d_n], \quad n \in \{1, \dots, N\} \quad (25)$$

$$(\bar{W}_1, \dots, \bar{W}_N) \in \mathcal{W}. \quad (26)$$

The next lemma shows (23)-(26) is equivalent to (18)-(20).

Lemma 4 *Let f_W^* and f_r^* be the optimal objective value of the two problems (18)-(20) and (23)-(26), respectively. Then $f_W^* = f_r^*$.*

Proof (Lemma 4) Let $(r_n, \bar{W}_n)_{n=1}^N$ be a feasible solution of (23)-(26). Then $(\bar{W}_n)_{n=1}^N$ is a feasible solution of (18)-(20) satisfying $\bar{W}_n \leq r_n$ for all classes n . Since f_n are nondecreasing, we have $f_W^* \leq \sum_{n=1}^N f_n(\bar{W}_n) \leq \sum_{n=1}^N f_n(r_n)$, which holds for all feasible choices of $(r_n)_{n=1}^N$ in (23)-(26). As a result, $f_W^* \leq f_r^*$.

Conversely, let $(\bar{W}_n)_{n=1}^N$ be a feasible solution of (18)-(20). Then the vector $(r_n, \bar{W}_n)_{n=1}^N$ with $r_n = \bar{W}_n$ for all classes n is a feasible solution of (23)-(26). It follows that $f_r^* \leq \sum_{n=1}^N f_n(r_n) = \sum_{n=1}^N f_n(\bar{W}_n)$, which holds for all feasible solutions $(\bar{W}_n)_{n=1}^N$ of (18)-(20). Thus $f_r^* \leq f_W^*$. We conclude $f_W^* = f_r^*$. \square

Our control policy, to be given shortly, solves (23)-(26) using the following ideas.

1. We use the same virtual queues $(Z_{1,k}, \dots, Z_{N,k})$ as in the first problem. The stability of the queues $Z_{n,k}$ attains the delay requirements (24).
2. We construct a new virtual queue $\{Y_{n,k}\}_{k=0}^\infty$ for each job class n , where $Y_{n,k+1}$ is computed at time t_{k+1} by

$$Y_{n,k+1} = \max \left[Y_{n,k} + \sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - r_{n,k}), 0 \right]. \quad (27)$$

The only difference between the virtual queues $Z_{n,k}$ and $Y_{n,k}$ is the new variable $r_{n,k} \in [0, d_n]$ chosen at time t_k . Assume initially $Y_{n,0} = 0$ for all n . We will use the stability of the virtual queues $(Y_{1,k}, \dots, Y_{N,k})$ to enforce the constraints (25).

3. We can regard the auxiliary variable r_n as the average service rate of the virtual queue $Y_{n,k}$. Then, optimizing the objective function (23) is equivalent to minimizing a separable convex function of the average service rates of the virtual queues $(Y_{1,k}, \dots, Y_{N,k})$.

The following policy solves (23)-(26).

Delay Fairness Policy (DelayFair):

- In the k th busy period, serve jobs by prioritizing job classes in the decreasing order of the ratio $(Z_{n,k} + Y_{n,k})/\mathbb{E}[S_n]$, where $\mathbb{E}[S_n]$ is the mean job size of class n ; ties are broken arbitrarily.
- At the end of the k th busy period, compute $Z_{n,k+1}$ and $Y_{n,k+1}$ for each job class n according to (9) and (27), respectively, where $r_{n,k}$ is the solution to the one-variable convex program:

$$\text{minimize} \quad V f_n(r_{n,k}) - Y_{n,k} \lambda_n r_{n,k} \quad (28)$$

$$\text{subject to} \quad 0 \leq r_{n,k} \leq d_n \quad (29)$$

where $V > 0$ is a predefined control parameter.

The DelayFair policy computes $r_{n,k}$ at the beginning of the k th frame, independent of the frame size T_k and the set of class n arrivals $A_{n,k}$ in that frame. This policy uses the knowledge of arrival rates λ_n and mean job sizes $\mathbb{E}[S_n]$, but not higher-order statistics. If f_n are differentiable, then the solution to (28)-(29) is easily computed.

We give two examples of the DelayFair policy. First, consider the minimization of the weighted sum of average queueing delays of all job classes subject to delay constraints, i.e.,

$$\text{minimize } \sum_{n=1}^N c_n \bar{W}_n, \quad \text{subject to } \bar{W}_n \leq d_n \quad n \in \{1, \dots, N\}, \quad (30)$$

where $c_n > 0$ for all n . The DelayFair policy chooses $r_{n,k}$ as the solution to

$$\text{minimize } (Vc_n - Y_{n,k} \lambda_n) r_{n,k}, \quad \text{subject to } r_{n,k} \in [0, d_n].$$

That is, we choose $r_{n,k} = 0$ if $Vc_n > Y_{n,k} \lambda_n$ and $r_{n,k} = d_n$ otherwise. When $c_n = \lambda_n$ for all n , i.e., we minimize the average queue occupancy $\sum_{n=1}^N \lambda_n \bar{W}_n$, the policy chooses $r_{n,k} = 0$ if $V > Y_{n,k}$ and $r_{n,k} = d_n$ otherwise—the knowledge of arrival rates is not needed here. Second, consider the problem of achieving delay proportional fairness with the penalty functions $f_n(\bar{W}_n) = \frac{1}{2} c_n (\bar{W}_n)^2$. The DelayFair policy solves, for each class n ,

$$\text{minimize } \frac{Vc_n}{2} (r_{n,k})^2 - Y_{n,k} \lambda_n r_{n,k}, \quad \text{subject to } r_{n,k} \in [0, d_n].$$

The solution is $r_{n,k}^* = \min \left\{ d_n, \frac{Y_{n,k} \lambda_n}{Vc_n} \right\}$.

Theorem 2 (Proof in Section 9.3) *If the delay requirements $\{d_1, \dots, d_N\}$ are feasible, then the DelayFair policy satisfies the constraints $\bar{W}_n \leq d_n$ for all classes n and yields convex delay cost satisfying*

$$\limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n \left(\frac{\mathbb{E} \left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} \right]}{\mathbb{E} \left[\sum_{k=0}^{K-1} |A_{n,k}| \right]} \right) \leq \frac{C \sum_{n=1}^N \lambda_n}{V} + \sum_{n=1}^N f_n(\bar{W}_n^*),$$

where $V > 0$ is a predefined control parameter and $C > 0$ a finite constant. The convex delay cost can be made arbitrarily close to the optimal value $\sum_{n=1}^N f_n(\bar{W}_n^*)$ by choosing V sufficiently large.

We remark that the DelayFair policy can be viewed as a learning algorithm. It updates controls by observing past queueing delays in each job class, and requires limited queue statistics. The effectiveness of the learning algorithm is controlled by the V parameter: Theorem 2 shows that a large V yields performance (average delay penalty) closer to optimal, at the expense of increasing the time to meet the time average constraints. Specifically, (77) suggests that the convergence speed of the DelayFair policy is related to $\sqrt{2(C+VD)}/K$, where C and D are positive constants, and K is the number of passed busy periods. Our control policies for the two service rate control problems presented later are also learning algorithms that have a similar tradeoff between performance and learning time.

5.3 Construction of the DelayFair policy

We derive a Lyapunov drift inequality that leads to the DelayFair policy. Define the Lyapunov function $L(\mathbf{Z}_k, \mathbf{Y}_k) \triangleq \frac{1}{2} \sum_{n=1}^N [(Z_{n,k})^2 + (Y_{n,k})^2]$ and the one-frame Lyapunov drift $\Delta(\mathbf{Z}_k, \mathbf{Y}_k) \triangleq \mathbb{E}[L(\mathbf{Z}_{k+1}, \mathbf{Y}_{k+1}) - L(\mathbf{Z}_k, \mathbf{Y}_k) \mid \mathbf{Z}_k, \mathbf{Y}_k]$. Squaring (27) yields

$$(Y_{n,k+1})^2 \leq \left[Y_{n,k} + \sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - r_{n,k}) \right]^2. \quad (31)$$

Summing (10) and (31) over $n \in \{1, \dots, N\}$, dividing the sum by two, and taking conditional expectation on the virtual queue vectors \mathbf{Z}_k and \mathbf{Y}_k , we get

$$\begin{aligned} \Delta(\mathbf{Z}_k, \mathbf{Y}_k) \leq & C - \sum_{n=1}^N Z_{n,k} d_n \mathbb{E}[|A_{n,k}| \mid \mathbf{Z}_k, \mathbf{Y}_k] - \sum_{n=1}^N Y_{n,k} \mathbb{E}[r_{n,k} | A_{n,k}| \mid \mathbf{Z}_k, \mathbf{Y}_k] \\ & + \sum_{n=1}^N (Z_{n,k} + Y_{n,k}) \mathbb{E}\left[\sum_{i \in A_{n,k}} W_{n,k}^{(i)} \mid \mathbf{Z}_k, \mathbf{Y}_k \right], \end{aligned} \quad (32)$$

where $C > 0$ is a finite constant, different from that used in the first problem, that upper-bounds the sum of all $(\mathbf{Z}_k, \mathbf{Y}_k)$ -independent terms in (32) (the existence of C can be proved similarly using Lemma 7 in Section 9.1).

Add to both sides of (32) the term $V \sum_{n=1}^N \mathbb{E}[f_n(r_{n,k}) T_k \mid \mathbf{Z}_k, \mathbf{Y}_k]$, where $V > 0$ is a control parameter. Evaluating the resulting inequality using the analysis in Section 4.1, we have the Lyapunov drift inequality:

$$\begin{aligned} \Delta(\mathbf{Z}_k, \mathbf{Y}_k) + V \sum_{n=1}^N \mathbb{E}[f_n(r_{n,k}) T_k \mid \mathbf{Z}_k, \mathbf{Y}_k] \leq & \left(C - \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \lambda_n d_n \right) \\ & + \mathbb{E}[T_k] \sum_{n=1}^N \mathbb{E}[V f_n(r_{n,k}) - Y_{n,k} \lambda_n r_{n,k} \mid \mathbf{Z}_k, \mathbf{Y}_k] \\ & + \mathbb{E}[T_k] \sum_{n=1}^N (Z_{n,k} + Y_{n,k}) \lambda_n \bar{W}_{n,k}, \end{aligned} \quad (33)$$

where $\bar{W}_{n,k}$ denotes the average queueing delay of class n if the control actions taken in the k th frame is independently repeated in all frames.

Over all admissible scheduling policies and all (possibly random) choices of $r_{n,k}$, we are interested in the one that minimizes the right-hand side of (33) in every frame. This is the DelayFair policy. In particular, the first and second step of the DelayFair policy minimizes the last sum (by the $c\mu$ rule in Lemma 2) and the second-to-last sum of (33), respectively. Note that we assume a constant service rate so that $\mathbb{E}[T_k]$ is fixed.

5.4 Intuition on minimizing the drift inequality

Following the ideas preceding the description of the DelayFair policy, we provide intuition on minimizing an upper bound on the left-hand side of (33). The Lyapunov

drift $\Delta(\mathbf{Z}_k, \mathbf{Y}_k)$ in (33) is the expected growth of the queue backlogs $Z_{n,k}$ and $Y_{n,k}$ over a frame. Similar to the first problem, minimizing $\Delta(\mathbf{Z}_k, \mathbf{Y}_k)$ in every frame stabilizes all $Z_{n,k}$ and $Y_{n,k}$ queues and satisfies the delay constraints in (24) and (25). Minimizing a separable convex function of the average service rates of the virtual queues $(Y_{1,k}, \dots, Y_{N,k})$ is closely related to minimizing $\sum_{n=1}^N \mathbb{E} [f_n(r_{n,k}) T_k | \mathbf{Z}_k, \mathbf{Y}_k]$ in the k th frame for all $k \in \mathbb{Z}^+$ (see the proof of Theorem 2 for details).

