Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 167 (2005) 179–207

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Stochastics and Statistics

Queueing systems with leadtime constraints: A fluid-model approach for admission and sequencing control

Constantinos Maglaras [a], Jan A. Van Mieghem [b,*]

[a] *Graduate School of Business, Columbia University, 409 Uris Hall, New York, NY 10027-6902, USA*
[b] *Kellogg Graduate School of Management, Department of Managerial Economics and Decision Sciences, Northwestern University, 2001 Sheridan Road, Evanston, IL 60208-2009, USA*

Received 2 February 2001; accepted 18 March 2004
Available online 18 May 2004

Abstract

We study how multi-product queueing systems should be controlled so that sojourn times (or end-to-end delays) do not exceed specified leadtimes. The network dynamically decides when to admit new arrivals and how to sequence the jobs in the system. To analyze this difficult problem, we propose an approach based on fluid-model analysis that translates the leadtime specifications into deterministic constraints on the queue length vector. The main benefit of this approach is that it is possible (and relatively easy) to construct scheduling and multi-product admission policies for leadtime control. Additional results are: (a) While this approach is simpler than a heavy-traffic approach, the admission policies that emerge from it are also more specific than, but consistent with, those from heavy-traffic analysis. (b) A simulation study gives a first indication that the policies also perform well in stochastic systems. (c) Our approach specifies a "tailored" admission region for any given sequencing policy. Such joint admission and sequencing control is "robust" in the following sense: system performance is relatively insensitive to the particular choice of sequencing rule when used in conjunction with tailored admission control. As an example, we discuss the tailored admission regions for two well-known sequencing policies: *Generalized Processor Sharing* and *Generalized Longest Queue*. (d) While we first focus on the multi-product single server system, we do extend to networks and identify some subtleties.

Keywords: Queueing; Scheduling; Lead times; Admission control; Fluid models

1. Introduction

We study how multi-product queueing systems should be controlled so that sojourn times—also called end-to-end delays, flow times, or throughput times—do not exceed specified leadtimes. Such systems are of obvious interest in manufacturing and service operations settings because they guarantee that due-dates, quoted as job arrival time plus leadtime, are met. Systems that can guarantee that flow times do not exceed

[*] Corresponding author.
E-mail addresses: c.maglaras@columbia.edu (C. Maglaras), vanmieghem@kellogg.northwestern.edu (J.A. Van Mieghem).

0377-2217/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2002.11.005

a specified upper bound also have become a much discussed topic in communication networks. The convergence of voice and data networks has led to different applications—each with different delay requirements—sharing the same network resources. In both settings, an important question is whether, and if so how, a multi-product network can guarantee differentiated "quality of service" specifications in the sense that flow times of different product types do not exceed product-specific leadtimes. This article aims to offer some answers to this question.

We consider a multi-class queueing network that is shared by many products or "job types" that differ in their arrival rates, processing requirements and routes that they follow through the system. Each product i's flow time must not exceed its product-specific leadtime D_i. Let d_i be the random variable that denotes the actual product i delay. Ideally, the network would like to ensure that $d_i \leqslant D_i$ for all products i. In reality, due to the inherent variability in stochastic networks, these service guarantees should be interpreted and expressed in a probabilistic form: the probability of violating the leadtime constraint should be small: $\mathbf{P}(d_i > D_i) \leqslant \epsilon_{d,i}$. Clearly there is a trade-off between D_i and $\epsilon_{d,i}$: small leadtimes are harder to satisfy and yield larger $\epsilon_{d,i}$. In addition, one should also consider blocking or admission control, which brings a second trade-off: small leadtimes are easier to satisfy with stricter admission control. Indeed, new arrivals when the system is heavily congested are more likely to exceed their delay bounds and the network is better off denying admission, if possible. Let b_i denote the product i blocking probability. Thus, probabilistic leadtime guarantees could be specified in terms of an exogenous parameter triplet $(D, \epsilon_d, \epsilon_b)$, where D, ϵ_d, and ϵ_b are non-negative vectors, as follows:

$$\mathbf{P}(d_i > D_i) \leqslant \epsilon_{d,i} \quad \text{and} \quad b_i \leqslant \epsilon_{b,i}, \quad \forall \text{ products } i. \tag{1}$$

Finally, a third trade-off in multi-product systems derives from sequencing: small leadtimes for product i are harder to satisfy if product j gets network priority.

Unfortunately, addressing all three trade-offs through joint admission and sequencing control in stochastic networks is not amenable to analytic study. Therefore, we propose to analyze dynamic control in the simpler deterministic and continuous fluid network. Our approach hinges on a simple articulation of the leadtime specifications in terms of deterministic, linear constraints on the queue length vector. In that setting, we can successfully analyze the admission and sequencing trade-off and construct multi-product admission and sequencing control policies that guarantee a given leadtime vector D. The intent of this paper is to lay out a basic fluid-model approach to deal with leadtime constraints through admission and sequencing and to illustrate the potential insights and applications of that approach. We highlight five:

1. This approach is simple yet effective. Indeed, the admission policies that emerge from it are not only consistent with, but also more specific than, those from more involved heavy-traffic analysis. (Heavy traffic typically yields policies that control admissions on an aggregate workload basis. For moderate traffic we propose true multi-product admission control, which is largely unexplored in the literature, and show that this conforms with aggregate workload admission in the heavy-traffic limit.) Surprisingly, as Corollary 4 will show, this fluid approach also prescribes policy parameter selection that is asymptotically optimal in heavy traffic.
2. This approach suggests specific control policies that could serve as a starting point for policy construction in the stochastic network. Indeed, a simulation study in this article will give a first indication that the policies derived through this approach perform well in the stochastic network; that is, they yield small violation probabilities ϵ_d. A follow-up probabilistic study would be needed to fully incorporate the third probabilistic trade-off.
3. This approach provides a characterization of the largest admission region (over all sequencing rules) in which the system can admit jobs and still guarantee that all jobs satisfy their delay constraints. The dynamic sequencing policy that achieves this maximum admission region is a hybrid between *Generalized Longest Queue* (GLQ) and *Shortest Delay First* (Proposition 2).

4. This approach shows how admission policies depend on the sequencing policy employed and specify a tailored admission region for any given sequencing rule. This is the largest region in which the system can admit jobs and still guarantee the desired delay bounds in the fluid model when using that sequencing rule. Simulations of our policies in the stochastic system confirm that tailored admission control compensates for performance differences that stem from the effectiveness of the sequencing rules alone. Thus, tailored admission and sequencing control provides "robust" system performance. As an example, we discuss the tailored admission regions for two well-known sequencing policies: *Generalized Processor Sharing* (GPS) and GLQ (Propositions 3 and 5).
5. The constraint formulation and some of our findings on sequencing and admission control extend to multi-class queueing networks. The approach also identifies possible complications that require special care (Proposition 9).

The rest of the paper is structured as follows. We conclude this section with a literature survey. Section 2 describes the multi-class single server system, its associated fluid model and develops the tractable delay constraint formulation that is used thereafter. Section 3 studies the fluid analysis of joint admission and sequencing control under delay constraints. Section 4 compares the performance of the policies that emerge from the fluid analysis in a simulation experiment and suggests some analytical results. Section 5 extends the delay constraint formulation to the multi-class network setting and provides some preliminary results on sequencing and admission control. We conclude in Section 6.

The literature on network control with delay objectives stems from two largely disconnected groups. Operations research has a long history on job shop scheduling, due-date setting, and tardiness objectives, as reviewed by, for example, Graves [14], Wein [35], Wein and Chevalier [36], Duenyas [10] and Spearman and Zhang [29]. Related to our sensitivity findings, Wein's [34] simulations of semiconductor manufacturing suggest that admission control impacts performance more than sequencing. We will also relate to Kanban and CONWIP policies [17], which include admission control on total workload. Lawler et al. [23] review advances in combinatorial optimization and sequencing for static, deterministic (mostly single server) systems. One important distinction with our work is that we do not consider determining the leadtimes D, but rather take the lead-times as given and focus on control to achieve those hard delay constraints. In contrast, other work typically incorporates delay objectives as soft constraints or indirectly as part of an objective function that guides the design of good control policies. (An exception are production-inventory systems with probabilistic service guarantees on fill rates or stock-out probabilities; see, for example, [2,12,13].) A second distinction is in the information structure of the models under investigation: the control policies in most prior work require arrival time or "age" information, whereas our modelling and control framework does not.

The second group of literature is from the engineering field of communications. A comprehensive overview is published by IEEE [11] and our work relates to the following three areas. The first concerns networks with deterministic service guarantees developed by Cruz [6,7]. The second area deals with the concept of "effective bandwidth" that specifies how much more capacity (bandwidth) a system should devote to an inflow with nominal traffic intensity ρ in order to guarantee that $\mathbf{P}(d > D) \leqslant \epsilon$; e.g., see Kelly [19]. This will be useful in our analysis in Section 4. The third area draws on large deviations theory, which has been used successfully to study asymptotic queueing phenomena in communications. For example, Bertsimas et al. [4], Stolyar and Ramanan [30] and Zhang [37] analyze multi-class $G/G/1$ queues; see [4] for a summary of the related literature.

Our paper builds on a body of work developed over the past 10–15 years that addresses stochastic network control problems through a hierarchy of approximating models that use fluid or Brownian approximations. The original inspiration for our work is the following observation by Harrison [15, p. 86]: "rigid constraints on total delay can be incorporated in heavy-traffic formulations of network control problems, although such constraints make no sense in conventional formulations. That is, an upper bound constraint on the total delay experienced by arriving jobs of a given type can be represented in the limiting Brownian control problem as an upper bound constraint on a certain positive linear combination of queue lengths". (This follows from Reiman's "snapshot principle," c.f. [28].) Formulating scheduling objectives in terms of delay is appealing because it is often easier to articulate delay bounds than the traditional objective in terms of holding costs. While delay formulations are typically much harder (because of the state-space explosion), they do not cause serious difficulties in heavy-traffic analysis, as demonstrated in [32]. The idea of rigid constraint formulation is not developed in [15, p. 86], but a proposal for the multi-class single server was put forward in [16] and is analyzed by Plambeck et al. [27]. That paper and ours are complementary studies of a similar constraint representation of delay objectives. Plambeck et al. address the delay-blocking trade-off in a single server through a profitability criterion (maximize rewards subject to delay constraints) and analyze an admission and sequencing policy under heavy-traffic conditions. In contrast, our approach is based on a fluid analysis and can be used to specify a tailored admission policy for any sequencing policy. The admission and sequencing policies that emerge from our analysis have more specific structure than those derived from heavy-traffic analysis. Their differences, however, become negligible in the heavy-traffic limit which reflects the relative merits of both approaches: the simpler fluid analysis may retain more detailed policy structure, while Brownian analysis retains stochastic structure that allows performance analysis.

Our analysis eventually reduces to a fluid-control problem with polytopic constraints on the state. Such problems have been addressed quite extensively in the work by Lu [22]. Specifically, Lu studied fluid-control problems with upper bound constraints on the queue length vector under a cost minimization criterion. In this context, he explicitly characterized the (fluid) optimal sequencing rule. Our work does not include a cost structure and in that does not derive sequencing rules that are optimal with respect to some associated cost criterion. Instead, our focus is on admission control and, specifically, the following two questions:

1. What is the largest admission region possible in the fluid model (Proposition 2), and what is the sequencing rule that achieves it (Proposition 2)?
2. What are the admission regions that correspond to commonly used sequencing rules (Section 3.2)?

While the objective of the fluid-control problems in our paper differs from those considered by Lu, his work provides the necessary background and techniques for extending our results to also incorporate a cost criterion in the problem formulation. That extension will be addressed in future work.

Finally, the GLQ policy discussed here is also analyzed by Bertsimas et al. [4], Plambeck et al. [27], Stolyar and Ramanan [30] and Van Mieghem [33].

2. The multi-product single server system with leadtime constraints

Consider a single-server station processing I products (or job-types), indexed by $i \in \mathbb{I} = \{1, \ldots, I\}$; the terms, products or types, will be used interchangeably. (Refer to Fig. 1 for an example with two products.) Exogenous arrivals for product i enter the network according to a renewal process with rate λ_i. Infinite storage size buffers are associated with each type, jobs within a class are served FIFO, [1] and preemptive-resume type of service is assumed. Service time requirements for type i jobs are i.i.d., drawn from some general distribution with mean m_i. As usual, service rates are denoted by $\mu_i = 1/m_i$, so that the nominal

[1] In our model, once jobs are admitted, they must be served, which is guaranteed by in-class FIFO service. This prevents policies that would neglect or discard waiting jobs close or beyond their delay bound by serving more recent arrivals first.

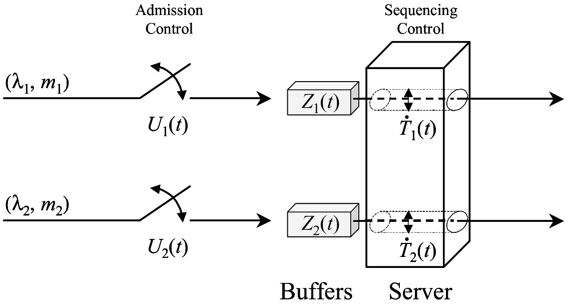

Fig. 1. A multi-class single server with admission and sequencing control.

load at the server due to type i jobs is given by $\rho_i = \lambda_i/\mu_i$. Hereafter, we will assume that the total traffic intensity $\sum_i \rho_i$ is less than one so that the server has enough capacity to process the incoming traffic. [2] The interarrival and service time sequences are mutually independent.

The system manager controls admissions to, and sequencing in, the system. An admission control policy takes the form of an I-dimensional cumulative admitted arrivals process $\{U(t), t \geqslant 0; U(0) \geqslant 0\}$, where $U_i(t)$ denotes the number of product i arrivals admitted up to time t. Thus, U_i increases in unit jumps whenever an arriving job is admitted in the system. The I-vector of queue lengths at time t is denoted by $Z(t)$, and the initial queue length configuration is denoted by $Z(0) = z$. The corresponding workload vectors are denoted by $W(t)$. A sequencing policy takes the form of an I-dimensional cumulative allocation process $\{T(t), t \geqslant 0; T(0) = 0\}$, where $T_i(t)$ denotes the cumulative time that the server has allocated to serving type i jobs up to time t. We allow server splitting, which means that at any given time the server can divide its effort into fractional allocations devoted to processing types $i \in \mathbb{I}$, denoted by $\dot{T}_i(t)$. For each type, all processing effort goes to the job at the top of the queue. The cumulative allocation process should be non-decreasing with $T(0) = 0$, $\dot{T}(t) \geqslant 0$ and $\sum_i \dot{T}_i(t) \leqslant 1$. Finally, both controls must be non-anticipating; that is, current decisions should only depend on information available up to time t. In summary, a control policy is a pair of controls $\{(T(t), U(t)), t \geqslant 0\}$ that satisfy the conditions listed above.

