INFORMS Journal on Computing
Articles in Advance, pp. 1–15
ISSN 1091-9856 (print) | ISSN 1526-5528 (online)

http://dx.doi.org/10.1287/ijoc.1120.0547

Algorithms for Time-Varying Networks of Many-Server Fluid Queues

Yunan Liu
Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, North Carolina, 27695, yunan_liu@ncsu.edu

Ward Whitt
Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027, ww2040@columbia.edu

Motivated by large-scale service systems with network structure, we introduced in a previous paper a time-varying open network of many-server fluid queues with customer abandonment from each queue and time-varying proportional routing among the queues, and showed how performance functions can be determined. The deterministic fluid model serves as an approximation for the corresponding non-Markovian stochastic network of many-server queues with Markovian routing, experiencing periods of overloading at the queues. In this paper we develop a new algorithm for the previous model and generalize the model to include non-exponential service-time distributions. In this paper we report results of implementing the algorithms and studying their computational complexity. We also conduct simulation experiments to confirm that the algorithms are effective in computing the performance functions and that these performance functions provide useful approximations for the corresponding stochastic models.

Key words: queues with time-varying arrival rates; nonstationary queues; queueing networks; many-server queues; deterministic fluid models; fluid approximation; nonstationary networks of fluid queues; customer abandonment; non-Markovian queues
History: Accepted by Winfried Grassmann, Area Editor for Computational Probability and Analysis; received February 2012; revised May 2012; accepted September 2012. Published online in *Articles in Advance*.

1. Introduction

Service systems typically have time-varying arrival rates, with significant variation over the day, that inhibit application of traditional stochastic modeling and analysis. Thus operations researchers have developed a growing collection of tools to cope with the time-varying arrival rates to analyze and improve the performance of these systems; see the recent survey by Green et al. (2007). Service systems are also becoming increasingly complex, exhibiting important network structure. Network structure is evident in many applications, e.g., healthcare delivery systems, distributed customer contact centers, and emergency response and relief organizations. Because the customers typically are people, these service systems also commonly have customer abandonment, including nonexponential patience distributions.

These factors motivated us in Liu and Whitt (2011a) to develop a new model incorporating all these features. In particular, we introduced a time-varying open network of many-server fluid queues, which we call a *fluid queue network* (FQNet). The specific model was the $(G_t/M/s_t+GI)^m/M_t$ FQNet, which has m fluid queues, each with a time-varying external arrival rate (the G_t), a time-varying staffing function (the s_t) with unlimited waiting space, exponential service (the M) and abandonment from queue according to a general distribution (the $+GI$), plus time-varying proportional routing from one queue to another (the final M_t). The general patience (time-to-abandon) distribution and service distribution (that appears in one algorithm) lead to considering two-parameter performance functions at each queue, such as $Q(t,y)$, the fluid content in queue at time t that has been so for at most time y, as a function of t and y.

In this paper we extend our previous work in four important directions. First, we solve the more general $(G_t/GI/s_t+GI)^m/M_t$ FQNet with nonexponential service-time distribution, which is important because service time distributions are commonly found to be nonexponential (often lognormal); e.g., see Brown et al. (2005). Second, we develop an entirely new algorithm based on solving an m-dimensional *ordinary differential equation* (ODE) to find the vector of time-varying arrival rates at each queue, for the $(G_t/M/s_t+GI)^m/M_t$ FQNet with exponential service times. Because the single-queue algorithm developed in Liu and Whitt (2012a) requires solving an ODE for the head-of-line waiting time, this new ODE method is valuable because it provides a unified

ODE framework for the entire analysis. Third, we show that the new ODE framework allows us to give closed-form expression for the arrival rates at each queue in the case of a two-queue network. Finally, we implement all the FQNet algorithms for the first time here and study their computational complexity, thus verifying that they can be efficiently applied. In particular, we compare the two different algorithms for solving the $(G_t/M/s_t + GI)^m/M_t$ FQNet and reveal the advantages of each. We study how the algorithms perform for large networks by considering a family of networks with m queues, with m going up to 160.

The FQNets studied here are deterministic fluid models so that the performance is necessarily described by deterministic functions. Nevertheless, these fluid models are intended for applications to systems that evolve with considerable uncertainty, as commonly captured by stochastic models with stochastic arrival processes, service times, abandonment, and routing. The fluid models can provide useful information when the *predictable* (deterministic) variation in arrival rates and other model elements dominates or is comparable to the *unpredictable* (stochastic) variation because of uncertainty. This tends to be the case when the system experiences periods of overloading. Accordingly, the fluid models here are analyzed under the assumption that the system alternates between successive *overloaded* (OL) and *underloaded* (UL) intervals. This behavior commonly occurs when it is too difficult or costly to dynamically adjust staffing in response to time-varying arrival rates to precisely balance supply and demand at all times—commonly occuring in healthcare.