Minimizing both terms $\Delta(\mathbf{Z}_k, \mathbf{Y}_k)$ and $\sum_{n=1}^N \mathbb{E} [f_n(r_{n,k}) T_k | \mathbf{Z}_k, \mathbf{Y}_k]$ in each frame induces a tradeoff. Minimizing the former needs large $r_{n,k}$ values because they represent service opportunities of the virtual queues $Y_{n,k}$ (see (27)). Yet, minimizing the latter requires small $r_{n,k}$ values because f_n are nondecreasing functions. It is therefore natural to minimize a weighted sum of them, which is the left-hand side of (33). The performance tradeoff is controlled by the V parameter. As we will see shortly, a large V value puts more emphasis on minimizing $\sum_{n=1}^N \mathbb{E} [f_n(r_{n,k}) T_k | \mathbf{Z}_k, \mathbf{Y}_k]$, resulting in a convex delay penalty closer to optimal. The resulting tradeoff is that the $Z_{n,k}$ queues take longer to approach mean rate stability (see (77)), requiring a longer time to meet the time average requirements $\bar{W}_n \leq d_n$.

6 Third problem: Delay-constrained service rate control

We incorporate dynamic service rate allocations into the queue control problem. As mentioned in Section 2, we focus on policies that allocate a fixed service rate $\mu(P_k)$ in the k th busy period with an instantaneous cost $P_k \in [P_{\min}, P_{\max}]$. Zero service rates are allocated when the system is idle. Here, the frame size T_k , busy period B_k , class n arrivals $A_{n,k}$ in a frame, and queueing delays $W_{n,k}^{(i)}$ are all functions of P_k . Similar to the definition of average delay in (1), we define the average service cost as

$$\bar{P} \triangleq \limsup_{K \rightarrow \infty} \frac{\mathbb{E} [\sum_{k=0}^{K-1} P_k B_k(P_k)]}{\mathbb{E} [\sum_{k=0}^{K-1} T_k(P_k)]}, \quad (34)$$

where $B_k(P_k)$ and $T_k(P_k)$ emphasize the dependence of B_k and T_k on P_k . It is easy to show that $B_k(P_k)$ and $T_k(P_k)$ are decreasing in P_k (i.e., in the service rate $\mu(P_k)$).

6.1 The cost-delay performance region

Before designing optimal queue control policies, we define the performance region of average delay and average service cost. We consider the set of control policies, denoted by $\hat{\Pi}$, with the properties: (i) for each feasible service rate $\mu(P)$, the limiting proportion of busy periods in which $\mu(P)$ is allocated exists; (ii) scheduling decisions in one busy period may depend on the service rate in that period, but are independent of scheduling decisions in other busy periods; (iii) scheduling decisions are stationary (but possibly random) over busy periods that have the same service rates.³ We define the performance region $\Lambda = \{(\bar{P}, \bar{W}_1, \dots, \bar{W}_N)\}$ as the set of average cost-delay vectors achieved by control policies in $\hat{\Pi}$.

³ These properties are used to guarantee that the limits of long-term average delay and average service cost exist, so that we have a well-defined performance region.

6.2 The queue control problem and the control policy

We consider the delay-constrained service rate control problem:

$$\text{minimize } \bar{P} \quad (35)$$

$$\text{subject to } \bar{W}_n \leq d_n, \quad n \in \{1, \dots, N\} \quad (36)$$

$$(\bar{P}, \bar{W}_1, \dots, \bar{W}_N) \in \Lambda. \quad (37)$$

The following policy solves (35)-(37). We set up the same virtual queues $(Z_{1,k}, \dots, Z_{N,k})$ as in (9) to satisfy the delay requirements (36). Initially, let $Z_{n,0} = 0$ for all n .

Dynamic Rate Control Policy (DynRate):

- In the k th busy period, use a strict priority policy π_k^* that prioritizes job classes in the decreasing order of $Z_{n,k}/\mathbb{E}[S_n]$, where $\mathbb{E}[S_n]$ is the mean job size of class n ; ties are broken arbitrarily. Define n_j^* as the job class that has the j th highest priority under π_k^* .
- In the k th busy period, allocate the service rate $\mu(P_k)$ where P_k solves:

$$\text{minimize } \left(V \sum_{n=1}^N \lambda_n \mathbb{E}[S_n] \right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^N Z_{n,k} \lambda_n \bar{W}_n(\pi_k^*, P_k) \quad (38)$$

$$\text{subject to } P_k \in [P_{\min}, P_{\max}], \quad (39)$$

where $\bar{W}_n(\pi_k^*, P_k)$ is the average delay of class n if the service rate $\mu(P_k)$ and the policy π_k^* are used in all busy periods—if class m has the j th highest priority, i.e., $m = n_j^*$, then from (8) we have

$$\bar{W}_m(\pi_k^*, P_k) = \frac{(1/2) \sum_{i=1}^N \lambda_i \mathbb{E}[S_i^2] / (\mu(P_k))^2}{(1 - \sum_{i=0}^{j-1} \rho_{n_i^*})(1 - \sum_{i=0}^j \rho_{n_i^*})}, \quad (40)$$

where $\rho_i = \lambda_i \mathbb{E}[S_i] / \mu(P_k)$, $\rho_0 \triangleq 0$, and $\rho_0 = 0$.

- Update $Z_{n,k}$ according to (9) at the end of busy periods.

The DynRate policy requires the knowledge of arrival rates and the first two moments of job sizes. We can remove its dependence on the second moments of job sizes so that the policy depends only on first-order statistics; see Section 9.4 for details.

Theorem 3 (Proof in Section 9.5) *Let P^* be the optimal average service cost in the problem (35)-(37). The DynRate policy satisfies all delay constraints $\bar{W}_n \leq d_n$ and attains average service cost \bar{P} satisfying*

$$\bar{P} \leq \frac{C \sum_{n=1}^N \lambda_n}{V} + P^*,$$

where $C > 0$ is a finite constant and $V > 0$ a predefined control parameter. The gap between \bar{P} and the optimal P^* can be made arbitrarily small by a sufficiently large V .

6.3 Construction of the DynRate policy

We provide a useful Lyapunov drift inequality and provide intuitions later. Define the Lyapunov function $L(\mathbf{Z}_k) = \frac{1}{2} \sum_{n=1}^N (Z_{n,k})^2$ and the Lyapunov drift $\Delta(\mathbf{Z}_k) = \mathbb{E}[L(\mathbf{Z}_{k+1}) - L(\mathbf{Z}_k) \mid \mathbf{Z}_k]$. Following the analysis in Section 4.1, we have

$$\Delta(\mathbf{Z}_k) \leq C + \sum_{n=1}^N Z_{n,k} \mathbb{E} \left[\sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \mid \mathbf{Z}_k \right]. \quad (41)$$

Adding $V \mathbb{E}[P_k B_k(P_k) \mid \mathbf{Z}_k]$ to both sides of (41), where $V > 0$ is a control parameter, we get

$$\Delta(\mathbf{Z}_k) + V \mathbb{E}[P_k B_k(P_k) \mid \mathbf{Z}_k] \leq C + \Phi(\mathbf{Z}_k), \quad (42)$$

where

$$\Phi(\mathbf{Z}_k) \triangleq \mathbb{E} \left[V P_k B_k(P_k) + \sum_{n=1}^N Z_{n,k} \sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \mid \mathbf{Z}_k \right].$$

We want the control policy that, in each frame k , makes admissible scheduling decisions and assigns a fixed service rate to minimize the ratio

$$\frac{\Phi(\mathbf{Z}_k)}{\mathbb{E}[T_k(P_k) \mid \mathbf{Z}_k]}. \quad (43)$$

The decisions are possibly random. The frame size $T_k(P_k)$ depends on \mathbf{Z}_k because the choice of P_k may be \mathbf{Z}_k -dependent. (For a given P_k , $T_k(P_k)$ is independent of \mathbf{Z}_k .)

The intuition on minimizing (43) is as follows. Similar to previous problems, minimizing the Lyapunov drift $\Delta(\mathbf{Z}_k)$ in every frame helps to achieve the delay constraints $\bar{W}_n \leq d_n$ for all classes n . We may increase the service rate with higher cost to improve queueing delay, which reduces $\Delta(\mathbf{Z}_k)$ because the delays of the jobs are ‘‘arrivals’’ to the virtual queues $Z_{n,k}$. Thus, a tradeoff is induced between service cost and the stability of the $Z_{n,k}$ queues, captured by the left-hand side of (42). If we were to follow what we do in previous problems, we would minimize the right-hand side of (42), i.e., $\Phi(\mathbf{Z}_k)$, in every frame. This is insufficient here because the frame size depends on the allocated service rate. The proper step is to minimize the ratio of $\Phi(\mathbf{Z}_k)$ over the average frame size, namely (43).

We simplify (43) to show the DynRate policy minimizes (43). Lemma 5 below shows that the minimizer of (43) is a deterministic service rate allocation and strict priority policy. We may consider every $x \in \mathcal{X}$ in Lemma 5 as one such deterministic policy, and (43) evaluated under policy x is equal to $\mathbb{E}[G(x)] / \mathbb{E}[H(x)]$. The random variable X denotes a randomized policy, and (43) evaluated under policy X is $\mathbb{E}[G] / \mathbb{E}[H]$.