The objective is to control this system while satisfying the probabilistic leadtime guarantees stated in (1). Without loss of generality, assume that products are labelled so that $D_1 \leqslant D_2 \leqslant \cdots \leqslant D_I$. Recall that the random variable d_i denotes the actual total time spent in the system by a product i admitted arrival (this is the end-to-end delay that includes waiting and service time). A first step towards a tractable articulation of the leadtime constraints in terms of state variables such as queue length, workload and control processes is found as follows. Consider the extreme (and not achievable) case where all type i jobs leave the system within their desired leadtime of D_i time units (that is, $\epsilon_d = 0$). The following key observation holds:

$$\begin{aligned} d_i \leqslant D_i &\iff \text{all type } i \text{ jobs in system at time } t \text{ arrived no longer than } D_i \text{ time units ago,} \\ &\iff \forall t \geqslant D_i : Z_i(t) \leqslant U_i(t) - U_i(t - D_i) \text{ a.s.,} \qquad (2) \\ &\iff \forall t \geqslant D_i : W_i(t - D_i) \leqslant T_i(t) - T_i(t - D_i) \text{ a.s.} \qquad (3) \end{aligned}$$

That is, a system that guarantees the leadtime constraints with probability one, would satisfy (2) and (3). These conditions become more useful when analyzing their "fluid analog". Specifically, consider the fluid

[2] Since the system has admission control capability, one could allow for the total load to exceed the nominal capacity ($\sum_i \rho_i \geqslant 1$), impose some revenue structure for accepting jobs and solve a revenue maximization problem subject to the leadtime constraints. While such revenue optimization may enhance performance (at least in a transient mode where the demand characteristics change and exceed capacity) it is beyond the scope of this paper.

model associated with the single server system subject to the deterministic leadtime specifications: $d_i \leqslant D_i$ for all types i. This fluid model is a system with deterministic and continuous dynamics that is described by the following equations. For all products $i \in \mathbb{I}$,

$$\dot{Z}_i(t) = \dot{U}_i(t) - \mu_i \dot{T}_i(t), \quad Z(0) = z, \quad 0 \leqslant \dot{U}(t) \leqslant \lambda, \quad \dot{T}(t) \geqslant 0 \quad \text{and} \quad \sum_i \dot{T}_i(t) \leqslant 1. \tag{4}$$

In the fluid model, the workload vector is given by $W(0) = w$ and $W_i(t) = m_i Z_i(t)$. This simple and tractable approximation describes the transient behavior of the stochastic system starting from large initial conditions. While the leadtime constraints cannot always be guaranteed in the stochastic system, this is achievable in the fluid model. In fact, we will restrict the fluid-model analysis to the case with no admission control so that $U(t) = \lambda t$. This is a natural starting point in order to construct good sequencing and admission control policies that will hopefully perform well in terms of the leadtime constraints in the stochastic system with minimal blocking. The key observation in condition (2) takes the form

$$d_i \leqslant D_i \Longleftrightarrow \forall t \geqslant D_i : Z_i(t) \leqslant \lambda_i D_i. \tag{5}$$

An equivalent condition in terms of the workload process W is

$$d_i \leqslant D_i \Longleftrightarrow \forall t \geqslant D_i : W_i(t) \leqslant \rho_i D_i. \tag{6}$$

Thus, to satisfy the leadtime specifications in the fluid model, one must choose the control $T(\cdot)$ so as to keep the queue length vector Z in the box $R_S(\lambda) \triangleq \{z : z_i \leqslant \lambda_i D_i\}$, or equivalently, to keep $W(t)$ in $R_S(\rho)$. Conditions (5) or (6) ensure that at any time the fluid content in queue i comprises of fluid that arrived within the past D_i time units, thereby satisfying the leadtime constraints at any time. In terms of the sequencing control, which is described by the allocation process $T(t)$, (5) or (6) imply that the server must exert sufficient processing capacity to steer and keep the state in the box R_S. From (3) and the fluid constraint $Z_i(t) = z_i + \lambda_i t - \mu_i T_i(t) \leqslant \lambda_i D_i$, for $t \geqslant D_i$, we derive the constraints

$$T_i(t) \geqslant m_i z_i + \rho_i(t - D_i), \quad t \geqslant D_i, \ i \in \mathbb{I}, \tag{7}$$

which are often referred to in the communications literature as the *service curves*. Superimposing the capacity constraint $T(t) \leqslant t$ onto the service curves characterizes the allowable region for the allocation control to satisfy the delay specifications, as shown in Fig. 2. We summarize these results in the following proposition.

Proposition 1. *Consider the fluid model associated with a multi-product single server system with no admission control* ($U(t) = \lambda t$). *The following conditions are equivalent*, $\forall i \in \mathbb{I}$:

(i) $d_i \leqslant D_i$;
(ii) $Z_i(t) \leqslant \lambda_i D_i$, *for* $t \geqslant D_i$, (*or* $Z(t) \in R_S(\lambda)$);

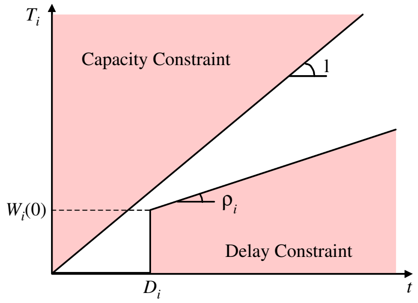

Fig. 2. Service curves: translation of $d_i \leqslant D_i$ into a constraint on the allocation process T.

(iii) $W_i(t) \leqslant \rho_i D_i$, *for* $t \geqslant D_i$, (*or* $W(t) \in R_S(\rho)$);
(iv) $T_i(t) \geqslant m_i z_i + \rho_i(t - D_i)$, *for* $t \geqslant D_i$ (*"service curves"*).

Systems with leadtime specifications typically suffer from state space explosion because, in addition to Z and W, a complete state descriptor includes information on the age of the jobs (or fluid) in the system. Expressions (5) and (7) have the desirable property that they do not depend on age information and are simple to analyze.

From condition (7) we see that it is necessary that $\rho = \sum_i \rho_i \leqslant 1$. The only other control constraint that can be extracted from (5) or (7) is

$$\forall t \geqslant D_i \text{ such that } Z_i(t) = \lambda_i D_i, \quad \dot{T}_i(t) \geqslant \rho_i, \tag{8}$$

which specifies a minimal processing rate to keep $Z_i(t) \in R_S(\lambda)$. There exist many policies that satisfy (8) and thus guarantee the leadtime specification $d_i \leqslant D_i$ for all products i. This allows for significant control flexibility that can be exploited in order to optimize other performance criteria or to satisfy auxiliary constraints. There are, however, some notable exceptions. For example, it is easy to verify that both FIFO and the $c\mu$ rule fail to guarantee the fluid-model delay specifications because in both cases the policy does not take corrective action when the system is about to violate one of these constraints. Specifically, FIFO fails by not allocating sufficient processing capacity to the jobs with the tightest deadline, and $c\mu$ fails by being myopic and disregarding leadtime considerations. More surprisingly, another commonly used policy referred to by *Shortest Delay First* also fails. This policy assigns static priorities to products in the reverse order of their leadtimes D_i. That is, smaller D_i implies higher priority, and with our labeling convention this means product 1 gets first priority, then product 2, and so forth. This is verified by considering the case $D_1 < D_2 < D_1\rho_1/(1 - \rho_1)$ and $z_i = \lambda_i D_i$, where the server will start processing type 2 jobs after time $t = D_1\rho_1/(1 - \rho_1)$, which is too late.

Shortest Delay First is but another example to illustrate that myopic static priorities cannot satisfy leadtime constraints. Hence, one must consider either dynamic priority rules that give priority to job types that are closest to violating their leadtime constraints, or processor sharing rules that guarantee a minimum level of service to each product. We conclude by reviewing two specific sequencing policies, one is a dynamic priority rule while the other is a processor sharing rule, that will prove useful later on.

Generalized Processor Sharing with parameter vector $\phi \geqslant 0$, where $\sum_i \phi_i = 1$, is denoted by GPS(ϕ) and defined as

$$\forall i : \text{if } Z_i(t) > 0, \text{ then } \dot{T}_i(t) = \frac{\phi_i}{\sum_{j:Z_j(t)>0} \phi_j}, \quad \text{otherwise } \dot{T}_i(t) = 0. \tag{9}$$

By choosing $\phi_i \in [\rho_i, 1]$, GPS is the obvious extension of (8). Note, however, that even if $\phi_i < \rho_i$ for some types, the single server system remains stable. Exploiting this remark, we can allow the vector ϕ to vary over the entire simplex $\sum_i \phi_i = 1$ and $\phi \geqslant 0$. This added flexibility allows us to shift capacity to products that have more stringent probabilistic guarantees or more stochastic variability. GPS is a natural generalization of uniform processor sharing (see [20]) and its packet-based version is known under the name Weighted Fair Queueing [9] and PGPS [26].

Generalized Longest Queue with parameter vector $\theta \geqslant 0$ is denoted by GLQ(θ) and gives at any time t preemptive priority to product

$$i^\theta(t) = \text{argmax}_i \theta_i Z_i(t), \tag{10}$$

where if $i^\theta(t)$ is not a singleton, we set $i^\theta(t) = \min\{j : j \in \text{argmax}_j \theta_j Z_j(t)\}$. By choosing

$$\theta_i^* = 1/\lambda_i D_i, \tag{11}$$

GLQ is a macroscopic implementation of a policy often referred to as *Earliest-Due-Date-First*, which gives priority to the product that is closest to its due date. In our model we cannot observe the age of the jobs in order to infer the due-dates, but we can focus on the constraints in (5) and give priority to the product that is closest to violating its corresponding constraint. The rule proposed by Harrison [16] is exactly GLQ(θ^*), which was shown asymptotically optimal in the heavy-traffic limit for systems with hard delay constraints by Plambeck et al. [27] and Van Mieghem [33]. Using large deviation analysis, Bertsimas et al. [3] showed that as θ and ϕ vary over the 2-D simplex, GLQ(θ) dominates GPS(ϕ) with respect to their delay violation probabilities. Stolyar and Ramanan [30] show that a variant of GLQ that uses explicit age information [3] is actually asymptotically optimal for an appropriate large deviations criterion.

3. Admission control in the single server system

Consider a single server system with only one product. In this system, an admission control policy simplifies to a threshold rule: admit jobs if queue length $Z(t)$ is less than a threshold, and reject otherwise. What should that threshold be to guarantee that $d \leqslant D$ with very high probability? The intuitive answer is to admit if $Z(t) \leqslant \mu D \Longleftrightarrow Z(t) \in R_S(\mu)$ because such a queue can always be cleared within the leadtime D by processing at full capacity. The analysis of Section 2, however, shows that for $t > D$ the queue length should lie in the region $R_S(\lambda)$ or $Z(t) \leqslant U(t) - U(t-D) \approx \lambda D$. How do we reconcile these two regions? Their difference stems from considering "transient" versus "steady-state" behavior. Indeed, the system cannot operate continuously with Z outside $R_S(\lambda)$ while satisfying the leadtime constraint $d \leqslant D$, but it could start there (or get there momentarily) and then quickly recover back into $R_S(\lambda)$. Which region should we use for admission control: $R_S(\mu)$ or $R_S(\lambda)$? (In workload space this becomes $R_S(\rho)$ or $R_S(e)$, where e is the vector of ones.) In a simple single-class stochastic system, the best one can do is choose a threshold K such that the delay violation $P(d > D)$ is less than the specification ϵ_d. (The threshold K can be found analytically for an $M/M/1/K$ system, where the blocking-delay trade-off is known.)

For a multi-product system, however, the problem becomes much more interesting because now the two controls, admission and sequencing, interact. (In a single product system, admission control and sequencing are de-coupled because sequencing is irrelevant and reduces to: process whenever $Z > 0$.) Assume homogeneous types with equal service times $m = 1/\mu$ and equal leadtime bounds D. Should we admit product i arrivals whenever $Z_i(t) \leqslant \mu D$? What if the μ's and D's differ among products? In this section we suggest specific admission control regions that exploit heterogeneity. These regions are the generalization of the transient single-product μD threshold rule and are larger than the steady-state box $R_S(\lambda)$, because sequencing will be used to steer Z into the steady-state region $R_S(\lambda)$. We will show how admission region depend on the sequencing rule that is used, and how to derive such "tailored" admission regions using fluid analysis and the leadtime constraint formulation (5).

3.1. The largest transient admission region R_T and GSD sequencing

Return to the fluid model associated with the multi-product single server system and ask the following question: "assuming no admission control (that is, $U(t) = \lambda t$), what is the set of initial conditions z for which the system can guarantee the leadtime constraints $d_i \leqslant D_i$?" We denote this set by R_T. This is essentially the largest admission region possible for the stochastic system under investigation: admissions outside R_T will result to leadtime constraint violation in the fluid model, which implies that similar leadtime

[3] In heavy traffic, the age of the oldest type i job has the same distribution as its delay and as Z_i/λ_i so that GLQ(θ) is equivalent to largest weighted delay, yet those policies differ in a large deviations sense (see [30,33]).

violations are very likely for the stochastic system. (This statement is "very likely" as opposed to "certain" because stochastic realizations could be particularly favorable in some cases.)

An alternative approach would be to impose a performance criterion that rewards the system when a job is admitted and penalizes the system when a job violates its delay constraint, and optimize overall performance. This will not be pursued in this paper but it is an interesting direction for future work. In this context, the work by Lu [22, Section 4] will provide a good starting point; the problem studied there is in terms of minimizing a holding cost criterion subject to an upper bound constraint on the queue length vector.