FQNets are legitimate models in their own right, but they also are intended to serve as approximations for corresponding non-Markovian *stochastic queueing networks* (SQNets), where the M_t routing becomes time-varying Markovian routing; a departure from queue i at time t goes (instantaneously) next to queue j with probability $P_{i,j}(t)$, independent of the system history up to that time. In the FQNet, a proportion $P_{i,j}(t)$ of the fluid flow out of queue i at time t goes next to queue j. In the SQNet, service times and patience times are random times for individual customers. In the FQNet, they specify flow proportions; i.e., with patience *cumulative distribution function* (cdf) F_i at queue i, $F_i(t)$ represents the proportion of all fluid that abandons by time t after it joins the queue, if it has not already entered service. For the associated non-Markovian SQNets, there are few useful analysis tools besides discrete-event stochastic simulation. We envision the FQNets here being used in performance analysis together with simulation of associated SQNets. The FQNets can be analyzed much more rapidly, and so may be used efficiently in preliminary analyses, e.g., to efficiently derive candidate staffing functions at all queues. Simulation of SQNets can then be applied to verify and refine the FQNet analysis.

There is a body of important related literature. First, there is a long history of fluid queue models (Newell 1982). Second, among the limited literature on SQNets with time-varying arrival rates, an important contribution was made by Mandelbaum et al. (1998), who established many-server heavy-traffic limits for Markovian SQNets, showing that FQNets and associated diffusion process refinements arise in the many-server heavy-traffic limit, in which the arrival rate and staffing are both allowed to grow; see also Mandelbaum et al. (1999a, b). Detailed analysis can also be successfully performed for *infinite-server* (IS) SQNets, having infinitely many servers at each queue. Markovian IS SQNets were studied by Massey and Whitt (1993), and IS SQNets with time-varying phase-type (PH_t) distributions were studied by Nelson and Taaffe (2004a, b). Nelson and Taaffe (2004a, b) investigated $(PH_t/PH_t/\infty)^m$ SQNets with multiple customer classes and time-varying phase-type arrival and service processes. They showed that this IS network with k classes is mathematically equivalent to k single-class IS networks, each of which is furthermore equivalent to the $PH_t/PH_t/\infty$ IS model with a modified service distribution. They therefore directly applied the numerical algorithm they first developed for the $PH_t/PH_t/\infty$ model to the $(PH_t/PH_t/\infty)^m$ SQNets. Paralleling that analysis technique, we demonstrate how the algorithm for the single $G_t/GI/s_t + GI$ fluid queue in Liu and Whitt (2012a) can be applied to the $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNet.

The motivation and theory for non-Markovian single many-server fluid queues was given by Whitt (2006) and Liu and Whitt (2011a, b; 2012a; 2013). Those works include extensive comparisons with simulations of stochastic models and supporting heavy-traffic limit theorems. Kang and Pang (2011) developed an alternative algorithm for a fluid queue based on a random-measure perspective that does not require alternating OL and UL intervals, but so far requires constant staffing (which can be applied more generally in a piecewise-constant manner).

We evaluate the performance of the algorithms by implementing them and conducting simulation experiments for associated SQNets for several examples. To relate the FQNets to associated SQNets, we use many-server heavy-traffic scaling, as in Liu and Whitt (2012b, 2013) and references therein. Thus, for a stochastic queue indexed by *scale parameter* n, we let the arrival rate be $n\lambda(t)$ and the number of servers be $\lceil ns(t) \rceil$, where $\lambda(t)$ and $s(t)$ are the fluid model counterparts, and $\lceil x \rceil$ is the least integer greater than or equal to x.