Lemma 5 *Let X be a continuous random variable with state space \mathcal{X} . Let G and H be two random variables that depend on the realization of X such that, for each $x \in \mathcal{X}$, $G(x)$ and $H(x)$ are well-defined positive random variables. Define*

$$x^* \triangleq \operatorname{argmin}_{x \in \mathcal{X}} \frac{\mathbb{E}[G(x)]}{\mathbb{E}[H(x)]}, \quad U^* \triangleq \frac{\mathbb{E}[G(x^*)]}{\mathbb{E}[H(x^*)]}.$$

Then $\frac{\mathbb{E}[G]}{\mathbb{E}[H]} \geq U^$ regardless of the distribution of X .*

Proof (Lemma 5) For each $x \in \mathcal{X}$, we have $\frac{\mathbb{E}[G(x)]}{\mathbb{E}[H(x)]} \geq U^*$. Then

$$\frac{\mathbb{E}[G]}{\mathbb{E}[H]} = \frac{\mathbb{E}_X[\mathbb{E}[G(x)]]}{\mathbb{E}_X[\mathbb{E}[H(x)]]} \geq \frac{\mathbb{E}_X[U^* \mathbb{E}[H(x)]]}{\mathbb{E}_X[\mathbb{E}[H(x)]]} = U^*,$$

which is independent of the distribution of X . \square

Next, we evaluate the ratio of expectation (43) under a fixed service rate $\mu(P_k)$ and a strict priority policy π_k in the k th frame. The value of \mathbf{Z}_k affects only how we choose the service rate and the control policy in the k th frame. After $\mu(P_k)$ and π_k are chosen, the vector \mathbf{Z}_k does not affect the value of (43). Therefore, (43) becomes

$$\frac{\Phi(\mathbf{Z}_k)}{\mathbb{E}[T_k(P_k) | \mathbf{Z}_k]} = \frac{\mathbb{E}[VP_k B_k(P_k)]}{\mathbb{E}[T_k(P_k)]} + \sum_{n=1}^N Z_{n,k} \left\{ \frac{\mathbb{E}[\sum_{i \in A_{n,k}} W_{n,k}^{(i)}]}{\mathbb{E}[T_k(P_k)]} - \frac{\mathbb{E}[\sum_{i \in A_{n,k}} d_n]}{\mathbb{E}[T_k(P_k)]} \right\}. \quad (44)$$

Under the fixed service rate $\mu(P_k)$, the first term on the right side of (44) is

$$VP_k \frac{\mathbb{E}[B_k(P_k)]}{\mathbb{E}[T_k(P_k)]}. \quad (45)$$

To compute (45), let us create a new multi-class $M/G/1$ queue with the same arrival processes with rates $(\lambda_1, \dots, \lambda_N)$ and job size distributions (S_1, \dots, S_N) as those given in Section 2; assume the queue is initially empty. We suppose that this new $M/G/1$ queue uses the service rate $\mu(P_k)$ and the strict priority policy π_k in all frames. Let \hat{B}_1 and \hat{T}_1 be the first busy and renewal period of the new $M/G/1$ queue, respectively. From renewal theory, we have

$$\frac{\mathbb{E}[\hat{B}_1]}{\mathbb{E}[\hat{T}_1]} = \sum_{n=1}^N \rho_n = \sum_{n=1}^N \lambda_n \frac{\mathbb{E}[S_n]}{\mu(P_k)}.$$

We observe that the busy period $B_k(P_k)$ in the k th frame of the $M/G/1$ queue with dynamic service rates is statistically identical to the first busy period \hat{B}_1 of the new $M/G/1$ queue. Likewise, the renewal period $T_k(P_k)$ is statistically identical to \hat{T}_1 . Therefore, (45) satisfies

$$VP_k \frac{\mathbb{E}[B_k(P_k)]}{\mathbb{E}[T_k(P_k)]} = VP_k \frac{\mathbb{E}[\hat{B}_1]}{\mathbb{E}[\hat{T}_1]} = VP_k \sum_{n=1}^N \lambda_n \frac{\mathbb{E}[S_n]}{\mu(P_k)}.$$

Next, we compute the second term on the right side of (44):

$$\frac{\mathbb{E}[\sum_{i \in A_{1,k}} W_{1,k}^{(i)}]}{\mathbb{E}[T_k(P_k)]}$$

for class 1 jobs under the strict priority policy π_k and service rate $\mu(P_k)$. From (13), we have

$$\frac{\mathbb{E}[\sum_{i \in A_{1,k}} W_{1,k}^{(i)}]}{\mathbb{E}[T_k(P_k)]} = \frac{\mathbb{E}[\int_{t_k}^{t_{k+1}} Q_1(t) dt]}{\mathbb{E}[T_k(P_k)]}. \quad (46)$$

To evaluate (46), let us look at the first renewal period of the new $M/G/1$ queue created above. From renewal reward theory, the long-term average number of waiting class 1 jobs in this new queue, denoted by \overline{Q}_1 , is

$$\overline{Q}_1 = \frac{\mathbb{E}\left[\int_0^{\hat{T}_1} \hat{Q}_1(t) dt\right]}{\mathbb{E}[\hat{T}_1]} = \lambda_1 \overline{W}_1^*, \quad (47)$$

where $[0, \hat{T}_1]$ is the first renewal period, $\hat{Q}_1(t)$ is the number of class 1 jobs waiting in the queue at time t , λ_1 is the class 1 arrival rate, and \overline{W}_1^* is the average queueing delay of class 1 jobs. The equality in (47) uses Little's Theorem. The value of \overline{W}_1^* is given in (8), depending on the priority of class 1 jobs. We observe that the k th frame of the $M/G/1$ queue in this paper, under the fixed service rate $\mu(P_k)$ and the strict priority policy π_k , behaves statistically the same as the first renewal period of the new $M/G/1$ queue. Therefore, from (46)-(47) we have

$$\frac{\mathbb{E}\left[\sum_{i \in A_{1,k}} W_{1,k}^{(i)}\right]}{\mathbb{E}[T_k(P_k)]} = \frac{\mathbb{E}\left[\int_{t_k}^{t_{k+1}} Q_1(t) dt\right]}{\mathbb{E}[T_k(P_k)]} = \frac{\mathbb{E}\left[\int_0^{\hat{T}_1} \hat{Q}_1(t) dt\right]}{\mathbb{E}[\hat{T}_1]} = \lambda_1 \overline{W}_1^*.$$

We can evaluate the rest of the terms in (44) using the same argument, and obtain

$$\frac{\Phi(\mathbf{Z}_k)}{\mathbb{E}[T_k(P_k) | \mathbf{Z}_k]} = V P_k \frac{\sum_{n=1}^N \lambda_n \mathbb{E}[S_n]}{\mu(P_k)} + \sum_{n=1}^N Z_{n,k} \lambda_n (\overline{W}_n(\pi_k, P_k) - d_n), \quad (48)$$

where $\overline{W}_n(\pi_k, P_k)$ is the average queueing delay of class n jobs in the new $M/G/1$ queue created above under the constant service rate $\mu(P_k)$ and the strict priority policy π_k , and $\overline{W}_n(\pi_k, P_k)$ is given in (40). It follows that our desired policy minimizes

$$\left(V \sum_{n=1}^N \lambda_n \mathbb{E}[S_n] \right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^N Z_{n,k} \lambda_n \overline{W}_n(\pi_k, P_k) \quad (49)$$

in each frame k over $P_k \in [P_{\min}, P_{\max}]$ and over the set of strict priority policies.

To further simplify, under a given fixed service rate $\mu(P_k)$, the second term of (49) can be re-written as

$$\mu(P_k) \sum_{n=1}^N \frac{Z_{n,k}}{\mathbb{E}[S_n]} \frac{\lambda_n \mathbb{E}[S_n]}{\mu(P_k)} \overline{W}_n(\pi_k, P_k),$$

which is minimized by the $c\mu$ rule that assigns strict priorities in the decreasing order of $Z_{n,k}/\mathbb{E}[S_n]$. This strict priority policy, denoted by π_k^* , is optimal regardless of the value of P_k , and thus is overall optimal. Interestingly, priority assignment is decoupled from optimal service rate allocation. Under policy π_k^* , choosing P_k to solve (38)-(39) reveals the optimal service rate allocation in the k th frame. These discussions lead to the DynRate policy.

7 Fourth problem: Cost-constrained convex delay optimization

In the fourth problem, we design a policy to minimize a separable convex function of the average queueing delay vector subject to an average service cost constraint:

$$\text{minimize } \sum_{n=1}^N f_n(\bar{W}_n) \quad (50)$$

$$\text{subject to } \bar{P} \leq P_{\text{const}} \quad (51)$$

$$(\bar{P}, \bar{W}_1, \dots, \bar{W}_N) \in \Lambda, \quad (52)$$

where $P_{\text{const}} > 0$ is a given feasible bound. The functions f_n are nondecreasing, non-negative, continuous, and convex. We use the same virtual queues $(Y_{1,k}, \dots, Y_{N,k})$ as in (27), except that the auxiliary variable $r_{n,k}$ takes values in a new interval $[0, R_{\text{max},n}]$. We need $R_{\text{max},n}$ to be greater than the optimal delay \bar{W}_n^* that solves (50)-(52); one feasible choice of $R_{\text{max},n}$ is the maximum per-class average delay over all job classes under the minimum service rate $\mu(P_{\text{min}})$. To satisfy (51), we define a *virtual cost queue* $\{X_k\}_{k=0}^{\infty}$, where $X_0 = 0$, that is updated at the end of busy periods $\{t_k\}_{k=0}^{\infty}$ by

$$X_{k+1} = \max[X_k + P_k B_k(P_k) - P_{\text{const}} T_k(P_k), 0]. \quad (53)$$

That is, X_{k+1} is computed at the end of the k th frame after observing the busy period $B_k(P_k)$ and the frame size $T_k(P_k)$ of the k th frame.

Lemma 6 (Proof in Section 9.6) *If the virtual queue X_k is mean rate stable, then $\bar{P} \leq P_{\text{const}}$.*

7.1 The control policy

The following policy, similar to the DelayFair and DynRate policy, solves (50)-(52).

Cost-Constrained Delay Fairness Policy (CostDelayFair):

- In the k th busy period, observe X_k and $Y_{n,k}$ and use the strict priority policy π_k^* that prioritizes job classes in the decreasing order of $Y_{n,k}/\mathbb{E}[S_n]$; ties are broken arbitrarily. Allocate the service rate $\mu(P_k)$ that solves:

$$\begin{aligned} & \text{minimize} && \frac{P_k}{\mu(P_k)} \left[X_k \sum_{n=1}^N \lambda_n \mathbb{E}[S_n] \right] + \sum_{n=1}^N Y_{n,k} \lambda_n \bar{W}_n(\pi_k^*, P_k) \\ & \text{subject to} && P_k \in [P_{\min}, P_{\max}], \end{aligned}$$

where $\bar{W}_n(\pi_k^*, P_k)$, given in (40), represents the average delay of class n if the policy π_k^* and the service rate $\mu(P_k)$ are used in all busy periods.

- At the end of busy periods, update $Y_{n,k}$ for all classes n and X_k by (27) and (53), respectively. In (27), the auxiliary variable $r_{n,k}$ is the solution to the one-variable convex program

$$\begin{aligned} & \text{minimize} && V f_n(r_{n,k}) - Y_{n,k} \lambda_n r_{n,k} \\ & \text{subject to} && 0 \leq r_{n,k} \leq R_{\max,n} \end{aligned}$$

which is easily solved if f_n is differentiable.

Theorem 4 (Proof in Section 9.7) *The CostDelayFair policy satisfies the average service cost constraint $\bar{P} \leq P_{\text{const}}$ and yields average delay penalty satisfying*

$$\limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n \left(\frac{\mathbb{E} \left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} \right]}{\mathbb{E} \left[\sum_{k=0}^{K-1} |A_{n,k}| \right]} \right) \leq \frac{C \sum_{n=1}^N \lambda_n}{V} + \sum_{n=1}^N f_n(\bar{W}_n^*), \quad (54)$$

where $V > 0$ is a control parameter and $(\bar{W}_n^*)_{n=1}^N$ the optimal average delay vector.