The precise derivation of this transient region goes as follows. Assume that the system is empty at time $t < 0$ and an initial (bulk) arrival of size z arrives to the system at time $t = 0$. (We will use both queue length and workload, denoted by w, depending on which one is more convenient; both are equivalent through the equation $w_i = m_i z_i$.) Recall the "service curve" constraints described earlier that provide a lower-bound on the cumulative allocation process T (as shown in Fig. 2) defined by

$$\underline{T}_i(t; w) = \begin{cases} 0, & t < D_i, \\ w_i + \rho_i(t - D_i), & t \geqslant D_i. \end{cases} \tag{12}$$

Any control $T_i(t)$ that guarantees that $d_i \leqslant D_i$ must satisfy the constraint $T_i(t) \geqslant \underline{T}_i(t; w)$. The jump at $t = D_i$ represents the requirement that by time D_i all initial workload w_i should have left the system, and the term $\rho_i(t - D_i)$ implies that all fluid that arrived in the system by the time $(t - D_i)$ will have left by time t, thus satisfying its leadtime specification. The fluid-model dynamics impose the obvious restriction that the allocation $T_i(t)$ can only increase at rate 1. This can be used to refine the lower bound $\underline{T}_i(t; w)$ to $\underline{T}_i(t; w) = (t - (D_i - w_i))^+$ for $t \leqslant D_i$. This refinement is implicit in the statement $T_i(t) \geqslant \underline{T}_i(t; w)$ that requires that $T_i(t)$ is a feasible allocation control for the fluid model. The transient region R_T is defined by

$$R_T \triangleq \left\{ w \geqslant 0 : \exists T(\cdot) \text{ such that } \forall i \in \mathbb{I}, T_i(t) \geqslant \underline{T}_i(t; w) \text{ and } T(t) = \sum_i T_i(t) \leqslant t \right\}. \tag{13}$$

Specifying R_T thus requires the specification of a sequencing policy that maximizes this admission region. We will show that the dynamic rule that we call *Generalized Shortest Delay First (GSD)* and that gives preemptive priority to product $i^\dagger(t)$ defined by

$$i^\dagger(t) = \begin{cases} i^{\theta^*}(t) & \text{if } Z(t) \in R_S, \\ \min\{i : Z_i(t) > \lambda_i D_i\} & \text{if } Z(t) \notin R_S, \end{cases} \tag{14}$$

where i^{θ^*} is the high priority product according to GLQ(θ^*) in (11), yields the maximal admission region R_T. The name GSD reflects the feature that outside $R_S(\lambda)$ the product with the nearest deadline is served. (Notice that the use of GLQ(θ^*) in the interior of $R_S(\lambda)$ is arbitrary; in fact, any other policy—including idling!—also yields R_T in the fluid model, since when the queue length vector is about to exit $R_S(\lambda)$ and start violating the delay constraints, the system switches to the GSD priorities that prevents that from happening.)

Proposition 2. *Suppose that $\rho = \sum_i \rho_i \leqslant 1$. The transient region R_T for the fluid model is*

$$R_T = \left\{ w \geqslant 0 : \sum_{j=1}^{i} w_j \leqslant D_i - \sum_{j=1}^{i-1} \rho_j (D_i - D_j), \forall \text{ types } i \right\}, \tag{15}$$

and starting from any $w \in R_T$, GSD sequencing guarantees that $d_i \leqslant D_i$.

All proofs in the paper are relegated to the Appendix A.

Example. ($I = 2$): Assume that there are only two products. The transient region R_T is the two-dimensional polyhedron defined by $w \geqslant 0$ and

$$w_1 \leqslant D_1,$$

$$w_1 + w_2 \leqslant D_2 - \rho_1(D_2 - D_1) = \rho_1 D_1 + (1-\rho_1)D_2 = W_{\max}.$$

$W_{\max}$ represents the maximal initial workload that the system can handle in D_2 time units (recall that $D_1 < D_2$). Our work suggests that admission control in terms of the total workload in the system (that is typically used), is not sufficient in the presence of leadtime guarantees. Specifically, one needs to add the more stringent constraint $w_1 \leqslant D_1$ that accounts for the difference between the two leadtimes for products 1 and 2; product 1 jobs admitted when $w_1 + w_2 \leqslant W_{\max}$ but $w_1 > D_1$ will violate their leadtime bound. As expected, we have that $R_S(\rho) \subset R_T$; see Fig. 3.

3.2. Admission control regions tailored to different sequencing rules

The proof of Proposition 2 shows that the admission region depends on the sequencing rule employed. Next, we show how to derive the admission region for a given sequencing policy. The approach is simple: we use the fluid-model equations under that sequencing policy and assuming that the system does not exercise admission control, and find the set of all initial conditions from which the system can clear its backlog and satisfy the leadtime specifications. The latter reduces to a simple check of whether starting from some initial state the system will satisfy the conditions in (5) or (6). To illustrate, we discuss the admission regions for a two type system under GPS(ϕ) or GLQ(θ).

3.2.1. Admission region under GLQ(θ)

Consider a two-product single server system with GLQ(θ) sequencing. Denote by $T^{\mathrm{GLQ}(\theta)}(\cdot; w)$ the allocation process under that sequencing policy starting from initial workload vector w. The corresponding admission region, denoted by $R_{\mathrm{GLQ}(\theta)}$, is defined by

$$R_{\mathrm{GLQ}(\theta)} \triangleq \left\{ w \geqslant 0 : \forall i \in \mathbb{I}, T_i^{\mathrm{GLQ}(\theta)}(\cdot; w) \geqslant \underline{T}_i(t; w) \text{ for } t \geqslant D_i \right\}.$$

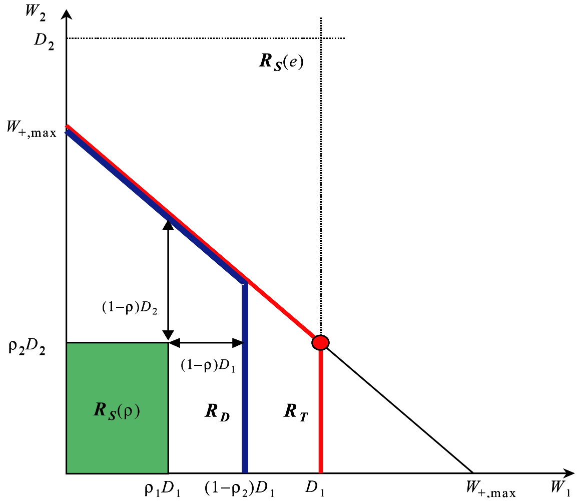

Fig. 3. The three admission control regions that guarantee leadtime constraints static $R_S(\rho)$, transient R_T, smallest dynamic R_D versus $R_S(e)$, which cannot guarantee leadtime constraints.

Proposition 3. *Consider a single server system with two products under GLQ(θ) sequencing and define $\alpha = \mu_1\theta_1/\mu_2\theta_2$. The GLQ admission region is*

$$R_{\mathrm{GLQ}(\theta)} = \{w \geqslant 0 : w_1 \leqslant D_1, w_1 + w_2 \leqslant W_{\mathrm{GLQ}(\theta)}\},$$

where $W_{\mathrm{GLQ}(\theta)} = \min((1 + \rho_1\alpha - \rho_2)D_1, (1 - \rho_1 + \alpha^{-1}\rho_2)D_2)$.

Corollary 4. *Let $\alpha^* = \mu_1\theta_1^*/\mu_2\theta_2^* = \rho_2D_2/\rho_1D_1$. $\forall\rho < 1 : R_{\mathrm{GLQ}(\theta)} \subset R_T$; only if $\rho = 1$ and $\alpha = \alpha^*$ is $R_{\mathrm{GLQ}(\theta)} = R_T$. In addition, $R_{\mathrm{GLQ}(\theta)}$ is maximized for some $\hat{\theta}$ such that $\hat{\alpha} = m_2\hat{\theta}_1/m_1\hat{\theta}_2 > \alpha^*$, and $\hat{\alpha} \to \alpha^*$ as $\rho \to 1$.*

(The corollary is proved as part of Proposition 3.) As shown in Fig. 4, $R_{\mathrm{GLQ}(\theta)}$ and R_T differ only by the maximal workload that they can handle. In moderate traffic ($\rho < 1$), the admission region $R_{\mathrm{GLQ}(\theta)}$ is strictly smaller than R_T for any choice of θ, while in heavy traffic ($\rho = 1$), the two regions agree when $\theta = \theta^*$. The $R_{\mathrm{GLQ}(\theta)}$-maximizing parameter $\hat{\theta}$ provides a useful selection rule for the parameter θ in the GLQ policy specification. In addition, the parameter θ^* derived here using simple fluid analysis agrees with the parameter that is asymptotically optimal in the heavy-traffic regime ([27,33])! Finally, the explanation of why $R_{\mathrm{GLQ}(\theta)}$ is smaller than R_T follows from analyzing the fluid trajectories. For example, under GLQ(θ^*) and starting from an initial condition with both products outside $R_S(\lambda)$, the sequencing rule will first equalize Z_i/λ_iD_i and then bring both queues simultaneously to the boundaries of the box. This disregards the fact that their leadtimes can be different, and it is clear that by careful selection of the initial workload this policy can fail to meet product 1's leadtime constraint. GSD, on the other hand, recognizes that $D_1 < D_2$ and gives priority to "lower" products outside $R_S(\lambda)$.

3.2.2. Admission region under GPS(ϕ)

Denote by $T^{\mathrm{GPS}(\phi)}(\cdot; w)$ the allocation process under GPS(ϕ) sequencing starting from initial workload vector w. The corresponding admission region, denoted by $R_{\mathrm{GPS}(\phi)}$, is defined by

$$R_{\mathrm{GPS}(\phi)} \triangleq \left\{w \geqslant 0 : \forall i \in \mathbb{I}, T_i^{\mathrm{GPS}(\theta)}(\cdot; w) \geqslant \underline{T}_i(t; w) \text{ for } t \geqslant D_i\right\}.$$

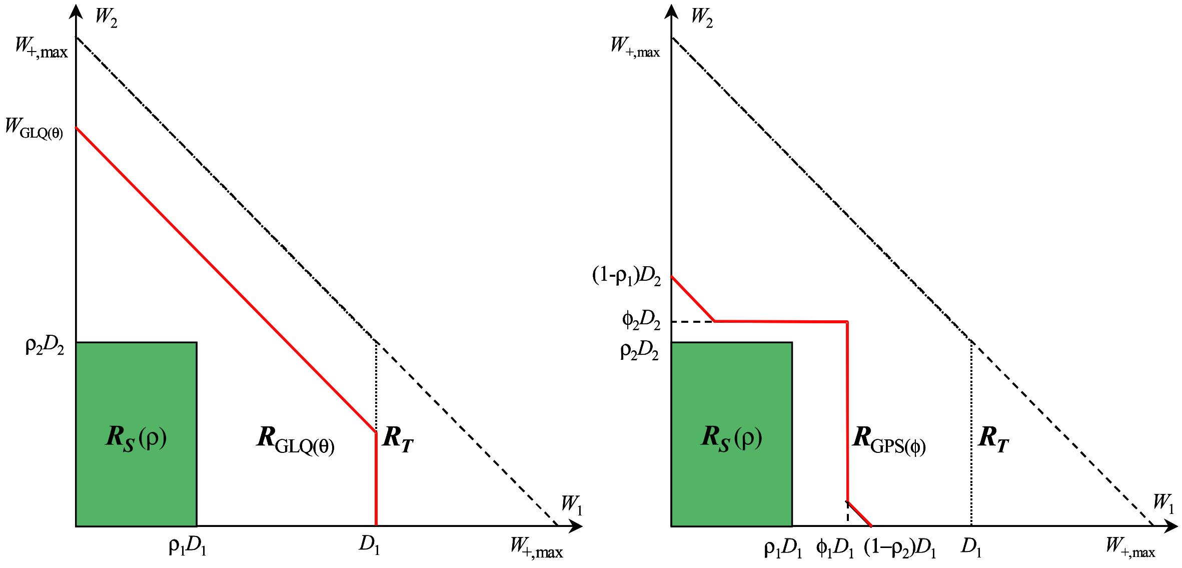

Fig. 4. Tailored admission control regions for GLQ(θ) and GPS(ϕ) for $\phi_i \geqslant \rho_i$, $i \in \mathbb{I}$.

Proposition 5. *Consider a single server system with two products under GPS(ϕ) sequencing and choose ϕ such that $\phi_i \geqslant \rho_i$ for $i = 1, 2$. The admission region is given by*

$$\begin{aligned} R_{\mathrm{GPS}(\phi)} = \{w \geqslant 0\} \cap \{\{\forall i : w_i \leqslant \phi_i D_i\} \cup \{w_1 > \phi_1 D_1, w_1 + w_2 \leqslant (1-\rho_2)D_1\} \\ \cup \{w_2 > \phi_2 D_2, w_1 + w_2 \leqslant (1-\rho_1)D_2\}\}. \end{aligned}$$

(The proof is relegated to the Appendix A.) We have restricted attention to the most natural regime where $\phi_i \geqslant \rho_i$ for $i \in \mathbb{I}$. Notice that this region is non-convex, as shown in Fig. 4. A related observation of non-convexity under the GPS policy has been made by Paschalidis in [25].

Corollary 6. $R_{\mathrm{GPS}(\phi)} \subset R_T$ *for any two-product single-server system with $\phi_i \geqslant \rho_i$ for $i \in \mathbb{I}$.*

It is important to note that in both cases the complexity of the derivation of the “tailored” admission regions increases as the number of product types grows, and can become impractical when I is large. In contrast, the specification of the transient region R_T that is associated with the GSD policy holds for an arbitrary number of types and thus does not suffer from this “curse of dimensionality”.

3.3. Mixed analysis: Fluid model with batch arrivals

Admission control based on the transient region R_T may be optimistic, since it assumes that all workload in the system has just arrived. In reality, jobs with less stringent leadtimes that are getting lower priority may have been in the system for some time already. This should affect the acceptance of higher priority jobs, since the server will have to process the former earlier than what the transient analysis had dictated. Therefore, we need to correct the admission region in order to capture the effect of aging jobs in the system. One approach would be to scale down the tailored transient admission region by a “safety factor” that is selected by simulation. Another approach applies the following “mixed” fluid-model analysis.

Consider a time $t^* \geqslant D_I$ when all initial transients have ended and the workload is $W(t^*)$. Assume that at t^* a batch arrival of workload size w arrives to the system. [4] We denote $W(t^*)$ by W, and the total workload immediately after the batch arrival by $W + w$. Given W, the admission region will be the set of vectors w that the system can admit such that it can still guarantee that $d_i \leqslant D_i$, for all types i. We refer to this region as the dynamic admission region, denoted by R_D. The basic premise is that R_D should provide more realistic admission and sequencing policies.