We illustrate now with an example of a two-queue SQNet as depicted in Figure 1 of Liu and

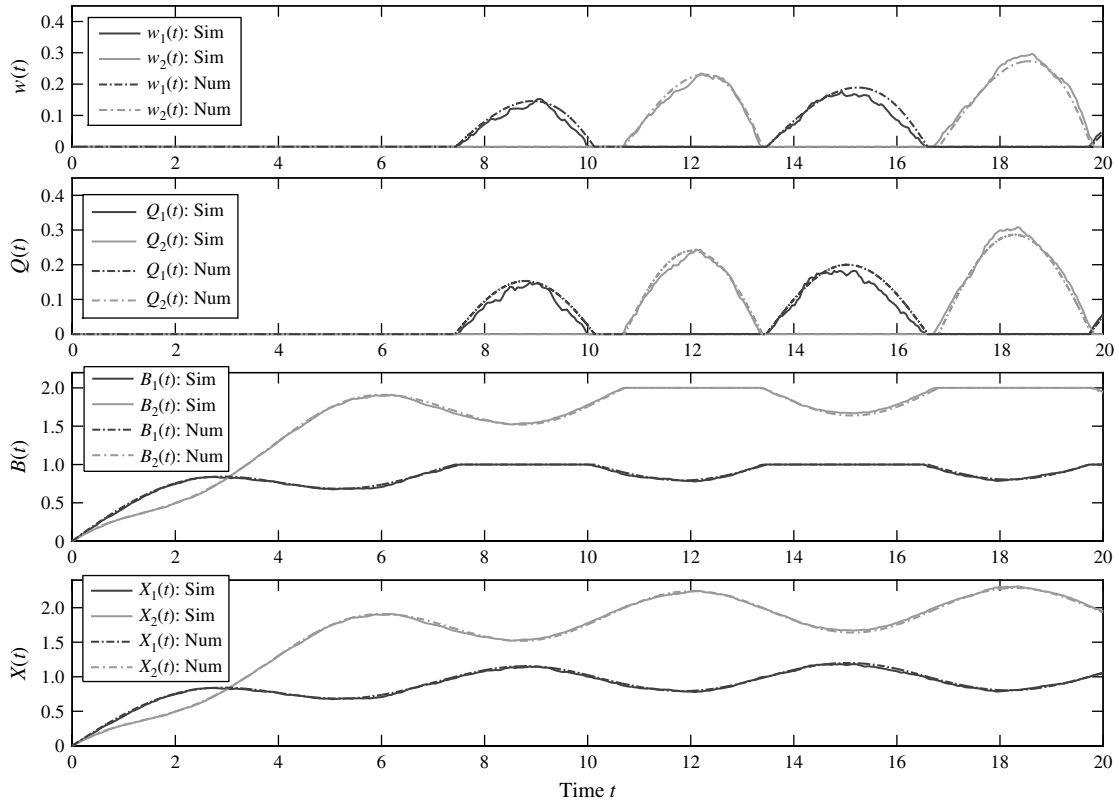

Figure 1 **A Comparison of Performance Functions in the Two-Queue FQNet with Single Sample Paths from a Simulation of the Corresponding SQNet with Scale Parameter $n = 4{,}000$**

Whitt (2011a); see §6.2 for details about this example. Figure 1 compares the fluid approximation (the dashed lines) with simulation estimates of the performance in the stochastic model (the solid lines) for $n = 4{,}000$. We plot single sample paths of the following processes: (i) the elapsed waiting time of the customer at the head of the line, $W_n(t)$; (ii) the scaled number of customers waiting in queue, $\bar{Q}_n(t) \equiv Q_n(t)/n$; (iii) the scaled number of customers in service, $\bar{B}_n(t) \equiv B_n(t)/n$; and (iv) the scaled total number of customers in the system, $\bar{X}_n(t) \equiv X_n(t)/n$. For this extremely large value of n, there is little variability in the simulation sample paths. Figure 1 shows that each simulated sample path falls right on top of the FQNet approximation. The close agreement confirms that both the numerical algorithm and the simulation must be done correctly, and it empirically validates the many-server heavy-traffic limit.

For more realistic stochastic models with fewer servers, the fluid performance functions serve as approximations for the *mean values* of the corresponding stochastic processes. A figure nearly identical to Figure 1 (Figure 8 in the online supplement, available at http://dx.doi.org/10.1287/ijoc.1120.0547) shows that the fluid model provides excellent approximations for the mean values of the same example with $n = 50$. Then the solid lines become simulation estimates of the mean of these scaled stochastic processes, obtained by averaging multiple independent sample paths.

The rest of this paper is organized as follows. In §2 we review the single $G_t/M_t/s_t + GI_t$ fluid queue studied in Liu and Whitt (2011a, 2012a). In §3 we review the $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNet and its results developed in Liu and Whitt (2011a). We also specify the first *fixed point equation* (FPE)-based algorithm, Alg(FPE), in §3.2. In §4 we develop the alternative algorithm, Alg(ODE), based on solving an m-dimensional ODE. In §5 we develop the new FPE-based algorithm, Alg(FPE, GI), for the $(G_t/GI/s_t + GI_t)^m/M_t$ model with general service-time distributions at each queue. In §6 we demonstrate the performance of the algorithms by considering several examples. We also confirm conclusions drawn about the computational complexity. Additional material appears in the online supplement, including a discussion about checking for violation of staffing feasibility.