7.2 Construction of the CostDelayFair policy

The design of the CostDelayFair policy follows closely with those in previous problems, and the details are omitted for brevity. Define the vector $\chi_k = [X_k; Y_{1,k}, \dots, Y_{N,k}]$, the Lyapunov function $L(\chi_k) \triangleq \frac{1}{2}(X_k^2 + \sum_{n=1}^N Y_{n,k}^2)$, and the Lyapunov drift $\Delta(\chi_k) \triangleq \mathbb{E}[L(\chi_{k+1}) - L(\chi_k) \mid \chi_k]$. There exists a finite constant $C > 0$ such that

$$\Delta(\chi_k) \leq C + X_k \mathbb{E}[P_k B_k(P_k) - P_{\text{const}} T_k(P_k) \mid \chi_k] + \sum_{n=1}^N Y_{n,k} \mathbb{E} \left[\sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - r_{n,k}) \mid \chi_k \right]. \quad (55)$$

Adding $V \sum_{n=1}^N \mathbb{E}[f_n(r_{n,k}) T_k(P_k) \mid \chi_k]$ to both sides of (55), and evaluating the resulting inequality under a control policy that makes admissible scheduling decisions and allocates a fixed service rate in the k th frame, we obtain

$$\Delta(\chi_k) + V \sum_{n=1}^N \mathbb{E}[f_n(r_{n,k}) T_k(P_k) \mid \chi_k] \leq C + \Psi(\chi_k), \quad (56)$$

where

$$\begin{aligned} \Psi(\chi_k) \triangleq & \mathbb{E}[T_k(P_k) \mid \chi_k] \sum_{n=1}^N Y_{n,k} \lambda_n \bar{W}_{n,k} + X_k \left(\mathbb{E}[P_k B_k(P_k) \mid \chi_k] - P_{\text{const}} \mathbb{E}[T_k(P_k) \mid \chi_k] \right) \\ & + \mathbb{E}[T_k(P_k) \mid \chi_k] \sum_{n=1}^N \mathbb{E}[V f_n(r_{n,k}) - Y_{n,k} \lambda_n r_{n,k} \mid \chi_k] \end{aligned}$$

and $\bar{W}_{n,k}$ denotes the average delay of class n if the control in the k th frame is independently repeated in all frames. Consider the policy that minimizes the ratio

$$\frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k) \mid \chi_k]} \quad (57)$$

in each frame k . Lemma 5 shows that the minimizer is a strict priority policy π_k with a fixed service rate, under which (57) is equal to

$$\sum_{n=1}^N Y_{n,k} \lambda_n \bar{W}_n(\pi_k, P_k) + X_k (P_k \rho_{\text{sum}}(P_k) - P_{\text{const}}) + \sum_{n=1}^N (V f_n(r_{n,k}) - Y_{n,k} \lambda_n r_{n,k}),$$

where $\bar{W}_n(\pi_k, P_k)$ is given in (40) and $\rho_{\text{sum}}(P_k) \triangleq \sum_{n=1}^N \lambda_n \mathbb{E}[S_n] / \mu(P_k)$. Under similar simplifications as those for the DynRate policy in Section 6, the CostDelayFair policy is the desired policy.

8 Simulations

We simulate all four control policies in a two-class nonpreemptive $M/G/1$ queue. Let $\mathcal{W}(P)$ be the delay performance region when the queue has a fixed service rate $\mu(P)$. Define $\rho_n \triangleq \lambda_n \mathbb{E}[X_n]$ and $R \triangleq \frac{1}{2} \sum_{n=1}^2 \lambda_n \mathbb{E}[X_n^2]$, where $X_n = S_n / \mu(P)$. We have, from (5),

$$\mathcal{W}(P) = \left\{ (\bar{W}_1, \bar{W}_2) \left| \begin{array}{l} \bar{W}_1 \geq \frac{R}{1-\rho_1}, \bar{W}_2 \geq \frac{R}{1-\rho_2}, \\ \rho_1 \bar{W}_1 + \rho_2 \bar{W}_2 = \frac{(\rho_1 + \rho_2)R}{1-\rho_1-\rho_2} \end{array} \right. \right\}. \quad (58)$$

In (58), the two inequalities say that the average queuing delay in one class is minimized when it has strict priority over the other class. The equality is the $M/G/1$ conservation law [21].

Each simulation below is a sample average over 10 runs, each of which lasts for 10^6 frames.

8.1 The DelayFeas and DelayFair policy

To simulate the DelayFeas and DelayFair policy, we consider a two-class nonpreemptive $M/M/1$ queue with arrival rates $(\lambda_1, \lambda_2) = (1, 2)$ and mean service times $(\mathbb{E}[X_1], \mathbb{E}[X_2]) = (0.4, 0.2)$; we consider service time directly instead of job sizes because there is no service rate control. The delay performance region, from (58), is

$$\mathcal{W} = \{ (\bar{W}_1, \bar{W}_2) \mid \bar{W}_1 + \bar{W}_2 = 2.4, \bar{W}_1 \geq 0.4, \bar{W}_2 \geq 0.4 \}, \quad (59)$$

which is presented in Fig. 1.

For the DelayFeas policy, we consider five sets of delay constraints $(d_1, d_2) = (0.45, 2.05), (0.85, 1.65), (1.25, 1.25), (1.65, 0.85),$ and $(2.05, 0.45)$; they are all $(0.05, 0.05)$ entrywise larger than a feasible point in the delay region \mathcal{W} . The simulation results in Fig. 1 show that the DelayFeas policy adaptively yields feasible average delays in response to different constraints. Over the 10 simulation runs in each of the five cases, the sample standard deviation of the average delay in each job class is at most 0.017. Therefore, different simulation runs produce consistent results. Fig. 2 shows the convergence of the running delay performance of the DelayFeas policy to a feasible delay vector under different delay requirements (d_1, d_2) .⁴

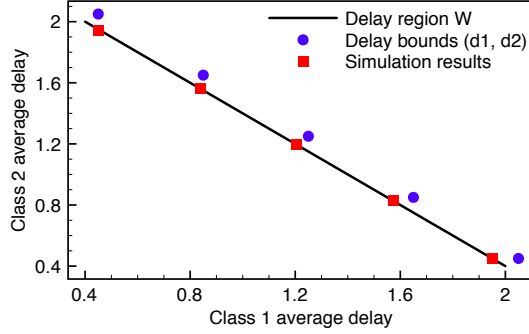


Fig. 1 The DelayFeas policy under different delay constraints (d_1, d_2) .

For the DelayFair policy, we consider the delay proportional fairness problem:

$$\text{minimize } 0.5(\bar{W}_1)^2 + 2(\bar{W}_2)^2 \quad (60)$$

$$\text{subject to } (\bar{W}_1, \bar{W}_2) \in \mathcal{W} \quad (61)$$

$$\bar{W}_1 \leq 1.95, \bar{W}_2 \leq 1 \quad (62)$$

where \mathcal{W} is given in (59). The optimal solution to (60)-(62) is $(\bar{W}_1^*, \bar{W}_2^*) = (1.92, 0.48)$, and the optimal objective is 2.304. We simulate the DelayFair policy for different values of the control parameter V , and the results are presented in Table 1. The values

⁴ In the fifth case of Fig. 2 with delay requirements $(d_1, d_2) = (0.45, 2.05)$, the class 1 delay of the DelayFeas policy is slightly larger than 0.45 because the initial learning phase of the policy is taken into account; the class 1 delay is strictly less than 0.45 if the initial phase is excluded.

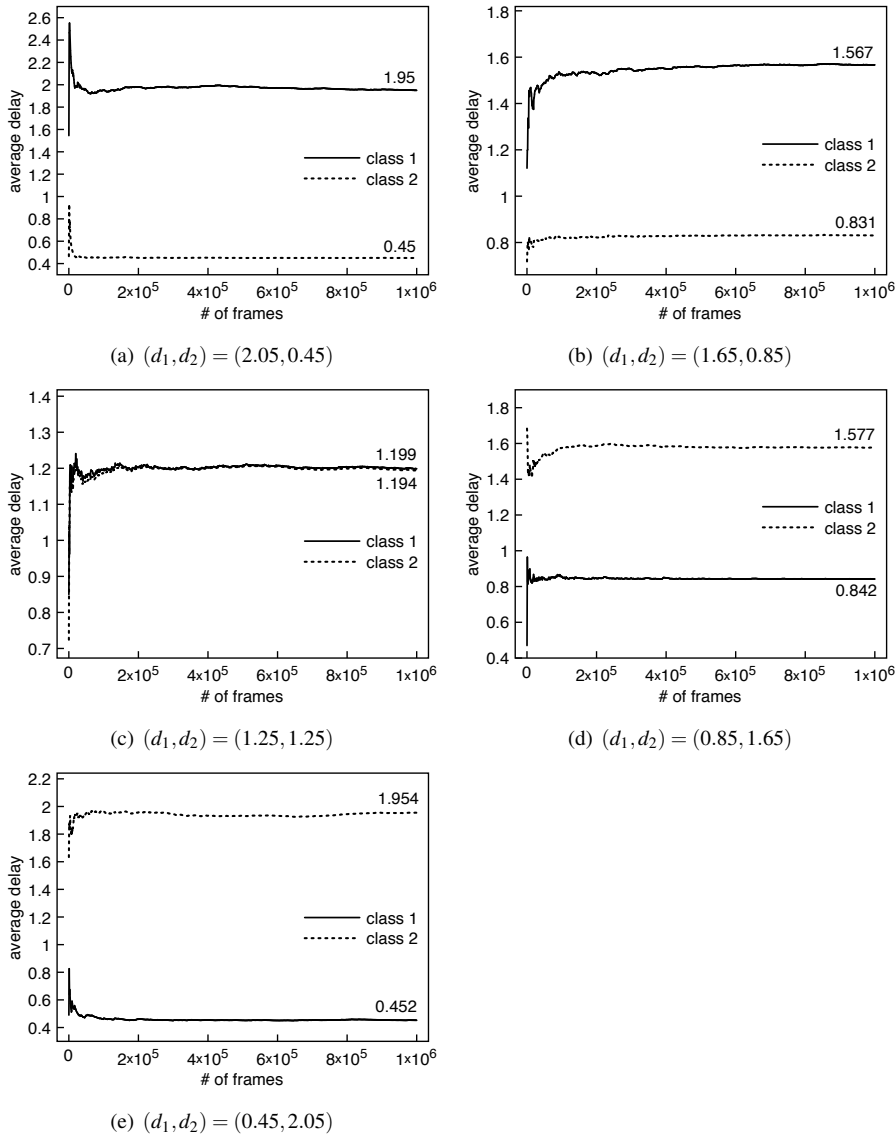
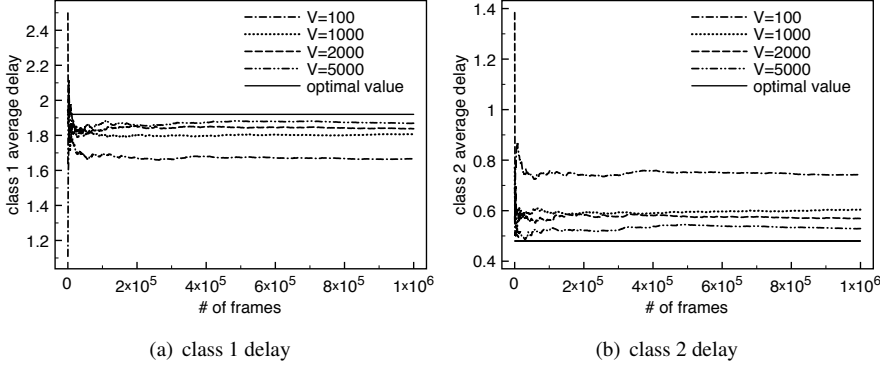


Fig. 2 Running average delay performance of the DelayFeas policy under different delay bounds (d_1, d_2) .

in parentheses in Table 1 are sample standard deviations over the 10 simulation runs. As V increases, the DelayFair policy yields average delay penalty approaching the optimal value. Fig. 3 shows the convergence of the running delay performance of the DelayFair policy.