We will provide the derivation for the largest such region, and in passing also describe the corresponding sequencing rule. The extensions to GPS or GLQ are omitted. We start by constructing the service curve constraints $\underline{T}(t; W, w)$ as follows:

$$\underline{T}_i(t; W, w) = \begin{cases} 0, & t - t^* < (D_i - W_i/\rho_i)^+, \\ \rho_i(t - t^* - D_i + W_i/\rho_i)^+, & (D_i - W_i/\rho_i)^+ \leqslant t - t^* < D_i, \\ w_i + \rho_i(t - t^* - D_i + W_i/\rho_i)^+, & t - t^* \geqslant D_i. \end{cases} \tag{16}$$

With all initial transients cleared prior to t^* and $d_i \leqslant D_i$, we know that $W_i \leqslant \rho_i D_i$ for all i. The intuition behind these service curve constraints is that at time $(D_i - W_i/\rho_i)^+$ the type i “old” workload will become equal to $\rho_i D_i$, so that the server must start devoting ρ_i to processing this product in order to satisfy the leadtime constraints of this “old” fluid. The dynamic admission region is defined by

[4] This batch arrival may be motivated by large deviations theory, which predicts that large queue lengths build up by sudden bursts of closely spaced arrivals or long service times.

$$R_D(W) \triangleq \left\{ W + w \geqslant 0 : \exists T(\cdot) \text{ s.t. } \forall i \in \mathbb{I}, T_i(t) \geqslant \underline{T}_i(t; W, w) \text{ and } T(t) = \sum_i T_i(t) \leqslant t \right\}. \tag{17}$$

Proposition 7. *For any time $t^* > D_I$, denote by W the workload $W = W(t^*)$, and consider a batch arrival of size w. Then, the fluid model can accept any such arrivals provided that the total workload $W + w$ remains in the dynamic admission region*

$$R_D(W) = \left\{ W + w \geqslant 0 : \sum_{j=1}^{i} (W_j + w_j) + \sum_{j=i+1}^{I} \left(W_j - \rho_j (D_j - D_i) \right)^+ \leqslant D_i - \sum_{j=1}^{i-1} \rho_j (D_i - D_j), \ \forall i \in \mathbb{I} \right\}. \tag{18}$$

The presence of "older fluid" restricts the magnitude of the batch arrival w. So, in general, $R_D(W) \subseteq R_T$. If the system is empty at time t^*, then the region $R_D(0)$ recovers the transient region R_T (as it should). Actually, the same is true if there is only a small amount of "old" fluid present: $W_j < \rho_j(D_j - D_i)$ for all $j > i$, in which case higher priority products (with less stringent leadtimes; recall that $D_1 \leqslant D_2 \leqslant \cdots \leqslant D_I$) will not reach the level $\rho_j D_j$ prior to time D_i, where the delay constraints due to W_j will become binding.

Example. ($I = 2$ `contd.`): The dynamic admission region becomes:

$$W_1 + w_1 + (W_2 - \rho_2(D_2 - D_1))^+ \leqslant D_1,$$

$$(W_1 + w_1) + (W_2 + w_2) \leqslant D_2 - \rho_1(D_2 - D_1) = \rho_1 D_1 + (1 - \rho_1) D_2 = W_{\max}.$$

The smallest dynamic region corresponds to the case $W_i = \rho_i D_i$ where the first condition becomes

$$W_1 + w_1 \leqslant (1 - \rho_2) D_1.$$

Fig. 3 compares this smallest dynamic region R_D to the other two regions $R_S(\rho)$ and R_T. If $\rho < 1$, then $1 - \rho_2 > \rho_1$ and $R_D(\rho D)$ strictly dominates the region $R_S(\rho)$, whereas, if $\rho = 1$, then the first condition becomes $W_1 + w_1 \leqslant \rho_1 D_1$, which is the same as in $R_S(\rho)$.

In practice, from the observation of the queue length vector z, one cannot figure out what fraction of the total workload $W_i + w_i = m_i z_i$ has been in the system for some time already (we called that fraction W_i in the mixed fluid analysis above). A conservative choice would be to set $W_i = \min(m_i z_i, \rho_i D_i)$ and $w_i = (m_i z_i - \rho_i D_i)^+$, and to proceed using (16); this is the largest amount of "old" fluid that can be embodied in z_i and that can be attributed to work arriving at a constant rate ρ_i while satisfying the leadtime constraint $d_i \leqslant D_i$.

4. Simulation study of the control policies in the stochastic single-server system

In the previous sections we have provided a simple framework to design control policies for multi-product systems with leadtime constraints using fluid analysis. These results can serve as a starting point for policy construction for the stochastic system. This section provides some initial justification of this conjecture by conducting a series of simulation experiments. In particular, we illustrate that the naive admission policy based on the single-product reasoning that admits new product i arrivals whenever $Z_i \leqslant \mu_i D_i$ adds little to the overall system performance. On the other hand, the tailored admission control proposals of Section 3 not only seem to perform very well, but also are rather robust in that they make the system performance insensitive to the chosen sequencing rule. We simulated a two-product $M/M/1$ single server system operating at moderate traffic intensity ($\rho = 0.83$) with parameters $\lambda = (0.5, 0.5)$ and $\mu = (1.5, 1)$.

4.1. Performance without admission control

As a benchmark, we simulated the system without admission control using GPS(ϕ) and GLQ(θ) sequencing. To investigate the full performance range, the parameters ϕ and θ were varied over their entire domain thereby tracing the frontier of their achievable regions for a fixed delay bound vector D. That is, any specification vector ϵ_d above and to the right of the curves in Fig. 5 can be guaranteed. (Recall that with no admission control the blocking levels ϵ_b are zero.) For each parameter value, three simulations of 200,000 service completions are reported. The figure shows the frontiers for two leadtime vectors D. Clearly, as D becomes larger, the bound is less stringent and easier to satisfy so that the frontier for $D = (20, 20)$ lies below and to the left of the frontier for $D = (10, 20)$.

More interestingly, while GLQ dominates GPS as asymptotic heavy traffic and large deviations analysis predict, the relative improvement is not dramatic; this difference clearly dependents on the probabilistic assumptions on the arrival and service time processes. (In the tail as $D \to \infty$, however, the improvement is more pronounced, as shown by the analytic results of Bertsimas et al. [4].) The figure also shows two analytic bounds. For extreme parameter values of ϕ and θ, both GPS and GLQ become static priority policies, which yield analytic lower performance bounds. The analytic upper bound is the performance of a "de-coupled" $M/M/1$ queue: GPS(ϕ) processes any non-empty buffer i at a rate of *at least* ϕ_i; the actual processing rate will be higher when the buffers of some other classes are empty. The processing gain that a class obtains when other classes are empty is called the *multiplexing gain*, which complicates performance analysis. By ignoring the multiplexing gain, the queues can be de-coupled and their performance is a lower bound on the multiplexed queues. Thus, under GPS(ϕ) the product i flow receives equal or better service than under an $M/M/1$ system with FIFO service and service rate $\tilde{\mu}_i = \mu_i\phi_i$ in isolation. This directly yields the following bound:

Lemma 8. *Under GPS(ϕ), product i delay d_i is stochastically smaller than the delay d_i^{MM1} of an $M/M/1$ queue with arrival rate λ_i and service rate $\tilde{\mu}_i = \mu_i\phi_i$. In particular,*

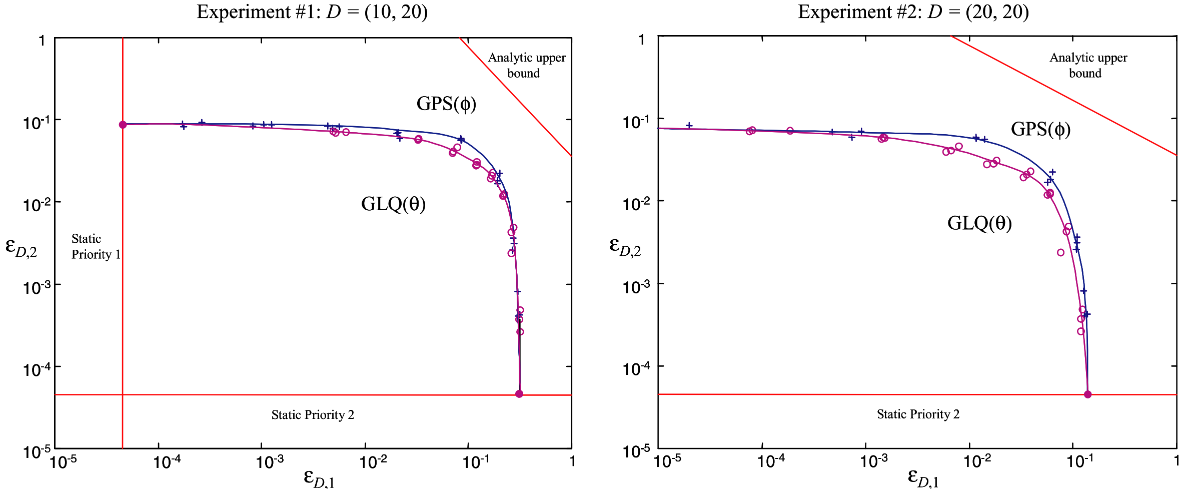

Fig. 5. Frontiers of the achievable regions under GPS and GLQ sequencing in a single server system without admission control with $\lambda = (0.5, 0.5)$ and $\mu = (1.5, 1)$.

$$\mathbf{P}(d_i > D_i) \leqslant \mathbf{P}(d_i^{MM1} > D_i) = \mathrm{e}^{-D_i(\mu_i\phi_i-\lambda_i)}. \tag{19}$$

Thus, in order for the specification $\mathbf{P}(d_i > D_i) \leqslant \epsilon_{d,i}$ to be achievable it suffices to set

$$\phi_i \geqslant \rho_i^{\mathrm{eb}}(D_i, \epsilon_{d,i}) \triangleq \rho_i\left(1 + \frac{\ln \epsilon_{d,i}^{-1}}{\lambda_i D_i}\right). \tag{20}$$

The quantity $\rho_i^{\mathrm{eb}} > \rho_i$ represents the *effective capacity* that should be allocated to the type i flow as prescribed by this single type approximation. Note that as either λ or D increases, $\rho_i^{\mathrm{eb}} \downarrow \rho_i$. The same results can be extended to general distributions and single class $G/G/1$ approximations. In this case, one does not have exact expressions as in (19) and (20), but has to rely on asymptotic analysis of the tail probabilities based on large deviations theory. In communication networks, the quantity ρ_i^{eb} is referred to as the *effective bandwidth* [5] [19] of an $M/M/1$ queue under the leadtime specifications $(D_i, \epsilon_{d,i})$. Clearly, by neglecting the multiplexing gain, our expression (20) is an upper estimate of the true "*effective bandwidth*" for the multiproduct system. An analysis that incorporates the multiplexing gain, such as the large deviations analysis of Bertsimas et al. [3] and Zhang [37] of a two-class single server under very general probabilistic assumptions, yields an effective bandwidth that is lower than the one given in (20). Nevertheless, the exceeding simplicity of (20) is appealing: it yields a simple linear constraint (in log–log scale) that is not completely ridiculous as shown in Fig. 5.

4.2. Performance with admission control

As a first exploration of the impact of admission control, we compared GPS and GLQ under four input control policies: (1) no admission control versus admission control using (2) the box $R_S(\mu)$ that admits product i arrivals if $Z_i \leqslant \mu_i D_i$), which also is used by Paschalidis [25, Section 7], (3) the largest region R_T and (4) the tailored regions $R_{\mathrm{GPS}(\phi^*)}$ or $R_{\mathrm{GLQ}(\theta^*)}$, respectively. We have simulated the same M/M/1 system as before, but now with GLQ(θ^*) and GPS(ϕ^*), where $\phi_i^* = \rho_i/\rho$. The results, with 95% confidence intervals, are reported in Table 1 for $D = (10, 20)$. We highlight a few observations:

1. Holding the sequencing policy constant, smaller admission regions result in smaller leadtime violation probabilities at the expense of increased blocking rates. Given that $R_{\mathrm{tailored}} \subset R_T \subset R_S(\mu)$, this leadtime-blocking trade-off is evident along horizontal rows in the Table.
2. The box admission region $R_S(\mu)$ provides little improvement over the case of no admission control. This is illustrated by comparing the results between the first and second columns.
3. The tailored admission region results in substantial performance improvements over both the largest region R_T, and the naive box admission region $R_S(\mu)$. For GLQ, blocking 0.49% (0.90%) more type 1 (2) jobs decreases the leadtime violation probabilities by 2.3% (2.4%) for admitted type 1 (2) jobs over the box admission region $R_S(\mu)$. Similarly for GPS, blocking 3.3% (0.6%) more type 1 (2) jobs decreases the leadtime violation probabilities by 7.2% (2.2%) for admitted type 1 (2) jobs over the box admission region $R_S(\mu)$.
4. GLQ(θ^*) tries to equalize the leadtime violation probabilities among product types by striving to equalize $Z_i/\lambda_i D_i$ over all i; this was also theoretically predicted by the asymptotic results in Van Mieghem [33]. In GPS, on the other hand, the choice $\phi = \phi^*$ does not correct for the difference between the leadtimes of

[5] The definition of effective bandwidth in the communications literature is somewhat different. It pertains to the asymptotic rate of decay of the tail queue-length probability, and thus relates to loss probabilities.