Table 1 Simulations for the DelayFair policy under different values of V

V	\bar{W}_1	\bar{W}_2	$0.5(\bar{W}_1)^2 + 2(\bar{W}_2)^2$
100	1.661 (.006)	0.742 (.005)	2.481 (.024)
1000	1.798 (.006)	0.598 (.004)	2.332 (.020)
2000	1.834 (.006)	0.564 (.005)	2.318 (.022)
5000	1.868 (.007)	0.528 (.005)	2.301 (.022)
optimal	1.92	0.48	2.304

**Fig. 3** Running average delay performance of the DelayFair policy under different values of V .

8.2 The DynRate and CostDelayFair Policy

In the two service rate control problems, we consider a two-class $M/G/1$ queue with arrival rates $(\lambda_1, \lambda_2) = (1, 2)$. The size of a class 1 job is 0.5 with probability 0.8 and 3 otherwise. The size of a class 2 job is always one. The feasible choice of service cost in a busy period is in the discrete set $P \in \{16, 25\}$. We consider the service rate $\mu(P) = \sqrt{P}$. The full delay performance region, denoted by \mathcal{W} , is the convex hull of the two individual regions (see Fig. 4):

$$\begin{aligned} \mathcal{W}(16) &= \{(\bar{W}_1, \bar{W}_2) \mid \bar{W}_1 + 2\bar{W}_2 = 3/2, \bar{W}_1 \geq 1/6, \bar{W}_2 \geq 1/4\}, \\ \mathcal{W}(25) &= \{(\bar{W}_1, \bar{W}_2) \mid \bar{W}_1 + 2\bar{W}_2 = 3/5, \bar{W}_1 \geq 1/10, \bar{W}_2 \geq 2/15\}, \end{aligned}$$

where $\mathcal{W}(16)$ and $\mathcal{W}(25)$ are the delay performance regions, given by (58), under a constant service cost of $P = 16$ and $P = 25$, respectively.

For the DynRate policy, we solve

$$\text{minimize } \bar{P} \tag{63}$$

$$\text{subject to } (\bar{W}_1, \bar{W}_2) \in \mathcal{W} \tag{64}$$

$$\bar{W}_1 \leq 0.4, \bar{W}_2 \leq 0.325 \tag{65}$$

where \mathcal{W} is the full delay region in Fig. 4. The minimum average service cost is achieved by satisfying the constraints (65) with equality. By finding the stationary

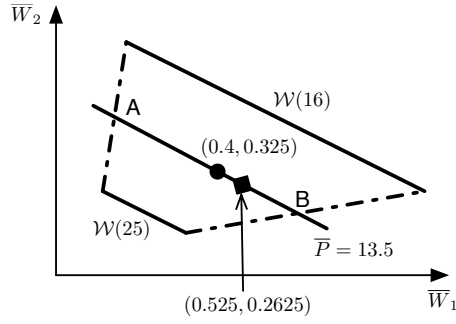


Fig. 4 The full delay performance region \mathcal{W} , as a convex hull of the two regions $\mathcal{W}(16)$ and $\mathcal{W}(25)$, in the simulations for the DynRate and the CostDelayFair policy.

Table 2 Simulations for the DynRate policy under different values of V

V	\bar{W}_1	\bar{W}_2	\bar{P}
1	0.356 (.00078)	0.301 (.00032)	13.802 (.018)
10	0.398 (.00022)	0.325 (.00005)	13.510 (.026)
100	0.400 (.00013)	0.325 (.00010)	13.504 (.022)
optimal	0.4	0.325	13.5

randomized policy that yields $(\bar{W}_1, \bar{W}_2) = (0.4, 0.325)$, we know the optimal average service cost is 13.5. Table 2 presents simulation results for the DynRate policy for different values of parameter V ; sample standard deviations over simulation runs are in parentheses. We observe that average service cost as well as average delay in each job class approaches optimal with the increase of V . Fig. 5 shows the convergence of the running delay and service cost performance of the DynRate policy.

For the CostDelayFair policy, we solve

$$\text{minimize } 0.5(\bar{W}_1)^2 + 2(\bar{W}_2)^2 \quad (66)$$

$$\text{subject to } \bar{P} \leq 13.5. \quad (67)$$

The optimal policy must satisfy (67) with equality. In Fig. 4, the set of feasible average delay vectors inducing average cost $\bar{P} = 13.5$ forms a line segment AB that is parallel to both delay regions $\mathcal{W}(16)$ and $\mathcal{W}(25)$ and passes $(0.4, 0.325)$. This can be shown geometrically in Fig. 6 by observing that any randomized policy that achieves some point on AB must use the same coefficients to form a convex combination of one point on $\mathcal{W}(25)$ and one on $\mathcal{W}(16)$, and thus this policy incurs the same average cost $\bar{P} = 13.5$. Consequently, (66)-(67) is equivalent to

$$\text{minimize } 0.5(\bar{W}_1)^2 + 2(\bar{W}_2)^2 \quad (68)$$

$$\text{subject to } \bar{W}_1 + \bar{W}_2 = 1.05 \quad (69)$$

$$\bar{W}_1 \geq 2/15, \bar{W}_2 \geq 23/120, \quad (70)$$

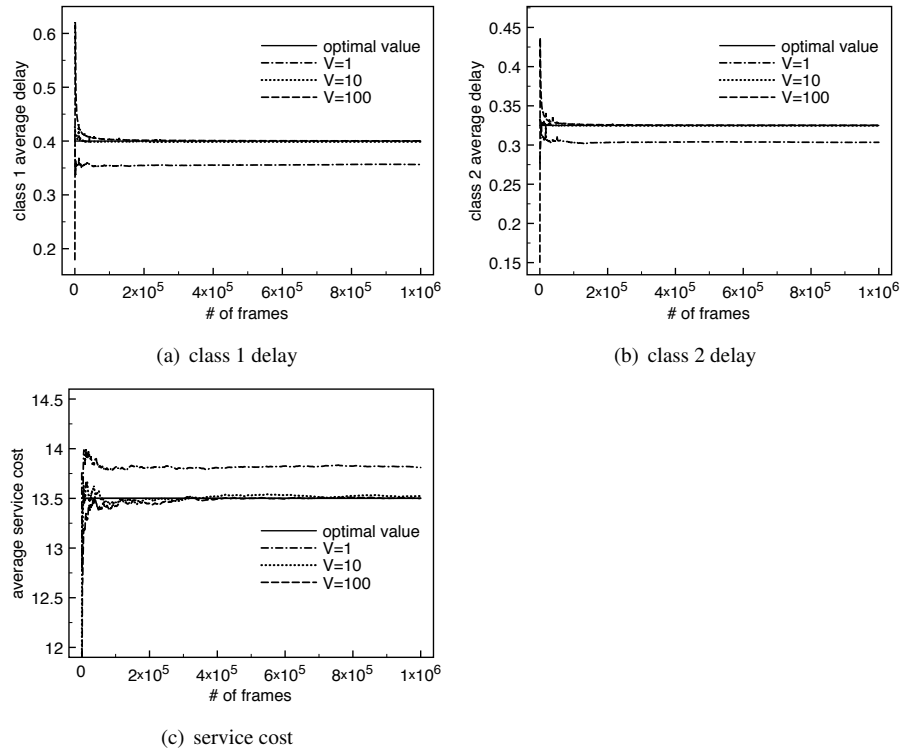


Fig. 5 Running average delay and average service cost performance of the DynRate policy under different values of V .

where the constraints (69)-(70) represent the line segment AB in Fig. 4. The optimal average delay vector is $(\bar{W}_1^*, \bar{W}_2^*) = (0.525, 0.2625)$. Table 3 presents the simulation results of the CostDelayFair policy. Again, the performance approaches optimal as V increases. Fig. 7 shows the convergence of the running delay and service cost performance of the CostDelayFair policy.

Table 3 Simulations for the CostDelayFair policy under different values of V

V	\bar{W}_1	\bar{W}_2	$0.5(\bar{W}_1)^2 + 2(\bar{W}_2)^2$	\bar{P}
100	0.566 (.0031)	0.304 (.0013)	0.345 (.0032)	13.082 (.0030)
200	0.542 (.0017)	0.286 (.0009)	0.310 (.0017)	13.274 (.0029)
500	0.525 (.0023)	0.271 (.0011)	0.284 (.0022)	13.454 (.0014)
1000	0.520 (.0022)	0.265 (.0010)	0.276 (.0022)	13.496 (.0006)
optimal	0.525	0.2625	0.2756	13.5

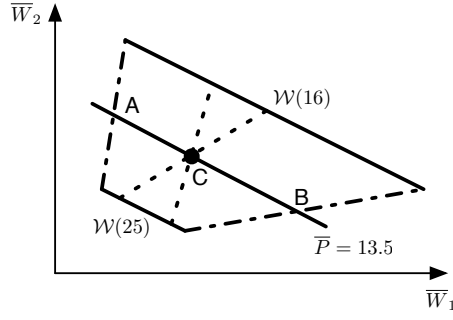


Fig. 6 The two dotted lines passing a point C on the line segment AB represent two randomized policies that achieve C . Geometrically, they have the same proportional mixture of one point on $\mathcal{W}(25)$ and one on $\mathcal{W}(16)$. Therefore, they incur the same average service cost.

9 Proofs and additional results

9.1 Lemma 7

Lemma 7 *In a nonpreemptive N -class $M/G/1$ queue with a constant service rate, if the first four moments of service times X_n are finite for all classes $n \in \{1, \dots, N\}$, and the system is stable with $\sum_{n=1}^N \lambda_n \mathbb{E}[X_n] < 1$, then, in every frame $k \in \mathbb{Z}^+$, the term $\mathbb{E} \left[\left(\sum_{i \in A_{n,k}} (W_{n,k}^{(i)} - d_n) \right)^2 \right]$ is finite for all classes n under any work-conserving policy.*

Proof (Lemma 7) For brevity, we only give a sketch of proof. Using $\mathbb{E}[(a-b)^2] \leq 2\mathbb{E}[a^2 + b^2]$, it suffices to show that $\mathbb{E}[(\sum_{i \in A_{n,k}} W_{n,k}^{(i)})^2]$ and $\mathbb{E}[|A_{n,k}|^2]$ are both finite. We only show the first expectation is finite; the finiteness of the second expectation then follows. Define $N_k \triangleq \sum_{n=1}^N |A_{n,k}|$ as the number of jobs (over all classes) served in frame k ; we have $|A_{n,k}| \leq N_k$ for all n and k . In frame k , the queueing delay $W_{n,k}^{(i)}$ of a class n job $i \in A_{n,k}$ is less than or equal to the busy period size B_k . Then we get $\mathbb{E}[(\sum_{i \in A_{n,k}} W_{n,k}^{(i)})^2] \leq \mathbb{E}[B_k^2 N_k^2]$. By Cauchy-Schwarz inequality, we have $\mathbb{E}[B_k^2 N_k^2] \leq \sqrt{\mathbb{E}[B_k^4] \mathbb{E}[N_k^4]}$. It suffices to show that both $\mathbb{E}[B_k^4]$ and $\mathbb{E}[N_k^4]$ are finite.

First we argue $\mathbb{E}[B_k^4] < \infty$. In the following we drop the index k for notational convenience. Since the busy period size B is the same under any work-conserving policy, it is convenient to consider LIFO scheduling with preemptive priority, and that jobs of all classes are treated equally likely. In this scheme, let a_0 denote the arrival that starts the current busy period. Arrival a_0 can be of any class, and the duration it stays in the system is equal to the busy period B . Next, let $\{a_1, \dots, a_M\}$ denote the M jobs that arrive during the service of job a_0 . Let $B(1), \dots, B(M)$ denote the durations they stay in the system. Under LIFO with preemptive priority, we observe that $B(1), \dots, B(M)$ are independent and identically distributed as the starting busy period B , since any new arrival *never sees* any previous arrivals and starts a new busy

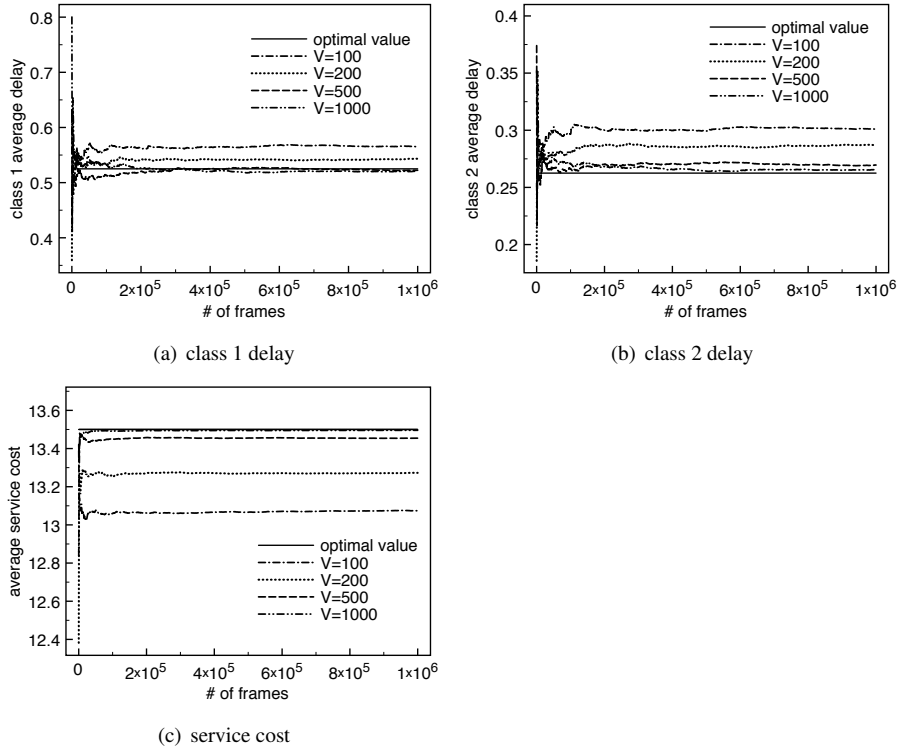


Fig. 7 Running average delay and average service cost performance of the CostDelayFair policy under different values of V .

period (by the memoryless property of Poisson arrivals). Consequently, we have

$$B = X + \sum_{m=1}^M B(m), \quad (71)$$

where X denote the service time of a_0 . Notice also that each duration $B(m)$ for all $m \in \{1, \dots, M\}$ is independent of M . By taking the square and expectation of (71), we can compute $\mathbb{E}[B^2]$ in closed form and show that it is finite if the first two moments of service times X_n are finite for all n . Likewise, by raising (71) to the third and fourth power and taking expectation, we can compute $\mathbb{E}[B^3]$ and $\mathbb{E}[B^4]$ and show they are finite if the first four moments of X_n are finite (showing $\mathbb{E}[B^4] < \infty$ requires the finiteness of the first three moments of B).

Likewise, to show $\mathbb{E}[N^4] < \infty$, under LIFO with preemptive priority we observe

$$N = 1 + \sum_{m=1}^M N(m), \quad (72)$$

where $N(m)$ denotes the number of arrivals, including a_m , served during the stay of arrival a_m in the system; $N(m)$ are i.i.d. and independent of M . By raising (72) to the

second, third, and fourth power and taking expectation, we can compute $\mathbb{E}[N^4]$ in closed form and show it is finite. \square

9.2 Proof of Theorem 1

By Lemma 3, it suffices to show that the DelayFeas policy stabilizes all $Z_{n,k}$ queues in the sense of mean rate stability. Let $(\bar{W}_1^*, \dots, \bar{W}_N^*)$ be a feasible average delay vector satisfying $\bar{W}_n^* \leq d_n$ for all n . From Lemma 1, there exists a stationary randomized priority policy π_{rand}^* that (i) randomly and independently uses a strict priority rule in each busy period according to a probability distribution, and (ii) achieves the average delay vector $(\bar{W}_1^*, \dots, \bar{W}_N^*)$. Since the DelayFeas policy minimizes (15) over all feasible scheduling decisions in a frame, comparing the DelayFeas policy with the randomized policy π_{rand}^* yields, in every frame k ,

$$\sum_{n=1}^N Z_{n,k} \rho_n \bar{W}_{n,k}^{\text{DelayFeas}} \leq \sum_{n=1}^N Z_{n,k} \rho_n \bar{W}_n^*. \quad (73)$$

From (73), the inequality (17) evaluated under the DelayFeas policy is further upper bounded by

$$\begin{aligned} \Delta(\mathbf{Z}_k) &\leq C + \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \rho_n (\bar{W}_{n,k}^{\text{DelayFeas}} - d_n) \\ &\leq C + \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \rho_n (\bar{W}_n^* - d_n) \leq C. \end{aligned}$$

Taking expectation, summing over $k \in \{0, \dots, K-1\}$, and noting $L(\mathbf{Z}_0) = 0$, we get

$$\mathbb{E}[L(\mathbf{Z}_K)] = \frac{1}{2} \sum_{n=1}^N \mathbb{E}[X_n] \mathbb{E}[(Z_{n,K})^2] \leq KC.$$

It follows that $\mathbb{E}[(Z_{n,K})^2] \leq 2KC/\mathbb{E}[X_n]$ for all job classes n , and

$$0 \leq \mathbb{E}[Z_{n,K}] \leq \sqrt{\mathbb{E}[(Z_{n,K})^2]} \leq \sqrt{\frac{2KC}{\mathbb{E}[X_n]}}, \quad n \in \{1, \dots, N\}.$$

Dividing the above by K yields

$$0 \leq \frac{\mathbb{E}[Z_{n,K}]}{K} \leq \sqrt{\frac{2C}{K\mathbb{E}[X_n]}}, \quad n \in \{1, \dots, N\}. \quad (74)$$

Passing $K \rightarrow \infty$ proves mean rate stability for all virtual queues $(Z_{1,k}, \dots, Z_{N,k})$. \square

9.3 Proof of Theorem 2

Consider the optimal stationary randomized priority policy π_{rand}^* that yields optimal average delays $\bar{W}_n^* \leq d_n$ for all classes n . Since the DelayFair policy minimizes the right side of (33) in every frame, if we compare the DelayFair policy with policy π_{rand}^* and the genie decisions $r_{n,k}^* = \bar{W}_n^*$ for all n and k , (33) under the DelayFair policy is further upper bounded by

$$\begin{aligned}
& \Delta(\mathbf{Z}_k, \mathbf{Y}_k) + V \sum_{n=1}^N \mathbb{E} [f_n(r_{n,k}) T_k \mid \mathbf{Z}_k, \mathbf{Y}_k] \\
& \leq C - \mathbb{E}[T_k] \sum_{n=1}^N Z_{n,k} \lambda_n d_n + \mathbb{E}[T_k] \sum_{n=1}^N (Z_{n,k} + Y_{n,k}) \lambda_n \bar{W}_n^* \\
& \quad + \mathbb{E}[T_k] \sum_{n=1}^N (V f_n(\bar{W}_n^*) - Y_{n,k} \lambda_n \bar{W}_n^*) \\
& \leq C + V \mathbb{E}[T_k] \sum_{n=1}^N f_n(\bar{W}_n^*). \tag{75}
\end{aligned}$$

Removing the second nonnegative term of (75) yields

$$\Delta(\mathbf{Z}_k, \mathbf{Y}_k) \leq C + V \mathbb{E}[T_k] \sum_{n=1}^N f_n(\bar{W}_n^*) \leq C + VD, \tag{76}$$

where $D \triangleq \mathbb{E}[T_k] \sum_{n=1}^N f_n(\bar{W}_n^*)$ is a finite constant. Taking expectation of (76), summing over $k \in \{0, \dots, K-1\}$, and noting $L(\mathbf{Z}_0, \mathbf{Y}_0) = 0$, we get $\mathbb{E}[L(\mathbf{Z}_K, \mathbf{Y}_K)] \leq K(C + VD)$. It follows that, for each queue $Z_{n,k}$, we have

$$0 \leq \frac{\mathbb{E}[Z_{n,K}]}{K} \leq \sqrt{\frac{\mathbb{E}[(Z_{n,K})^2]}{K^2}} \leq \sqrt{\frac{2\mathbb{E}[L(\mathbf{Z}_K, \mathbf{Y}_K)]}{K^2}} \leq \sqrt{\frac{2(C + VD)}{K}}. \tag{77}$$

Passing $K \rightarrow \infty$ proves mean rate stability of all virtual queues $(Z_{1,k}, \dots, Z_{N,k})$. Consequently, the delay constraints $\bar{W}_n \leq d_n$ are satisfied by Lemma 3. Likewise, all virtual queues $(Y_{1,k}, \dots, Y_{N,k})$ are mean rate stable.

Next, taking expectation of (75), summing over $k \in \{0, \dots, K-1\}$, dividing by V , and noting $L(\mathbf{Z}_0, \mathbf{Y}_0) = 0$, we get

$$\frac{\mathbb{E}[L(\mathbf{Z}_K, \mathbf{Y}_K)]}{V} + \sum_{n=1}^N \mathbb{E} \left[\sum_{k=0}^{K-1} f_n(r_{n,k}) T_k \right] \leq \frac{KC}{V} + \mathbb{E} \left[\sum_{k=0}^{K-1} T_k \right] \sum_{n=1}^N f_n(\bar{W}_n^*).$$

Removing the first nonnegative term and dividing the rest by $\mathbb{E}[\sum_{k=0}^{K-1} T_k]$ yield

$$\begin{aligned}
\sum_{n=1}^N \frac{\mathbb{E}[\sum_{k=0}^{K-1} f_n(r_{n,k}) T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]} & \leq \frac{KC}{V \mathbb{E}[\sum_{k=0}^{K-1} T_k]} + \sum_{n=1}^N f_n(\bar{W}_n^*) \\
& \stackrel{(a)}{\leq} \frac{C \sum_{n=1}^N \lambda_n}{V} + \sum_{n=1}^N f_n(\bar{W}_n^*), \tag{78}
\end{aligned}$$

where (a) uses $\mathbb{E}[T_k] \geq \mathbb{E}[I_k] = 1/(\sum_{n=1}^N \lambda_n)$ for all k . By a generalized Jensen's inequality [25, Lemma 7.6] and convexity of $f_n(\cdot)$, we get

$$\sum_{n=1}^N \frac{\mathbb{E}[\sum_{k=0}^{K-1} f_n(r_{n,k}) T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]} \geq \sum_{n=1}^N f_n\left(\frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]}\right). \quad (79)$$

Combining (78) and (79), and taking a limsup as $K \rightarrow \infty$, we get

$$\limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n\left(\frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]}\right) \leq \frac{C \sum_{n=1}^N \lambda_n}{V} + \sum_{n=1}^N f_n(\bar{W}_n^*).$$