Table 1
Comparative simulated performance, including 95% confidence intervals, of GPS($\phi^{\star}$) and GLQ($\theta^{\star}$) under four admission control policies; $\lambda = (0.5, 0.5)$, $\mu = (1.5, 1)$ and $D = (10, 20)$

Sequencing	Admission			
	None	$R_S(\mu)$	R_T	R_{tailored}
GPS(ϕ^*)				
$\epsilon_{d,1}$	0.1269 ± 0.0139	0.1188 ± 0.0118	0.1041 ± 0.0079	0.0463 ± 0.0020
$\epsilon_{d,2}$	0.0397 ± 0.0055	0.0351 ± 0.0042	0.0183 ± 0.0034	0.0127 ± 0.0023
$\epsilon_{b,1}$	0	0.0027 ± 0.0008	0.0052 ± 0.0007	0.0359 ± 0.0015
$\epsilon_{b,2}$	0	0.0010 ± 0.0004	0.0061 ± 0.0012	0.0073 ± 0.0009
GLQ(θ^*)				
$\epsilon_{d,1}$	0.0542 ± 0.0068	0.0481 ± 0.0044	0.0316 ± 0.0029	0.0250 ± 0.0017
$\epsilon_{d,2}$	0.0517 ± 0.0076	0.0443 ± 0.0046	0.0260 ± 0.0025	0.0202 ± 0.0026
$\epsilon_{b,1}$	0	0.0000 ± 0.0000	0.0041 ± 0.0007	0.0049 ± 0.0005
$\epsilon_{b,2}$	0	0.0010 ± 0.0006	0.0068 ± 0.0009	0.0103 ± 0.0016

products 1 and 2 ($D_1 = 10$ while $D_2 = 20$), and leads to higher violation probabilities for product 1 that has the shorter leadtime. (Clearly, choosing an appropriate ϕ, for example $\phi_i \geqslant \rho_i^{\text{eb}}(D_i, \epsilon_{d,i})$, would improve this balance.)

The simulation prompts an interesting remark regarding blocking probabilities of an admission region R. Theoretically, these can be obtained from the steady-state queue-count Z distribution Π as follows. Product i's blocking probability b_i equals the sum of probabilities $\Pi(z_i)$ of the boundary points $z_i \in \{Z \in R \text{ and } Z + e_i \notin R\}$. For our two-product example using a "cut-off triangular" admission region like R_T and $R_{\text{GLQ}(\theta^*)}$ this means: b_1 is the probability of the sloped boundary $\{Z \geqslant 0 : m_1 Z_1 + m_2 Z_2 = W_{\text{GLQ}}, m_1 Z_1 \leqslant D_1\}$ and the vertical segment $\{Z \geqslant 0 : m_1 Z_1 + m_2 Z_2 < W_{\text{GLQ}}, m_1 Z_1 = D_1\}$. Similarly, b_2 is the probability of the sloped boundary only. Therefore, one expects $b_1 \geqslant b_2$, regardless of the sequencing policy that is used. (With a pure triangular admission region, one expects $b_1 = b_2$ for any sequencing policy.) While this holds for a continuous-state process, the simulation results indicate the reverse and the culprit lies in the discreteness of Z. [6]

Finally, to get an indication of the comparative performance of the joint sequencing-admission control policies presented in Section 3, we compared GPS(ϕ^*), GLQ(θ^*) and GSD, each with their tailored admission region. The results for our simulated system are reported in Table 2. Both GLQ and GSD seem to outperform GPS, each using their tailored admission region, however, the differences seem to be smaller than those reported in Fig. 5 and Table 1. The explanation is that all three pairs of control policies were designed in such a way so that (a) jobs admitted should never exceed their delay bounds, and (b) the system will never have to block. In this way, tailored admission control compensates for some of the performance differences that can be attributed to sequencing alone, and makes system performance robust to the specific choice of sequencing rule. GLQ and GSD provide roughly similar performance (we know that both policies

[6] For example, consider our simulated system with admission region $\frac{2}{3}Z_1 \leqslant 10$ and sloped line $\frac{2}{3}Z_1 + Z_2 = \zeta$, where $\zeta = W_{+,\max} = 16\frac{2}{3}$ for R_T and $\zeta = W_{\text{GLQ}} = 15$ for R_{GLQ}. Because of integer constraints, b_1 includes only roughly two-thirds of the points along the sloped line (to be precise: 10 for R_T and 11 for R_{GLQ} out of 16), while b_2 includes them all (16). Therefore, if the vertical segment $Z_1 = 15$ would have negligible probability, we would expect that b_1 to be roughly two-thirds of b_2. (This would be exact if the steady-state measure Π is uniform along the sloped line; in general, it will be different.) This agrees with the simulation results for GLQ, which indeed puts minimal mass on the vertical segment. GPS yields $b_1 \simeq \frac{5}{6}b_2$ under R_T, reflecting more probability mass on the vertical segment.

Table 2
Comparative simulated performance, including 95% confidence intervals, of three sequencing policies, each using their fluid-optimal tailored admission regions

D	Policy		
	1. GPS(ϕ^*) with $R_{\mathrm{GPS}(\phi^*)}$	2. GLQ(θ^*) with $R_{\mathrm{GLQ}(\theta^*)}$	3. GSD with R_T
(10, 20)			
$\epsilon_{d,1}$	0.0463 ± 0.0020	0.0250 ± 0.0017	0.0275 ± 0.0024
$\epsilon_{d,2}$	0.0127 ± 0.0023	0.0202 ± 0.0026	0.0275 ± 0.0037
$\epsilon_{b,1}$	0.0359 ± 0.0015	0.0049 ± 0.0005	0.0048 ± 0.0009
$\epsilon_{b,2}$	0.0073 ± 0.0009	0.0103 ± 0.0016	0.0059 ± 0.0011
(20, 20)			
$\epsilon_{d,1}$	0.0104 ± 0.0012	0.0129 ± 0.0018	0.0116 ± 0.0021
$\epsilon_{d,2}$	0.0157 ± 0.0017	0.0161 ± 0.0028	0.0169 ± 0.0028
$\epsilon_{b,1}$	0.0054 ± 0.0006	0.0017 ± 0.0004	0.0023 ± 0.0004
$\epsilon_{b,2}$	0.0079 ± 0.0009	0.0035 ± 0.0008	0.0029 ± 0.0006

are identical if $\rho \to 1$), while GSD appears to provide slightly better performance in the tails for large D. Again, a definite comparison should use more precise, analytical techniques.

5. Multi-class networks with leadtime constraints

This section extends the leadtime constraint formulation of Section 2 to open multi-class queueing networks, and analyzes the simpler fluid-control problem to gain some insights on admission and sequencing control in networks. (Again, we do not consider routing control.)

5.1. Network model

The network consists of S single server stations (or servers), indexed by $s = 1, \ldots, S$. As before, we have exogenous external arrivals of I products, indexed by $i \in \mathbb{I} = \{1, \ldots, I\}$, which enter the network according to a renewal process with rate λ_i for type i. Each product type follows a deterministic *route* or processing sequence through the network denoted by

$$r_i = [s(i,1), \ldots, s(i,k), \ldots, s(i,n_i)], \tag{21}$$

where n_i is the total number of processing steps for jobs of product i, (i,k) is the class designation at the kth processing step along this route, and $s(i,k)$ denotes the server responsible for class (i,k). Upon completion of this processing sequence, jobs exit the network. There are $\sum_i n_i$ classes in total. The network description is completed by extending all other assumptions of Section 2 to this setting. Mean processing times will be denoted by $m_{(i,k)}$ and the corresponding rates by $\mu_{(i,k)} = 1/m_{(i,k)}$. The nominal load at server $s(i,k)$ due to class (i,k) traffic is $\rho_{(i,k)} = \lambda_i/\mu_{(i,k)}$. The aggregate load due to all product i flows through server s will be denoted by ρ_i^s, where

$$\rho_i^s = \lambda_i \sum_{j:(i,j)\in s} m(i,j) = \sum_{j:(i,j)\in s} \rho_{(i,j)},$$

and the total traffic intensity at each station is given by $\rho^s = \sum_i \rho_i^s$. A two station example with two products is shown in Fig. 6. Both products follow identical routes, first visiting server 1 and then server 2 before exiting the system. Thus, there are four job classes $\{(1,1),(1,2),(2,1),(2,2)\}$ and two routes $r_1 = r_2 = [1,2]$.

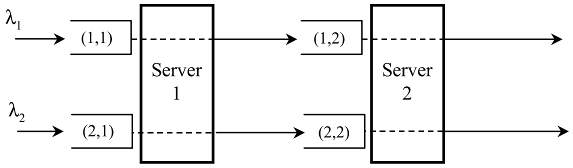

Fig. 6. A two-station two-type multi-class network.

Queue lengths will be denoted by $Z_{(i,k)}$. The (expected) workload or total processing requirement embodied in all class (i,k) jobs present in the system at time t is denoted by $W_{(i,k)}(t)$; same as in the fluid model. Similarly, $W_i^s(t)$ denotes the type i workload for server s:

$$W_{(i,k)}(t) = m_{(i,k)} \sum_{j \leqslant k} Z_{(i,j)}(t) \quad \text{and} \quad W_i^s(t) = \sum_{k:(i,k)\in s} W_{(i,k)}(t). \tag{22}$$

This network is a slight extension of the so called *re-entrant line* [21], that allows for many exogenous arrival streams, each following a re-entrant path through the network. Markovian switching between classes can be modelled in the usual way, where one correctly labels all possible routes through the system (accounting for probabilistic routing), and then proceeds in the framework described above; this enumeration of routes can be found, for example, in Kelly [18]. The caveat, of course, is that probabilistic switching in a feedback configuration leads to infinitely many routes.

5.2. Modeling and control for multi-class networks with leadtime constraints

Observation (2) extends directly to the network setting by summing all type i queue lengths:

$$\begin{aligned} d_i \leqslant D_i &\iff \text{all type } i \text{ jobs have arrived no longer than } D_i \text{ time units ago} \\ &\iff \forall t \geqslant D_i : \sum_k Z_{(i,k)}(t) \leqslant U_i(t) - U_i(t - D_i). \end{aligned} \tag{23}$$

Starting form initial condition $Z(0) = z$, the fluid model associated with this stochastic network is defined by: $\forall i \in \mathbb{I}$:

$$Z_{(i,1)}(t) = z_{(i,1)} + U_i(t) - \mu(i,1)T_{(i,1)}(t), \tag{24}$$

$$Z_{(i,k)}(t) = \mu_{(i,k-1)}T_{(i,k-1)}(t) - \mu_{(i,k)}T_{(i,k)}(t), \quad \forall k = 2,\ldots,n_i, \tag{25}$$

$$\sum_{(i,k)\in s} \dot{T}_{(i,k)}(t) \leqslant 1, \quad \dot{T}(t) \geqslant 0, \quad 0 \leqslant \dot{U}_i(t) \leqslant \lambda_i. \tag{26}$$

Without admission control, $U(t) = \lambda t$ and (23) simplifies to the obvious generalization of (5):

$$d_i \leqslant D_i \iff \forall t \geqslant D_i : \sum_k Z_{(i,k)}(t) \leqslant \lambda_i D_i, \tag{27}$$

Extending previous notation, we define $R_S(\lambda)$ as follows:

$$R_S(\lambda) \triangleq \left\{ Z : \sum_k Z_{(i,k)} \leqslant \lambda_i D_i, \forall i \in \mathbb{I} \right\}.$$

Let $Z_i(t) = \sum_k Z_{(i,k)}(t)$ be the total number of product i jobs in the network, $W_i(t) = \sum_s W_i^s(t) = \sum_k W_{(i,k)}(t)$ the corresponding total type i workload, and $\rho_i = \sum_s \rho_i^s$ the rate at which product i

workload is arriving to the system. The leadtime constraint for product i jobs can be expressed as $Z_i(t) \leqslant \lambda_i D_i$ for $t \geqslant D_i$, or in terms of workload in the form

$$d_i \leqslant D_i \iff \forall t \geqslant D_i : W_i(t) \leqslant \rho_i D_i.$$

That is, the total product i workload in the system at any point in time is less than or equal to the total amount of product i work that entered the system in the past D_i time units. We summarize these results in the following proposition.

Proposition 9. *Consider the fluid model associated with an open multi-class queueing network with no admission control* ($U(t) = \lambda t$). *The following conditions are equivalent*: $\forall i \in \mathbb{I}$:

(i) $d_i \leqslant D_i$;
(ii) $Z_i(t) = \sum_k Z_{(i,k)}(t) \leqslant \lambda_i D_i$ (*i.e.*, $Z_i(t) \in R_S(\lambda)$), *for* $t \geqslant D_i$;
(iii) $W_i(t) = \sum_s W_i^s(t) = \sum_k W_{(i,k)}(t) \leqslant \rho_i D_i$, *for* $t \geqslant D_i$.

5.3. Admission control in networks with leadtime constraints

This section will show that there is no simple generalization of the single-server expressions for the largest admission region R_T to multi-class networks. We shows that a complete characterization of R_T for a network is more complicated via a counter example and discuss some of its consequences. As before, we start with the service curve constraints. Let w denote the initial class level workload vector. For each class (i,k) we define

$$\underline{T}_{(i,k)}(t;w) = \begin{cases} 0, & t < D_i, \\ w_{(i,k)} + \rho_{(i,k)}(t - D_i), & t \geqslant D_i. \end{cases} \tag{28}$$

The leadtime specification $d_i \leqslant D_i$ is satisfied if and only if $T_{(i,k)}(t) \geqslant \underline{T}_{(i,k)}(t;w)$ for $k = 1, \ldots, n_i$. Recall that products are ordered such that $D_1 \leqslant D_2 \leqslant \cdots \leqslant D_I$. Summing over all classes (i,k) served at a station s and checking its capacity constraints yields

$$R_T \subseteq \left\{ w \geqslant 0 : \sum_{j=1}^{i} w_i^s \leqslant D_i - \sum_{j=1}^{i-1} \rho_i^s (D_i - D_j), \forall i, \forall s \right\}. \tag{29}$$

While this characterization was shown necessary and sufficient for a single server system, these conditions are no longer sufficient in the network setting as the following example will show.

Consider the network of Fig. 6 with initial condition $z = [1,0,1,0]$ and the following set of parameters: $\lambda_1 = 1/2$, $D_1 = 1$, $m_{(1,1)} = 1$, $m_{(1,2)} = 1/2$, and $\lambda_2 = 1/4$, $D_2 = 2$, $m_{(2,1)} = 1/2$, $m_{(2,2)} = 5/4$. The initial workload vector is given by

$$w_1^1 = m_{(1,1)}, \quad w_1^2 = m_{(1,2)}, \quad w_2^1 = m_{(2,1)}, \quad w_2^2 = m_{(2,2)},$$

and the corresponding capacity constraints are

$$w_1^1 \leqslant D_1, \quad w_1^2 \leqslant D_1, \quad w_1^1 + w_2^1 \leqslant D_1 - \rho_1^1(D_2 - D_1), \quad w_1^2 + w_2^2 \leqslant D_2 - \rho_1^2(D_2 - D_1).$$

It is easy to check that the capacity constraints are all satisfied. Also, given that $w_1^1 = 1$ and $m_{(1,1)} > m_{(1,2)}$, server 1 must process exclusively type 1 jobs until $t = D_1$ to satisfy the delay constraint for type 1 jobs. This implies that server 2 will be idling half of its capacity in the time interval $[0, D_1]$ instead of allocating it to class (2, 2) jobs whose queue is empty. Thus, server 2 will no longer be able to process all type 2 jobs present in the system at time $t = 0$ in the interval $[D_1, D_2]$ so that some type 2 jobs will violate their delay constraint.

The problem is that the capacity conditions for the network do not incorporate the constraint that $Z(t) \geqslant 0$. This non-negativity constraint was implicitly satisfied in the single server, where positive workload translates to positive queue length. In the network case, however, positive workload for class (i,k) or server s could be the result of jobs buffered upstream, and it does not necessarily imply positive buffer content. The non-negativity condition $Z(t) \geqslant 0$ can be added using the fluid-model equations:

$$Z_{(i,k)}(t) \geqslant 0 \Rightarrow T_{(i,k)}(t) \leqslant \begin{cases} z_{(i,k)}m_{(i,k)} + \frac{\lambda_i}{\mu_{(i,k)}}t, & k=1, \\ z_{(i,k)}m_{(i,k)} + \frac{\mu_{(i,k-1)}}{\mu_{(i,k)}}T_{(i,k-1)}(t), & 1<k\leqslant n_i. \end{cases} \tag{30}$$

This gives the complete characterization of the transient region R_T for the network:

$$R_T \triangleq \left\{ w : \forall (i,k) T_{(i,k)}(t) \text{ satisfies (30)}, \; T_{(i,k)}(t) \geqslant \underline{T}_{(i,k)}(t;w), \text{ and } \sum_{(i,k)\in s} T_{(i,k)}(t) \leqslant t, \forall t \geqslant 0 \right\}. \tag{31}$$

Thus, starting from an initial workload position w, the network can guarantee the leadtime constraints $d_i \leqslant D_i$ if and only if there exists an allocation policy $T(t)$ that satisfies this set of conditions. Unfortunately, (31) is not a finite dimensional characterization of the transient region. Instead, R_T is expressed in terms of a set of constraints that must be verified for all times $t \geqslant 0$. So far, we have been unable to reduce it to a simpler (polytopic) characterization like for the single server system. This is not surprising in light of the inherent complexity of networks and of their associated fluid models. See Dai and Weiss [8] for some results on fluid-model stability and Avram et al. [1], Chen and Yao [5], and Maglaras [24] for network control based on fluid-model analysis.

5.4. Sequencing in networks with leadtime constraints

We conclude by applying the GPS and GLQ sequencing rules to the network setting, and comparing their performance for the two station example of Fig. 6 through a simulation experiment. Recall that policies that satisfy (27) will guarantee the (fluid model) leadtime specifications $d_i \leqslant D_i$. Let $l_i(Z(t)) = \max\{k : Z_{(i,k)}(t) > 0\}$ be the furthest downstream non-empty class i buffer. Then, Proposition 9 and (27) imply the control constraint

$$\forall i \in \mathbb{I} : Z_i(t) = \lambda_i D_i \text{ for } t \geqslant D_i \Rightarrow \dot{Z}_i(t) \leqslant 0 \Rightarrow \dot{T}_{(i,k)}(t) \geqslant \rho_{(i,k)}, \quad \forall k \geqslant l_i(Z(t)). \tag{32}$$

Generalized Processor Sharing (GPS) in a network: The simplest and most conservative control that satisfies (32) is to disregard the distinction of upstream and downstream classes through $l_i(Z(t))$, and instead allocate capacity to every class (i,k) along the route r_i such that the flow along the route is equal to the input rate λ_i. This allocation keeps $Z_i(t)$ constant and the re-entrant path for product i flow now behaves like a "tandem line." In general given a vector $\phi \geqslant 0$ such that $\sum_{(i,k)\in s}\phi_{(i,k)} = 1$, the GPS($\phi$) policy is defined as follows: denote by $I_{B,s}(Z(t)) = \{(i,k) : Z_{(i,k)}(t) > 0\}$ the set of non-empty queues at server s at time t, and set

$$\forall (i,k) \in I_{B,s}(Z(t)) : \dot{T}_{(i,k)}(t) = \frac{\phi_{(i,k)}}{\sum_{(j,l)\in I_{B,s}(Z(t))}\phi_{(j,l)}} \quad \text{and} \quad \forall (i,k) \notin I_{B,s}(Z(t)) : \dot{T}_{(i,k)}(t) = 0.$$

The simplest choice for the parameters $\phi_{(i,k)}$ would be to set $\phi_{(i,k)} = \rho_{(i,k)} + \Delta\rho_{(i,k)}$ for some $\Delta\rho_{(i,k)} > 0$. Following the simple effective bandwidth calculation of Section 4, one could also study this network using a product-by-product analysis, where each re-entrant path is modelled as a tandem line operating in isolation. One should proceed with a large deviations analysis for a tandem line of $G/G/1$ queues, as in Bertsimas et al. [4]. The product-by-product analysis is a doable extension of [4], but the optimization step over the ϕ's seems hard. Finally, the analysis of GPS for the multi-class network (that would capture the "multiplexing gains" that occur when some classes are empty and their nominal capacity is redistributed to other classes at the same server) as in [3], is quite hard and presents an interesting and challenging open problem for further research.

Generalized Longest Queue (GLQ) in a network: We propose the following new extension of GLQ to the network setting: given an I-vector $\theta \geqslant 0$,

1. Rank products according to $\theta_i Z_i(t)$ and give highest priority to product $i^{\theta}(t)$ according to (10),
2. Within each route, give priority to classes closer to exit from the system; that is, $(i,k) \in s$ gets higher priority from all other classes $(i,l) \in s$, where $l < k$.

Once again, our fluid-model analysis suggests that one should focus on satisfying the deterministic leadtime constraints $Z_i(t) \leqslant \lambda_i D_i$, and set $\theta_i^* = 1/\lambda_i D_i$. Then, GLQ($\theta^*$) will try to minimize the "distance" from the boundary of the region $R_S(\lambda)$ at any given time. Indeed, the product that is closest to $\lambda_i D_i$ is also most likely to be closest to violating its leadtime constraint. The priority rule within each re-entrant path that corresponds to each type is known as *Last Buffer First Serve (LBFS)* policy. The *information requirements* of this network GLQ policy are attractive: each server s must know only the total number of jobs along each route that passes through that server. It is also interesting and reassuring that this network GLQ is stable for multi-class networks. (The proof shows that $V(z) = \max_i z_i/\lambda_i D_i$ serves as a Lyapunov function for the associated fluid model and then appeals to Dai's stability theorem to conclude that the underlying stochastic network is also stable; see [8]. Intuitively, the GLQ rule selects the route the gets high priority which corresponds to the index i that maximizes $z_i/\lambda_i D_i$. This route operates as a re-entrant line under the LBFS policy, which is known to be stable and for which $V_i(z_i) = z_i$ serves as a Lyapunov function [8, Theorem 4.4]. The details are omitted as this would require a substantial amount of new notation that is beyond the scope of this paper.)

We complete this section with a simulation experiment that compares these achievable frontiers under the network generalizations of GPS(ϕ) and GLQ(θ) for the two station example of Fig. 6 without admission control. The results are shown in Fig. 7. As expected, the leadtime violations and thus the network frontiers are higher than those of the isolated first-station, which were shown before in Fig. 5. As

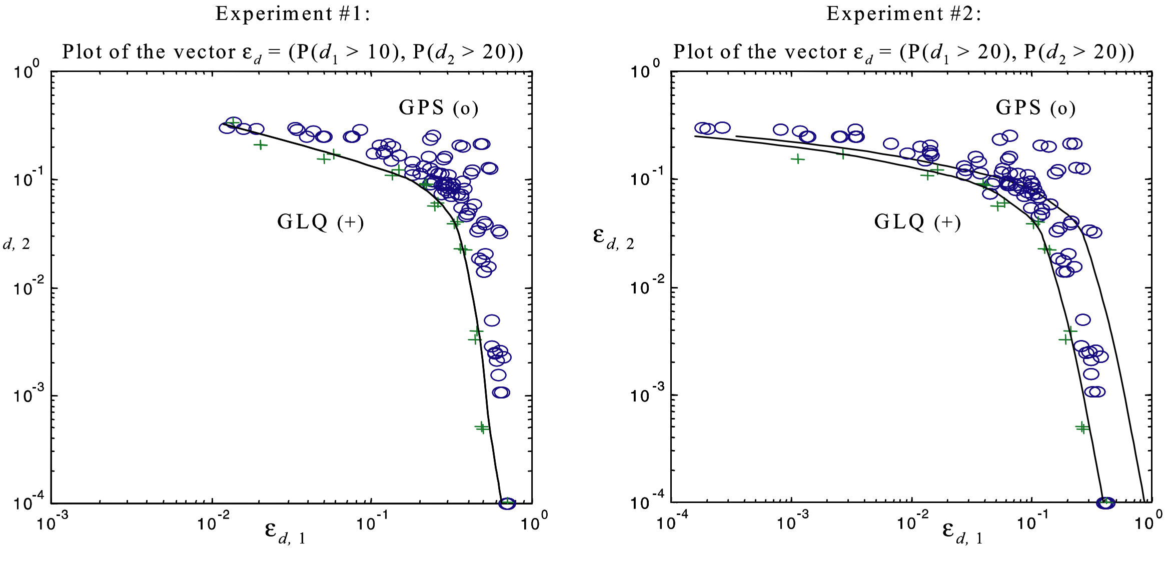

Fig. 7. Achievable regions under GPS versus GLQ sequencing for the network of Fig. 6 with $\lambda = [0.5, 0.5], \mu_1 = [1.5, 1]$ and $\mu_2 = [1, 1.5]$.

with the single server, it is striking how well GPS performs relative to GLQ. Indeed, careful selection of the GPS parameters yields almost similar performance; the caveat, of course, is that the selection of the ϕ's for GPS requires more work than the selection θ for GLQ. [7] Finally, in a recent paper Stolyar [31] extended the single server results of [30] and showed that a simpler version of the policy proposed above is in fact asymptotically optimal in a large deviations sense in the network setting. His results imply that our proposal is also asymptotically optimal according to his large deviations criterion.

6. Concluding remarks

The main objective of this paper was to provide a tractable approach to deal with leadtime constraints through admission and sequencing control. Our main tool was to invoke a deterministic fluid-model analysis which allowed as to translate the leadtime specification into simple linear constraints on the variables that are directly controllable in multi-product networks. This general constraint formulation was derived from the key observation (2) and its fluid analog (5). We also illustrated the power of this proposal by designing and analyzing various admission and sequencing policies and deriving parameter selection results that agree with heavy-traffic results. The main benefit of this approach is that it is possible to construct true multi-product admission policies for leadtime control. We showed how admission regions must be tailored to the sequencing rule employed, how such admission regions are derived, and how one chooses sequencing policies in conjunction with their associated admission controls by selecting appropriate parameters. Our simulation study gave first indication that these policies provide good performance when used in a stochastic setting. Finally, we made a first step in extending our approach to the network setting. We proposed network control policies that are simple, scalable and efficient. They do not require age information tracking and may provide a promising proposition in the context of large decentralized systems such as distributed supply chains.

Several interesting directions for future work exist for both theory development and applications. Our modeling framework can be extended to more general network structures and, specifically, to systems with dynamic routing control capability. It would be interesting to analyze, perhaps using large deviations, the effect of tailoring the admission control to a given sequencing policy. The starting point would be to analytically verify our preliminary simulation comparison of GPS and GLQ. This paper also may be useful in several applications. For example, to be able to guarantee tight service levels across a supply chain, our work suggests a minimal level of coordination in the sense that each entity (server, division, or company...) in the supply chain should have knowledge of the total inventory position of each product that it serves or produces. Several questions naturally arise: How does this proposal compare to traditional multi-echelon results? How does it impact supply chain performance? Which coordination and incentive mechanisms will achieve this performance?

Acknowledgements

We are grateful to Mike Harrison for providing the original idea to analyze leadtime performance via deterministic hard constraints.

[7] In the network, GPS has two degrees of freedom ($\phi_{(2,1)} = 1 - \phi_{(1,1)}$ and that $\phi_{(2,2)} = 1 - \phi_{(1,2)}$), while GLQ has only one (the relative weight θ between the two types). We simulated 49 parameter choices for GPS, which correspond to a 7×7 grid on the unit square. For GLQ we simulated 9 cases as θ varied from 0.1 to 10 together with the extreme cases of strict priority to one of the types, where GLQ and GPS coincide. For each policy-parameter combination we simulated two runs of 100,000 service completions.

Appendix A. Proofs

A.1. Proof of Proposition 2

We need to prove that (15) is necessary and sufficient.

Necessity: The capacity constraint $T(t) \leqslant t$ implies that $\sum_i \underline{T}_i(t; w) \leqslant t$, which in turn implies that at time $t = D_i$,

$$\sum_j \underline{T}_j(D_i; w) \leqslant D_i \Rightarrow \sum_{j=1}^{i} w_j + \sum_{j=1}^{i-1} \rho_j (D_i - D_j) \leqslant D_i,$$

which is (15). Note that by superimposing the service curves for each type i (shown in Fig. 2) we get a piecewise linear aggregate constraint on $T(t)$, and (15) simply requires that this aggregate constraint lies below the server's capacity. This can be verified by checking at the "corner" points at times D_i. Finally, we also need that $\sum_i \rho_i \leqslant 1$ to satisfy the long term leadtime constraints.

Sufficiency: We must show that our candidate GSD sequencing policy guarantees the leadtime bounds starting from any $w \in R_T$. The proof of sufficiency is by induction on the highest type i that satisfies its leadtime bound.