The next lemma completes the proof. \square

Lemma 8 *If all $Y_{n,k}$ queues are mean rate stable, then*

$$\limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n\left(\frac{\mathbb{E}[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}]}{\mathbb{E}[\sum_{k=0}^{K-1} |A_{n,k}|]}\right) \leq \limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n\left(\frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]}\right).$$

Proof (Lemma 8) From (27), we get

$$Y_{n,k+1} \geq Y_{n,k} - r_{n,k} |A_{n,k}| + \sum_{i \in A_{n,k}} W_{n,k}^{(i)}.$$

Summing over $k \in \{0, \dots, K-1\}$ and using $Y_{n,0} = 0$ yield

$$\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} - Y_{n,K} \leq \sum_{k=0}^{K-1} r_{n,k} |A_{n,k}|.$$

Taking expectation and dividing by $\lambda_n \mathbb{E}[\sum_{k=0}^{K-1} T_k]$ yield

$$\frac{\mathbb{E}[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}]}{\lambda_n \mathbb{E}[\sum_{k=0}^{K-1} T_k]} - \frac{\mathbb{E}[Y_{n,K}]}{\lambda_n K \mathbb{E}[T_0]} \leq \frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} |A_{n,k}|]}{\lambda_n \mathbb{E}[\sum_{k=0}^{K-1} T_k]}, \quad (80)$$

where in the second term we use $\mathbb{E}[T_k] = \mathbb{E}[T_0]$ for all k . In the last term of (80), since the value $r_{n,k}$ is chosen independent of $|A_{n,k}|$ and T_k , we use $\mathbb{E}[|A_{n,k}|] = \lambda_n \mathbb{E}[T_k]$ and get

$$\frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} |A_{n,k}|]}{\lambda_n \mathbb{E}[\sum_{k=0}^{K-1} T_k]} = \frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]}.$$

Define $\theta_K^{(n)}$ as the left side of (80). Then we have

$$\theta_K^{(n)} \leq \frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]}.$$

Since $f_n(\cdot)$ is nondecreasing for all classes n , we get

$$\limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n(\theta_K^{(n)}) \leq \limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n\left(\frac{\mathbb{E}[\sum_{k=0}^{K-1} r_{n,k} T_k]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k]}\right). \quad (81)$$

Using (80), define the value

$$\eta_K^{(n)} \triangleq \frac{\mathbb{E} \left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} \right]}{\mathbb{E} \left[\sum_{k=0}^{K-1} |A_{n,k}| \right]} = \theta_K^{(n)} + \frac{\mathbb{E} [Y_{n,K}]}{\lambda_n K \mathbb{E} [T_0]}. \quad (82)$$

To complete the proof, from (81), it suffices to show

$$\limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n(\eta_K^{(n)}) = \limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n(\theta_K^{(n)}). \quad (83)$$

We show that inequality \leq holds in (83); the other direction is proved similarly. Let the left side of (83) attain its limsup in the subsequence $\{K_m\}_{m=1}^\infty$. It follows

$$\begin{aligned} \limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n(\eta_K^{(n)}) &= \lim_{m \rightarrow \infty} \sum_{n=1}^N f_n(\eta_{K_m}^{(n)}) \stackrel{(a)}{=} \sum_{n=1}^N f_n \left(\lim_{m \rightarrow \infty} \eta_{K_m}^{(n)} \right) \\ &\stackrel{(b)}{=} \sum_{n=1}^N f_n \left(\lim_{m \rightarrow \infty} \theta_{K_m}^{(n)} \right) \leq \limsup_{K \rightarrow \infty} \sum_{n=1}^N f_n(\theta_K^{(n)}), \end{aligned}$$

where (a) follows the continuity of $f_n(\cdot)$ for all classes n , and (b) follows (82) and mean rate stability of the $Y_{n,k}$ queues. \square

9.4 Independence of second-order statistics in the DynRate policy

We show how to remove the dependence on the second moments of job sizes S_n in the DynRate policy in Section 6.2. For simplicity, we assume that job classes are properly re-ordered so that class n has the n th highest priority for $n \in \{1, \dots, N\}$. We rewrite (38) using (8) as

$$\widehat{R} \left[\left(\frac{V}{\widehat{R}} \sum_{n=1}^N \lambda_n \mathbb{E} [S_n] \right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^N \frac{Z_{n,k} \lambda_n}{(\mu(P_k) - \sum_{m=0}^{n-1} \widehat{\rho}_m)(\mu(P_k) - \sum_{m=0}^n \widehat{\rho}_m)} \right] \quad (84)$$

where

$$\widehat{R} \triangleq \frac{1}{2} \sum_{n=1}^N \lambda_n \mathbb{E} [S_n^2], \quad \widehat{\rho}_m \triangleq \begin{cases} \lambda_m \mathbb{E} [S_m], & 1 \leq m \leq N \\ 0, & m = 0 \end{cases}$$

By ignoring the constant \widehat{R} and redefining $\widetilde{V} \triangleq V/\widehat{R}$ in (84), running the DynRate policy in the k th frame is equivalent to allocating service rate $\mu(P_k)$ that minimizes

$$\left(\widetilde{V} \sum_{n=1}^N \lambda_n \mathbb{E} [S_n] \right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^N \frac{Z_{n,k} \lambda_n}{(\mu(P_k) - \sum_{m=0}^{n-1} \widehat{\rho}_m)(\mu(P_k) - \sum_{m=0}^n \widehat{\rho}_m)},$$

which is independent of second moments of job sizes. The control parameter \widetilde{V} can be chosen to be a large value without the knowledge of second moments of job sizes. From Theorem 3 and using $V = \widetilde{V} \widehat{R}$, this alternative policy yields average service cost \bar{P} satisfying

$$\bar{P} \leq \frac{C \sum_{n=1}^N \lambda_n}{\widetilde{V} \widehat{R}} + P^*,$$

and the property that the performance gap, which is $O(1/\widetilde{V})$, can be made arbitrarily small by a sufficiently large \widetilde{V} is preserved.

9.5 Proof of Theorem 3

The cost-delay performance region Λ is defined by the set $\widehat{\Pi}$ of control policies described in Section 6.1. It is useful to consider a stationary randomized control policy π_{rand} that: (i) randomly allocates a fixed service rate in a busy period according to a stationary distribution, and (ii) randomly uses a strict priority rule in each busy period according to a probability distribution that may depend on the service rate in that period. Then, for each control policy $\pi \in \widehat{\Pi}$, there exists a randomized policy π_{rand} inducing the same performance. To see this, let $\mathcal{W}(P)$ be the delay performance region when the service rate $\mu(P)$ is used in all busy periods. Under a policy $\pi \in \widehat{\Pi}$, let \mathbf{w}_P be the average delay vector of the multi-class jobs served in the subset of busy periods with service rate $\mu(P)$. From Lemma 1, we have $\mathbf{w}_P \in \mathcal{W}(P)$ and the delay vector \mathbf{w}_P is achieved by a randomized priority policy. Since the service rate allocations of policy π are ergodic decisions, the overall average delay vector \mathbf{w} of policy π is a convex combination of the delay vectors $\{\mathbf{w}_P\}_{P \in [P_{\min}, P_{\max}]}$. The randomized policy π_{rand} that induces the same performance as the policy $\pi \in \widehat{\Pi}$ works as follows: (i) randomly allocating a service rate in each busy period according to a stationary distribution defined by the limiting frequencies of service rate allocations in π ; (ii) in a busy period with service rate $\mu(P)$, using the randomized priority policy that induces \mathbf{w}_P . Also, since every randomized policy π_{rand} described above belongs to the policy space $\widehat{\Pi}$, optimizing the queue performance over the cost-delay performance region Λ is equivalent to optimizing it over the set of stationary randomized policies π_{rand} .

Consider the optimal stationary randomized policy π_{rand}^* that yields optimal average cost P^* and feasible average queueing delay $\bar{W}_n^* \leq d_n$ for all job classes n . Let P_k^* denote its service cost allocation in the k th frame. Since policy π_{rand}^* makes i.i.d. decisions over frames, by renewal reward theory we have

$$P^* = \frac{\mathbb{E}[P_k^* B(P_k^*)]}{\mathbb{E}[T(P_k^*)]}.$$

The ratio (43) under policy π_{rand}^* is equal to (see (48))

$$V \frac{\mathbb{E}[P_k^* B(P_k^*)]}{\mathbb{E}[T(P_k^*)]} + \sum_{n=1}^N Z_{n,k} \lambda_n (\bar{W}_n^* - d_n) \leq VP^*. \quad (85)$$

Since the DynRate policy minimizes (43) over frame-based policies that update controls over busy periods, which include the optimal policy π_{rand}^* , (43) under the DynRate policy satisfies, from (85),

$$\frac{\Phi(\mathbf{Z}_k)}{\mathbb{E}[T_k(P_k) | \mathbf{Z}_k]} \leq VP^*, \quad k \in \mathbb{Z}^+.$$

Using this bound, (42) under the DynRate policy satisfies

$$\Delta(\mathbf{Z}_k) + V \mathbb{E}[P_k B_k(P_k) | \mathbf{Z}_k] \leq C + VP^* \mathbb{E}[T_k(P_k) | \mathbf{Z}_k].$$

Taking expectation, summing over $k \in \{0, \dots, K-1\}$, and noting $L(\mathbf{Z}_0) = 0$ yield

$$\mathbb{E}[L(\mathbf{Z}_K)] + V \sum_{k=0}^{K-1} \mathbb{E}[P_k B_k(P_k)] \leq KC + VP^* \mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right]. \quad (86)$$

Since $\mathbb{E}[T_k(P_k)]$ is decreasing in P_k and is independent of scheduling policies under a fixed service rate, we get $\mathbb{E}[T_k(P_k)] \leq \mathbb{E}[T_0(P_{\min})]$ for all k . It follows that

$$\mathbb{E}[L(\mathbf{Z}_K)] + V \sum_{k=0}^{K-1} \mathbb{E}[P_k B_k(P_k)] \leq K(C + VP^* \mathbb{E}[T_0(P_{\min})]).$$

Removing the second term and dividing by K^2 yield

$$\frac{\mathbb{E}[L(\mathbf{Z}_K)]}{K^2} \leq \frac{C + VP^* \mathbb{E}[T_0(P_{\min})]}{K}.$$

By combining it with

$$0 \leq \frac{\mathbb{E}[Z_{n,K}]}{K} \leq \sqrt{\frac{\mathbb{E}[(Z_{n,K})^2]}{K^2}} \leq \sqrt{\frac{2\mathbb{E}[L(\mathbf{Z}_K)]}{K^2}}, \quad \forall n \in \{1, \dots, N\}$$

and passing $K \rightarrow \infty$, we prove that all $Z_{n,k}$ queues are mean rate stable. All delay requirements $\bar{W}_n \leq d_n$ are therefore satisfied by Lemma 3.