We start with some preliminary facts about the fluid-model behavior under our candidate policy. For the GLQ(θ^*) policy, let $\mathscr{I}^{\theta^*}(t)$ be the set of maximizers in (10). Let $c = \min_i 1/\lambda_i D_i$ and define $f(T) = \max_i Z_i(t)/\lambda_i D_i$. Then, it is easy to show that

$$\dot{F}(t) \leqslant \frac{c}{I} \left(\sum_{j \in \mathscr{I}^{\theta^*}(t)} \rho_j - 1 \right); \tag{A.1}$$

that is, $\dot{f}(t) < 0$, unless $\mathscr{I}^{\theta^*}(t) = \mathbb{I}$ and $\rho = 1$. Let $t_{R_S} = \inf\{t \geqslant 0 : W(t) \in R_S(\rho)\}$ when using the GSD policy. Then, from (A.1) it follows that $W(t) \in R_S(\rho)$ for all $t \geqslant t_{R_S}$; that is, once the workload process enters the box $R_S(\rho)$ it will stay there forever.

Whenever there are types outside the region $R_S(\rho)$ the GSD policy follows a static priority rule that is described by the following conditions: for all $j \in \mathbb{I}$,

$$\int_0^\infty (W_j(t) - \rho_j D_j)^+ \,\mathrm{d}\left(\sum_{l>j} T_l \right) = 0 \quad \text{and} \quad \int_0^t \mathbf{1}_{\{W_j(t) < \rho_j D_j\}} \,\mathrm{d}T_j = 0 \quad \text{for all } t < t_{R_S}; \tag{A.2}$$

that is, if $W(t) \notin R_S(\rho)$, the system never serves types that are in the strict interior of the region $R_S(\rho)$, and if type j is outside $R_S(\rho)$, then all types $l > j$ that have lower priority according to GSD cannot receive any service. This implies that for all $t < t_{R_S}$, $\dot{T}_j(t) = 0$ unless $W_j(t) \geqslant \rho_j D_j$. Moreover, for all $j < i^\dagger(t)$, for $t \in [D_j, t_{R_S}]$, $W_j(t) = \rho_j D_j$ and $\dot{T}_j(t) = \rho_j$; that is, high priority types receive just enough service to remain on the corresponding boundaries of $R_S(\rho)$; see, for example, [8, Section 4] for a discussion of fluid models under buffer priority rules. Finally, we note that GSD is a non-idling policy.

Type 1: Condition $T_1(t) \geqslant \underline{T}_1(t; w)$ reduces to $W_1(D_1) \leqslant \rho_1 D_1$. We argue by contradiction: Because type 1 gets preemptive priority whenever $W_1(t) > \rho_1 D_1$, condition $W_1(D_1) > \rho_1 D_1$, implies that $w_1 > D_1$ or else that $w \notin R_T$, which is false. Hence, $W_1(D_1) \leqslant \rho_1 D_1$. From (A.1) and (A.2), it follows that $W_1(t) \leqslant \rho_1 D_1$ for $t \geqslant D_1$, which implies that $d_1 \leqslant D_1$ and proves the induction hypothesis for type 1.

Type i: Assume that $d_j \leqslant D_j$ for all types $j < i$, and consider type i. We will analyze all possible scenarios for the workload of type i at time D_{i-1} and show that in all cases, $W_i(D_i)$ will have entered the region R_S and will stay there for all $t \geqslant D_i$, which implies that $d_i \leqslant D_i$.

(a) $D_{i-1} \geqslant t_{R_S}$: From (A.1), we have that $W(t) \in R_S$ for all $t \geqslant t_{R_S}$, and the induction hypothesis follows.

(b) $D_{i-1} < t_{R_S}$: The induction hypothesis and (A.2) imply that $W_j(t) = \rho_j D_j$ for $t \in [D_j, t_{R_S}]$ and all types $j < i$. We divide case (b) in two scenarios depending on $W_i(D_{i-1})$.

(b1) $W_i(D_{i-1}) \leqslant \rho_i D_i$: In this case $W_i \in R_S(\rho)$ and according to GSD $i^{\dagger}(D_{i-1}) > i$. From (A.2), the system will never work on types that are strictly inside $R_S(\rho)$ prior to time t_{R_S}, and thus type i must have started in the strict interior of $R_S(\rho)$ and $T_i(D_{i-1}) = 0$. Let $t_i = D_{i-1} + (W_i(D_{i-1})/\rho_i - D_i)$ be the time that type i will reach the boundary of $R_S(\rho)$ if it does not receive any service. Clearly, $t_i \leqslant D_i$ because D_i is the upper bound that is achieved if $W_i(0) = 0$. If $t_i \geqslant t_{R_S}$, then we are back to case (a), and we are done. If $t_i < t_{R_S}$, then $W_i(t_i) = \rho_i D_i$ and $t_i \leqslant D_i$. From (A.2) it follows that $\dot{T}_i(t) = \rho_i$ and $W_i(t) = \rho_i D_i$ for all $t \in [t_i, t_{R_S}]$, and from (A.1) we have that $W_i(t) \leqslant \rho_i D_i$ for all $t \geqslant t_{R_S}$. Hence, $d_i \leqslant D_i$ and the induction hypothesis again follows.

(b2) $W_i(D_{i-1}) > \rho_i D_i$: In this case, $i^{\dagger}(D_{i-1}) = i$. If $D_i \geqslant t_{R_S}$, then we are back in case (a) and again we are done. So, assume that $D_i < t_{R_S}$. As in (b), $W_j(t) = \rho_j D_j$ for $t \in [D_j, t_{R_S}]$ and all types $j < i$. This implies that $T_j(D_i) = W_j + \rho_j(D_i - D_j)$ for all $j < i$. We also claim that $T_j(D_{i-1}) = 0$ for all lower priority types $j > i$. We show this by contradiction. If $T_j(D_{i-1}) > 0$ for some $j > i$, then it must be that at some time $t' < D_{i-1}$, we had that $W_j(t') > \rho_j D_j$ and $W_l(t') \leqslant \rho_l D_l$ for all types $l < j$. Then according to (A.2), $W_l(t) \leqslant \rho_l D_l$ for all $t \in [t', t_{R_S}]$ and all $l < j$. But this is contradiction since $W_i(D_{i-1}) > \rho_i D_i$, and we are done.

Suppose now that $W_i(D_i) > \rho_i D_i$. That implies that $T_i(D_i) < W_i$. From the non-idling property of GSD and for $D_i \leqslant t_{R_S}$ we have that

$$\sum_j T_j(D_i) = \sum_{j<i} (W_j + \rho_j(D_i - D_j)) + T_i(D_i) = D_i;$$

that is, the server has not idled until time D_i. Under the assumption that $T_i(D_i) < W_i$, this would imply that $\sum_{j \leqslant i} W_j + \sum_{j<i} \rho_j(D_i - D_j) > D_i$, which violates the conditions of R_T, and leads to a contradiction. Hence, $W_i(D_i) \leqslant \rho_i D_i$. As in (b1) we establish the induction hypothesis for type i. The proof of the proposition follows by induction on i.

Note that (A.2) together with the non-idling property of our policy whenever $W(t) \notin R_S(\rho)$, also imply that $W(t) \in R_S(\rho)$ for all $t \geqslant t_{R_S}$. Hence, even if we were to idle in the strict interior of $R_S(\rho)$, GSD would still satisfy all leadtime constraints starting from any initial condition in R_T. □

A.2. Proof of Proposition 3: GLQ(θ) fluid admission region

The equations of motion for GLQ(θ) are most easily expressed in workload space using the parameter $\alpha = \frac{m_2}{m_1}\theta$:

If $N_2 < \theta N_1 \iff W_2 < \alpha W_1$, then $\dot{W}_1 = -(1-\rho_1)$ and $\dot{W}_2 = \rho_2$. Denote slope $\beta_1 = \frac{\dot{W}_2}{\dot{W}_1} = -\frac{\rho_2}{1-\rho_1}$.
If $N_2 > \theta N_1 \iff W_2 > \alpha W_1$, then $\dot{W}_1 = \rho_1$ and $\dot{W}_2 = -(1-\rho_2)$. Call slope $\beta_2 = \frac{\dot{W}_2}{\dot{W}_1} = -\frac{1-\rho_2}{\rho_1}$.
If $N_2 > \theta N_1 \iff W_2 = \alpha W_1$, then W moves down on α line at rate:

$$\left\{ \begin{array}{l} \dot{W}_1 + \dot{W}_2 = -(1-\rho) \\ \dot{W}_2 = \alpha \dot{W}_1 \end{array} \right| \begin{array}{l} \dot{W}_1 = -\frac{1}{1+\alpha}(1-\rho), \\ \dot{W}_2 = -\frac{\alpha}{1+\alpha}(1-\rho). \end{array}$$

The admission region is the set of initial workload vector w that can be cleared within their leadtime. Invoking (6), we must verify that $\forall t \geqslant D_i$ the trajectory remains in the box R_S (so that $W_i(t) \leqslant \rho_i D_i$). This verification involves simple linear algebra that is case dependent (refer to Fig. 8). In the sequel $w_+ = w_1 + w_2$.

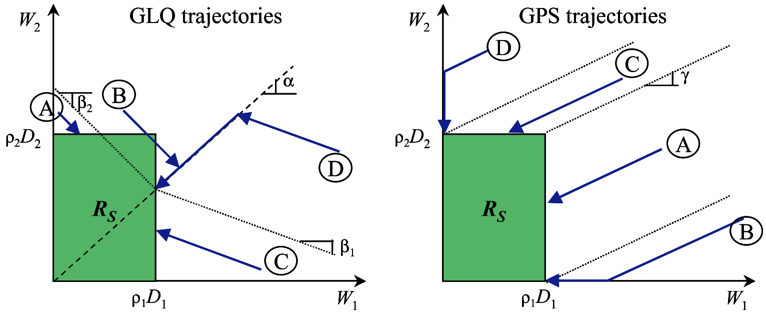

Fig. 8. Workload trajectories depend on the initial workload and the sequencing policy.

Case A: w above α line but below the β_2 line. Then W will move down with slope β_2 and hit upper box border first after time t: the trajectory keeps $W_1 < \rho_1 D_1$, but must get W_2 to $\rho_2 D_2$ within D_2, which requires

$$w_2 - (1-\rho_2)t = \rho_2 D_2 \iff t = \frac{w_2 - \rho_2 D_2}{1-\rho_2} < D_2 \iff w_2 < D_2.$$

Case B1: $\alpha < \alpha^* = \frac{\rho_2 D_2}{\rho_1 D_1}$ and w above α and β_2 lines. Now both types must reach the box. W will move down with slope β_2 until the α line, and then decrease along α line to box boundary. The condition that type 2 hit $\rho_2 D_2$ within D_2, is guaranteed by $w_2 < D_2$ and by type 1 getting to $\rho_1 D_1$ within $D_1 \leqslant D_2$. Thus: time t to get to α line is

$$\alpha(w_1 + \rho_1 t) = w_2 - (1-\rho_2)t \Rightarrow t = \frac{w_2 - \alpha w_1}{1 + \alpha\rho_1 - \rho_2}.$$

Then $W_1(t) = \frac{w_1(1-\rho_2)+\rho_1 w_2}{1+\alpha\rho_1-\rho_2}$ on the α line. Additional time s_1 to get to box boundary is

$$W_1(t) - \frac{1}{1+\alpha}(1-\rho)s_1 = \rho_1 D_1 \Rightarrow s_1 = \frac{w_1(1-\rho_2) + \rho_1 w_2 - (1+\rho_1\alpha - \rho_2)\rho_1 D_1}{(1+\alpha\rho_1-\rho_2)(1-\rho)}(1+\alpha).$$

The total trajectory must be completed by time D_1

$$t + s_1 \leqslant D_1 \iff w_+ \leqslant W_{\mathrm{GLQ},1} = (1 + \rho_1\alpha - \rho_2)D_1.$$

Case B2: $\alpha > \alpha^* = \frac{\rho_2 D_2}{\rho_1 D_1}$ and w above α line and above the β_2 line. W will move down with slope β_2 until the α line, and then decrease along α line to upper box boundary. Now type 1 may remain below $\rho_1 D_1$, or go temporarily beyond it, in which case that must happen within D_1. The entire trajectory must also be completed within time D_2. Time t to get to α line is as in case B1. Then we are at the point $W_2(t) = \alpha\frac{w_1(1-\rho_2)+\rho_1 w_2}{1+\alpha\rho_1-\rho_2}$ on the α line. Additional time s_2 to get to upper box boundary is

$$W_2(t) - \frac{\alpha}{1+\alpha}(1-\rho)s_2 = \rho_2 D_2 \Rightarrow s_2 = \frac{\alpha w_1(1-\rho_2) + \rho_1 w_2 \alpha - (1+\alpha\rho_1-\rho_2)\rho_2 D_2}{(1+\alpha\rho_1-\rho_2)\alpha(1-\rho)}(1+\alpha).$$

The total trajectory must be done in D_2 for type 2

$$t + s_2 \leqslant D_2 \iff w_+ \leqslant W_{\mathrm{GLQ},2} = \left(1 - \rho_1 + \alpha^{-1}\rho_2\right)D_2.$$

If $w_2 > \alpha\rho_1 D_1$, then the trajectory will go beyond $\rho_1 D_1$ and it must return within time D_1. This condition for type 1 is the same as in B1: $w_+ \leqslant W_{\mathrm{GLQ},1}$.

Case C: w below α and β_1 lines. W will move up with slope β_1, while keeping $W_2 < \rho_2 D_2$, and will hit the right boundary border first after time t. This must happen in time D_1

$$w_1 - (1-\rho_1)t = \rho_1 D_1 \iff t = \frac{w_1 - \rho_1 D_1}{1-\rho_1} < D_1 \iff w_1 < D_1.$$

Case D1: $\alpha < \alpha^* = \frac{\rho_2 D_2}{\rho_1 D_1}$ and w below α line but above the β_1 line. Now both types must reach the box. W will move up with slope β_1 until the α line, and then decrease along the α line to the box boundary. The entire trajectory must be completed within time D_1. This condition is the same as in B2: $w_+ \leqslant W_{\mathrm{GLQ},1}$. Given that type 2 will also move beyond $\rho_2 D_2$, that part of the trajectory must be done by D_2, which is implied by the total trajectory being completed in $D_1 \leqslant D_2$.

Case D2: $\alpha > \alpha^* = \frac{\rho_2 D_2}{\rho_1 D_1}$ and w below α line but above the β_1 line. Now both types must reach the box. W will move up with slope β_1 until the α line, and then decrease along the α line to the box boundary. The entire trajectory must be completed within time D_1, which again requires that $w_+ \leqslant W_{\mathrm{GLQ},1}$. Similarly, for type 2 to reach $\rho_2 D_2$ within D_2 that requires $w_+ \leqslant W_{\mathrm{GLQ},2}$.

Comparison of $W_{\mathrm{GLQ},1}$ and $W_{\mathrm{GLQ},2}$: Let $\alpha^* = \frac{\rho_2 D_2}{\rho_1 D_1}$ and

$$\Delta(\alpha,\rho) = W_{\mathrm{GLQ},2} - W_{\mathrm{GLQ},1} = \left(1 - \rho_1 + \alpha^{-1}\rho_2\right)D_2 - (1 + \rho_1\alpha - \rho_2)D_1,$$

which has a pole at $\alpha = 0$ and one positive zero

$$\hat{\alpha} = \frac{1}{2\rho_1 D_1}\left(D_2(1-\rho_1) + \sqrt{D_2^2(1-\rho_1)^2 + 4\rho_1 D_1 D_2 \rho_2}\right) \geqslant \frac{(1-\rho_1)D_2}{\rho_1 D_1} \geqslant \alpha^*.$$

Hence, Δ is clearly positive for $\alpha \leqslant \alpha^*$.

Comparison of $W_{\mathrm{GLQ},1}$ and $W_{+,\max} = \rho_1 D_1 + (1-\rho_1)D_2$:

$$\text{At } \alpha = \alpha^* : W_{+,\max} - W_{\mathrm{GLQ},1} = (1-\rho)(D_2 - D_1) \geqslant 0,$$

$$\text{At } \alpha \neq \alpha^* : W_{+,\max} - W_{\mathrm{GLQ},1} = \rho_1 D_1 + (1-\rho_1)D_2 - (1 + \rho_1\alpha - \rho_2)D_1,$$

which is clearly positive for $\alpha \leqslant \alpha^*$.

Comparison of $W_{\mathrm{GLQ},2}$ and $W_{+,\max} = \rho_1 D_1 + (1-\rho_1)D_2$:

$$\text{At } \alpha = \alpha^* : W_{+,\max} - W_{\mathrm{GLQ},2} = 0,$$

$$\text{At } \alpha \neq \alpha^* : W_{+,\max} - W_{\mathrm{GLQ},2} = \rho_1 D_1 + (1-\rho_1)D_2 - \left(1 - \rho_1 + \alpha^{-1}\rho_2\right)D_2,$$

which is clearly positive for $\alpha > \alpha^*$. Conclusion: $\forall \alpha \geqslant 0 : W_{\mathrm{GLQ}} = \min(W_{\mathrm{GLQ},1}, W_{\mathrm{GLQ},2}) \leqslant W_{+,\max}$. □

A.3. Proof of Proposition 5: GPS(ϕ) fluid admission region

The equations of motion for GPS(ϕ) in workload space are:

$$\dot{W}_i = \begin{cases} \rho_i - \phi_i & \text{if} \quad W_1, W_2 > 0, \\ \rho_i - (1 - \min(\phi_j, \rho_j)) & \text{if} \quad W_i > 0, \quad W_j = 0, \\ (\rho_i - \phi_i)^+ & \text{if} \quad W_i = 0, \quad W_j > 0. \end{cases}$$

(If $W_j = 0$, additional type j workload arriving at rate ρ_j receives processing at rate $\min(\phi_j, \rho_j)$.) Denote the slope $\gamma = \frac{\dot{W}_2}{\dot{W}_1} = \frac{\phi_2 - \rho_2}{\phi_1 - \rho_1}$ and consider the natural case $\phi_i \geqslant \rho_i$. Again consider the cases depicted in Fig. 8:

Case A: initial w between the line through upper right and lower right box corner with slope γ. When draining, W will hit the $W_1 = \rho_1 D_1$ boundary of box, which must happen in time $t \leqslant D_1$ (this also implies that W will hit $W_2 = \rho_2 D_2$ in less than $t \leqslant D_1 \leqslant D_2$)

$$w_1 - (\phi_1 - \rho_1)t = \rho_1 D_1 \Rightarrow t = \frac{w_1 - \rho_1 D_1}{\phi_1 - \rho_1} \leqslant D_1 \iff w_1 < \phi_1 D_1.$$

Case B: initial w below the line through lower right box corner with slope γ. When draining, W will hit the $W_2 = 0$ axis first, and then along that axis reach the $W_1 = \rho_1 D_1$ corner of box. First part will take a time t

$$w_2 - (\phi_2 - \rho_2)t = 0 \Rightarrow t = \frac{w_2}{\phi_2 - \rho_2},$$

at which time $W_1(t) = w_1 - \gamma^{-1}w_2 > \rho_1 D_1$. Getting to $\rho_1 D_1$ takes an additional time s_1

$$W_1(t) - (1-\rho)s_1 = \rho_1 D_1 \Rightarrow s_1 = \frac{w_1(\phi_2 - \rho_2) - w_2(\phi_1 - \rho_1) - \rho_1 D_1(\phi_2 - \rho_2)}{(\phi_2 - \rho_2)(1-\rho)}.$$

Total time to corner of box must be less than D_1 (which also implies that if $w_2 > \rho_2 D_2$, $W_2 = \rho_2 D_2$ is reached in less than $D_1 \leqslant D_2$)

$$t + s_1 \leqslant D_1 \Longleftrightarrow w_+ \leqslant D_1(1-\rho_2).$$

Case C: initial w between the line through upper right and upper left box corner with slope γ. When draining, W will hit the $W_2 = \rho_2 D_2$ boundary of box, and this must happen within D_2 time units (because $w_2 > \rho_2 D_2$). If $w_1 > \rho_1 D_1$, then it must first hit the $W_1 = \rho_1 D_1$ line in D_1 time units. This condition for type 1 is the same as for case A and requires $w_1 < \phi_1 D_1$. For type 2, hitting $W_2 = \rho_2 D_2$ must happen in time $t \leqslant D_2$

$$w_2 - (\phi_2 - \rho_2)t = \rho_2 D_2 \Rightarrow t = \frac{w_2 - \rho_2 D_2}{\phi_2 - \rho_2} \leqslant D_2 \Longleftrightarrow w_2 < \phi_2 D_2.$$

Case D: initial w above the line through upper left box corner with slope γ. When draining, W will hit the $W_1 = 0$ axis first, and then along that axis reach for the $W_2 = \rho_2 D_2$ corner of box. If $w_1 > \rho_1 D_1$, then it must first hit the $W_1 = \rho_1 D_1$ line in D_1 time units. This condition for type 1 is the same as for case A and requires $w_1 < \phi_1 D_1$. For type 2, the first part of the trajectory will take a time t

$$w_1 - (\phi_1 - \rho_1)t = 0 \Rightarrow t = \frac{w_1}{\phi_1 - \rho_1},$$

at which time $W_2(t) = w_2 - \gamma w_1 > \rho_2 D_2$. Getting to $\rho_2 D_2$ takes an additional time s_2

$$W_2(t) - (1-\rho)s_2 = \rho_2 D_2 \Rightarrow s_2 = \frac{w_2(\phi_1 - \rho_1) - w_1(\phi_2 - \rho_2) - \rho_2 D_2(\phi_1 - \rho_1)}{(\phi_1 - \rho_1)(1-\rho)}.$$

For type 2, total time to corner of box must be less than D_2

$$t + s_2 \leqslant D_2 \Longleftrightarrow w_+ \leqslant D_2(1-\rho_1).$$

A.4. Proof of Proposition 7

Summing over all types and checking the capacity constraint we establish necessity of these conditions. Sufficiency is proved by constructing a policy that satisfies all leadtime constraints given (16). This policy is a variation of GSD: it allocates a minimum level of effort to each type in order to clear old fluid out of the system while $d_i \leqslant D_i$, and gives all remaining capacity to class $i^\dagger(t)$ defined by the GSD policy. Let Z denote the queue length vector at time $t \geqslant t^*$.

Step 1: For all types i, if $Z_i \geqslant \lambda_i D_i$, $\dot{T}_i(t) = \rho_i$, else, $\dot{T}_i(t) = 0$; this takes care of the "old" fluid.
Step 2: Allocate unused capacity, $\delta = 1 - \sum_{i:Z_i \geqslant \lambda_i D_i} \rho_i$, to type $i^\dagger(t)$: $\dot{T}_{i^\dagger(t)}(t) = \dot{T}_{i^\dagger(t)}(t) + \delta$.

The proof is by induction on the type i and is very similar to that of Proposition 2. The only difference is that for type i we need to account for all processing time allocated to lower priority types $j > i$ to clear "old" type j fluid given by $\sum_{j>i}(W_j - \rho_j(D_j - D_i))^+$. The remaining capacity must go to type $i^\dagger(t)$ and the argument of Proposition 2 completes the proof.

References

[1] F. Avram, D. Bertsimas, M. Ricard, Fluid models of sequencing problems in open queueing networks: An optimal control approach, in: F. Kelly, R. Williams (Eds.), Stochastic Networks, Proceedings of the IMA, vol. 71, Springer-Verlag, New York, 1995, pp. 199–234.

[2] D. Bertsimas, I. Paschalidis, Probabilistic service level guarantees of make-to-stock manufacturing systems, 1999, preprint.

[3] D. Bertsimas, I.C. Paschalidis, J.N. Tsitsiklis, Asymptotic buffer overflow probabilities in multiclass multiplexers: An optimal control approach, IEEE Transactions on Automatic Control 43 (3) (1998) 315–335.

[4] D. Bertsimas, I.C. Paschalidis, J.N. Tsitsiklis, On the large deviations behavior of acyclic networks of G/G/1 queues, Annals of Applied Probability 8 (4) (1998) 1027–1069.

[5] H. Chen, D. Yao, Dynamic scheduling of a multiclass fluid network, Operations Research 41 (6) (1993) 1104–1115.

[6] R.L. Cruz, A calculus for network delay, part I: Network elements in isolation, IEEE Transactions on Information Theory 37 (1) (1991) 114–131.

[7] R.L. Cruz, A calculus for network delay, part II: Network analysis, IEEE Transactions on Information Theory 37 (1) (1991) 132–141.

[8] J.G. Dai, G. Weiss, Stability and instability of fluid models for certain re-entrant lines, Mathematics of Operations Research 21 (1996) 115–134.

[9] A. Demers, S. Keshav, S. Shenker, Analysis and simulation of a fair queueing algorithm, Internetworking—Research and Experience 1 (1990).

[10] I. Duenyas, Single facility due date setting with multiple customer classes, Management Science 41 (1995) 608–619.

[11] R.G. Gallager, B. Hajek, F.P. Kelly, D. Mitra, P.P. Varaiya (Eds.), Advances in the fundamentals in networking—Parts I–II, IEEE Journal on Selected Areas of Communications 13 (1995).

[12] P. Glasserman, Bounds and asymptotics for planning critical safety stocks, Operations Research 45 (1997) 244–257.

[13] P. Glasserman, Y. Wang, Fill-rate bottlenecks in production-inventory networks, Manufacturing & Service Operations Management 1 (1999) 62–76.

[14] S. Graves, A review of production scheduling, Operations Research 29 (1981) 646–675.

[15] J.M. Harrison, The BIGSTEP approach to flow management, in: F.P. Kelly, S. Zachary, I. Ziedins (Eds.), Stochastic Networks: Theory and Applications, Clarendon Press, Oxford, 1996, pp. 57–90.

[16] J.M. Harrison, Network control problems with constraints on throughput time, in: 10th INFOMRS Applied Probability Conference, Ulm, Germany, 1999.

[17] W.J. Hopp, M.L. Spearman, Factory Physics, Irwin-McGraw-Hill, 1996.

[18] F.P. Kelly, Reversibility and Stochastic Networks, John Wiley & Sons, New York, NY, 1979.

[19] F.P. Kelly, Notes on effective bandwidths, in: F. Kelly, S. Zachary, I. Ziedins (Eds.), Stochastic Networks: Theory and Applications, Oxford University Press, 1996, pp. 141–168.

[20] L. Kleinrock, Queuing Systems Vol. 2: Computer Applications, 1976.

[21] P.R. Kumar, Re-entrant lines, Queueing Systems 13 (1993) 87–110.

[22] Y. Lu, Dynamic scheduling with side constraints, Ph.D. Thesis, IEOR Dept., Columbia University, 1998.

[23] E.L. Lawler, J.K. Lenstra, A.H.G.R. Kan, D.B. Shmoys, Sequencing and scheduling: Algorithms and complexity, in: Logistics of Production and Inventory, in: S.C. GravesA.H.G.R. Kan, P.H. Zipkin (Eds.), Handbooks in OR & MS, vol. 4, North-Holland, 1993, pp. 445–522.

[24] C. Maglaras, Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality, Annals of Applied Probability 10 (3) (2000).

[25] I. Paschalidis, Class-specific quality of service guarantees in multimedia communication networks, Automatica 35 (12) (1999) 1951–1969.

[26] A.K. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: The single-node case, IEEE/ACM Transactions on Networking 1 (3) (1993) 344–357.

[27] E. Plambeck, S. Kumar, J.M. Harrison, A multiclass queue in heavy traffic with throughput time constraints: Asymptotically optimal dynamic controls, Queueing Systems 39 (2001) 23–54.

[28] M.I. Reiman, Open queueing networks in heavy traffic, Mathematics of Operations Research 9 (1984) 441–458.

[29] M.L. Spearman, R.Q. Zhang, Optimal lead time policies, Management Science 45 (1999) 290–295.

[30] A. Stolyar, K. Ramanan, Largest weighted delay first scheduling: Large deviations and optimality, 1999, preprint.

[31] A. Stolyar, Control of end-to-end delay tails in a multiclass network: LWDF discipline optimality, 2000, preprint.

[32] J.A. Van Mieghem, Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule, Annals of Applied Probability 5 (1995) 809–833.

[33] J.A. Van Mieghem, Due-date scheduling: Asymptotic optimality of generalized longest queue and generalized largest delay rules, Operations Research 51 (1) (2003) 113–122.

[34] L.M. Wein, Scheduling semiconductor wafer fabrication, IEEE Transactions of Semiconductor Manufacturing 1 (1988) 115–130.

[35] L.M. Wein, Due-date setting and priority sequencing in a multiclass M/G/1 queue, Management Science 37 (1991) 834–850.

[36] L.M. Wein, P.B. Chevalier, A broader view of the job-shop scheduling problem, Management Science 38 (1992) 1018–1033.

[37] Z.-L. Zhang, Large deviations and the generalized processor sharing scheduling for a two-queue system, Queueing Systems 26 (1997) 229–264.