Next, removing the first nonnegative term in (86) and dividing the result by $V \mathbb{E}[\sum_{k=0}^{K-1} T_k(P_k)]$ yield

$$\frac{\mathbb{E}[\sum_{k=0}^{K-1} P_k B_k(P_k)]}{\mathbb{E}[\sum_{k=0}^{K-1} T_k(P_k)]} \leq \frac{C}{V} \frac{K}{\mathbb{E}[\sum_{k=0}^{K-1} T_k(P_k)]} + P^* \stackrel{(a)}{\leq} \frac{C \sum_{n=1}^N \lambda_n}{V} + P^*,$$

where (a) uses $\mathbb{E}[T_k(P_k)] \geq \mathbb{E}[I_k] = 1/(\sum_{n=1}^N \lambda_n)$. Passing $K \rightarrow \infty$ finishes the proof. \square

9.6 Proof of Lemma 6

From (53), we have $X_{k+1} \geq X_k + P_k B_k(P_k) - P_{\text{const}} T_k(P_k)$. Summing over $k \in \{0, \dots, K-1\}$, taking expectation, and using $X_0 = 0$, we get

$$\mathbb{E}[X_K] \geq \mathbb{E}\left[\sum_{k=0}^{K-1} P_k B_k(P_k)\right] - P_{\text{const}} \mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right].$$

Dividing by $\mathbb{E}[\sum_{k=0}^{K-1} T_k(P_k)]$ and passing $K \rightarrow \infty$, we obtain

$$\bar{P} \leq P_{\text{const}} + \limsup_{K \rightarrow \infty} \frac{\mathbb{E}[X_K]}{K} \frac{K}{\mathbb{E}[\sum_{k=0}^{K-1} T_k(P_k)]} \stackrel{(a)}{\leq} P_{\text{const}} + \limsup_{K \rightarrow \infty} \frac{\mathbb{E}[X_K]}{K} \sum_{n=1}^N \lambda_n \stackrel{(b)}{=} P_{\text{const}}$$

where (a) uses $\mathbb{E}[T_k(P_k)] \geq \mathbb{E}[I_k] = 1/(\sum_{n=1}^N \lambda_n)$, and (b) follows mean rate stability of the virtual cost queue X_k . \square

9.7 Proof of Theorem 4

Let π_{rand}^* be the stationary randomized policy of service rate control and priority assignment that solves (50)-(52). Let $(\bar{W}_n^*)_{n=1}^N$ be the optimal mean delay vector, and \bar{P}^* the associated average service cost satisfying $\bar{P}^* \leq P_{\text{const}}$. In the k th frame, the ratio $\frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k) | \chi_k]}$ under the policy π_{rand}^* and the genie decision $r_{n,k}^* = \bar{W}_n^*$ for all n and k satisfies

$$\begin{aligned} \frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k) | \chi_k]} &= \sum_{n=1}^N Y_{n,k} \lambda_n \bar{W}_n^* + X_k \bar{P}^* - X_k P_{\text{const}} \\ &\quad + \sum_{n=1}^N (V f_n(\bar{W}_n^*) - Y_{n,k} \lambda_n \bar{W}_n^*) \leq V \sum_{n=1}^N f_n(\bar{W}_n^*). \end{aligned} \quad (87)$$

Since the CostDelayFair policy minimizes $\frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k) | \chi_k]}$ in every frame, this ratio under the CostDelayFair policy satisfies, by (87),

$$\frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k) | \chi_k]} \leq V \sum_{n=1}^N f_n(\bar{W}_n^*).$$

Then (56) under the CostDelayFair policy satisfies

$$\Delta(\chi_k) + V \mathbb{E} \left[\sum_{n=1}^N f_n(r_{n,k}) T_k(P_k) | \chi_k \right] \leq C + V \mathbb{E}[T_k(P_k) | \chi_k] \sum_{n=1}^N f_n(\bar{W}_n^*). \quad (88)$$

Removing the second term in (88) and taking expectation, we get

$$\mathbb{E}[L(\chi_{k+1})] - \mathbb{E}[L(\chi_k)] \leq C + V \mathbb{E}[T_k(P_k)] \sum_{n=1}^N f_n(\bar{W}_n^*).$$

Summing over $k \in \{0, \dots, K-1\}$, and using $L(\chi_0) = 0$ yields

$$\mathbb{E}[L(\chi_K)] \leq KC + V \mathbb{E} \left[\sum_{k=0}^{K-1} T_k(P_k) \right] \sum_{n=1}^N f_n(\bar{W}_n^*) \leq KC_1 \quad (89)$$

where $C_1 \triangleq C + V \mathbb{E}[T_0(P_{\min})] \sum_{n=1}^N f_n(\bar{W}_n^*)$ and we have used $\mathbb{E}[T_k(P_k)] \leq \mathbb{E}[T_0(P_{\min})]$ for all k . Inequality (89) suffices to conclude that the virtual cost queue X_k and all $Y_{n,k}$ queues are mean rate stable. From Lemma 6 we have $\bar{P} \leq P_{\text{const}}$. The proof of (54) follows that of Theorem 2. \square

10 Concluding remarks

We revisit the classical problem of optimal stochastic scheduling in a multi-class $M/G/1$ queue with nonpreemptive service. We study four delay-optimal priority scheduling and dynamic service rate control problems, and cast them as convex optimization problems with side constraints. We solve these problems by developing simple adaptive $c\mu$ rules that use per-class running delay performance to greedily

re-prioritize job classes over busy periods. The near-optimal performance of these online policies is proved, and further validated through simulations. Technically, we showcase that the Lyapunov drift analysis, typically used to establish system stability and to design throughput-optimal policies in queueing systems, can be utilized as a new methodology to solve stochastic convex optimization problems with respect to other performance metrics such as delay in dynamic systems.

There are some interesting future research directions. There are many other multi-class queueing systems that have polymatroidal performance regions, such as those discussed in [7, 37]. Our method may be applicable to solving interesting convex optimization problems and developing online policies there. Next, frame-based policies that update controls over busy periods are sometimes considered impractical due to large variance incurred in performance, especially when the system is heavily loaded. Using our methodology to develop job-level adaptive control policies with small variance is another interesting future work.

References

1. Ansell, P.S., Glazebrook, K.D., Niño-Mora, J., O’Keeffe, M.: Whittle’s index policy for a multi-class queueing system with convex holding costs. *Math. Methods of Oper. Res.* **57**, 21–39 (2003)
2. Bansal, N., Kimbrel, T., Pruhs, K.: Speed scaling to manage energy and temperature. *Journal of the ACM* **54**(1) (2007)
3. Baras, J.S., Dorsey, A.J., Makowski, A.M.: Two competing queues with linear costs and geometric service requirements: The μc -rule is often optimal. *Adv. Appl. Probab.* **17**(1), 186–209 (1985)
4. Bertsekas, D.P.: *Nonlinear Programming*, 2nd edn. Athena Scientific (1999)
5. Bertsekas, D.P., Gallager, R.G.: *Data Networks*, 2nd edn. Prentice Hall (1992)
6. Bertsimas, D.: The achievable region method in the optimal control of queueing systems; formulations, bounds, and policies. *Queueing Syst.* **21**(3-4), 337–389 (1995)
7. Bertsimas, D., Niño-Mora, J.: Conservation laws, extended polymatroids, and multiarmed bandit problems; a polyhedral approach to indexable systems. *Math. of Oper. Res.* **21**(2), 257–306 (1996)
8. Bhattacharya, P.P., Georgiadis, L., Tsoucas, P.: Problems of adaptive optimization in multiclass M/G/1 queues with bernoulli feedback. *Math. of Oper. Res.* **20**(2), 355–380 (1995)
9. Buyukkoc, C., Varaiya, P., Walrand, J.: The $c\mu$ rule revisited. *Adv. Appl. Probab.* **17**, 237–238 (1985)
10. Federgruen, A., Groenevelt, H.: Characterization and optimization of achievable performance in general queueing systems. *Oper. Res.* **36**(5), 733–741 (1988)
11. Federgruen, A., Groenevelt, H.: M/G/c queueing systems with multiple customer classes: Characterization and control of achievable performance under nonpreemptive priority rules. *Manage. Sci.* **34**(9), 1121–1138 (1988)
12. Folland, G.B.: *Real Analysis: Modern Techniques and Their Applications*, 2nd edn. Wiley (1997)
13. Gelenbe, E., Mitrani, I.: *Analysis and Synthesis of Computer Systems*, 2nd edn. Imperial College Press (2010)
14. George, J.M., Harrison, J.M.: Dynamic control of a queue with adjustable service rate. *Oper. Res.* **49**(5), 720–731 (2001)
15. Georgiadis, L., Neely, M.J., Tassiulas, L.: Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking* **1**(1) (2006)
16. Glazebrook, K.D., Lumley, R.R., Ansell, P.S.: Index heuristics for multiclass M/G/1 systems with nonpreemptive service and convex holding costs. *Queueing Syst.* **45**(2), 81–111 (2003)
17. Gurvich, I., Whitt, W.: Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing Service Operations Management* **11**(2), 237–253 (2009)
18. Hou, I.H., Kumar, P.R.: Queueing systems with hard delay constraints: a framework for real-time communication over unreliable wireless channels. *Queueing Syst.* **71**, 151–177 (2012)
19. Hou, I.H., Truong, A., Chakraborty, S., Kumar, P.R.: Optimality of periodwise static priority policies in real-time communications. In: *IEEE Conf. Decision and Control (CDC)* (2011)

20. Kelly, F.P.: Charging and rate control for elastic traffic. *European Trans. Telecommunications* **8**, 33–37 (1997)
21. Kleinrock, L.: *Queueing Systems, vol. II: Computer Applications*. Wiley Interscience (1976)
22. Li, C.P., Neely, M.J.: Network utility maximization over partially observable Markovian channels. *Performance Evaluation* (2012). Accepted for publication
23. Mandelbaum, A., Stolyar, A.L.: Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$ -rule. *Oper. Res.* **52**(6), 836–855 (2004)
24. van Mieghem, J.A.: Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *Ann. Appl. Probab.* **5**(3), 809–833 (1995)
25. Neely, M.J.: *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool (2010)
26. Niño-Mora, J.: Stochastic scheduling. In: C.A. Floudas, P.M. Pardalos (eds.) *Encyclopedia of Optimization*, 2nd edn., pp. 3818–3824. Springer (2009)
27. Ross, K.W., Chen, B.: Optimal scheduling of interactive and noninteractive traffic in telecommunication systems. *IEEE Trans. Autom. Control* **33**(3), 261–267 (1988)
28. Ross, K.W., Yao, D.D.: Optimal dynamic scheduling in Jackson networks. *IEEE Trans. Autom. Control* **34**(1), 47–53 (1989). DOI 10.1109/9.8648
29. Ross, S.M.: *Stochastic Processes*, 2 edn. Wiley (1996)
30. Shanthikumar, J.G., Yao, D.D.: Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Oper. Res.* **40**(2), S293–S299 (1992)
31. Stidham, S., Weber, R.: Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Oper. Res.* **37**(4), 611–625 (1989)
32. L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
33. ———, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.
34. Walrand, J.: *An Introduction to Queueing Networks*. Prentice Hall (1988)
35. Wang, W.H., Palaniswami, M., Low, S.H.: Application-oriented flow control: Fundamentals, algorithms, and fairness. *IEEE/ACM Trans. Netw.* **14**(6), 1282–1291 (2006)
36. Welsh, D.J.A.: *Matroid Theory*. Academic Press, London, UK (1976)
37. Yao, D.D.: Dynamic scheduling via polymatroid optimization. In: *Performance Evaluation of Complex Systems: Techniques and Tools*, Performance 2002, Tutorial Lectures, pp. 89–113. Springer-Verlag, London, UK (2002)