2. The $G_t/M_t/s_t + GI_t$ Single Fluid Queue

In this section we review the $G_t/M_t/s_t + GI_t$ fluid queue model and its performance; see Liu and Whitt (2011a, 2012a) for more details.

2.1. Specifying the Model

A single fluid queue is a service facility with finite capacity and an associated waiting room or queue with unlimited capacity. Fluid is a deterministic, divisible, and incompressible quantity that arrives over time. Fluid input flows directly into the service facility if there is free capacity available; otherwise it flows into the queue. Fluid leaves the queue and enters service in a first-come, first-served (FCFS) manner whenever service capacity becomes available. There cannot be simultaneously free service capacity and positive queue content.

The staffing function (service capacity) is an absolutely continuous positive function $s(t)$ with derivative $s'(t)$. The service capacity is exogenously specified, providing a hard constraint. In general, there is no guarantee that some fluid that has entered service will not be later forced to leave without completing service, because we allow s to decrease. We directly assume that phenomenon does not occur; i.e., we directly assume that the given staffing function is *feasible*. However, Liu and Whitt (2011a, Theorem 6) show how to construct a minimum feasible staffing function greater than or equal to an initial infeasible staffing function.

The total fluid input over an interval $[0, t]$ is $\Lambda(t)$, the integral of a positive arrival rate function $\lambda(t)$. Service and abandonment occur deterministically in proportions. Because the service is M_t, the proportion of fluid in service at time t that will still be in service at time $t+x$ is

$$\bar{G}_t(x) = e^{-M(t,\,t+x)}, \quad \text{where} \quad M(t,t+x) \equiv \int_t^{t+x} \mu(y)\,dy, \tag{1}$$

for $t \geq 0$ and $x \geq 0$. The cdf of the service time of a quantum of fluid that enters service at time t is $G_t \equiv 1 - \bar{G}_t(x)$; $\bar{G}_t(x)$ is the complementary cdf (ccdf). The cdf G_t has density $g_t(x) = \mu(t+x)\bar{G}_t(x)$ and hazard rate $h_{G_t}(x) = \mu(t+x)$, $x \geq 0$.

The model allows for abandonment of fluid waiting in the queue. In particular, a proportion $F_t(x)$ of any fluid to enter the queue at time t will abandon by time $t+x$ if it has not yet entered service, where F_t is a cdf with density $f_t(y)$ for each t. Let $h_{F_t}(y) \equiv f_t(y)/\bar{F}_t(y)$ be the hazard rate associated with the patience (abandonment) cdf F_t.

System performance is described by a pair of two-parameter deterministic functions $(\hat{B}, \hat{Q})$, where $\hat{B}(t,y)$ $(\hat{Q}(t,y))$ is the total quantity of fluid in service (in queue) at time t that has been so for a *duration* at most y, for $t \geq 0$ and $y \geq 0$. (Alternatively, $(\hat{B}, \hat{Q})$ can be regarded as a pair of time-varying measures.) These functions were shown to be absolutely continuous in the second parameter, so that

$$\hat{B}(t,y) \equiv \int_0^y b(t,x)\,dx \quad \text{and} \quad \hat{Q}(t,y) \equiv \int_0^y q(t,x)\,dx, \tag{2}$$

for $t \geq 0$ and $y \geq 0$. Performance is primarily characterized through the pair of two-parameter fluid content densities (b, q). Let $B(t) \equiv \hat{B}(t,\infty)$ and $Q(t) \equiv \hat{Q}(t,\infty)$ be the total fluid content in service and in queue, respectively. Let $X(t) \equiv B(t) + Q(t)$ be the total fluid content in the system at time t. Because service is assumed to be M_t, the performance will primarily depend on b via B. (We will not directly discuss $\hat{B}$.) The total service completion rate and abandonment rate at time t are

$$\sigma(t) \equiv \int_0^\infty b(t,x) h_{G_t}(x)\,dx = B(t)\mu(t), \quad t \geq 0, \tag{3}$$

$$\alpha(t) \equiv \int_0^\infty b(t,x) h_{F_t}(x)\,dx, \tag{4}$$

respectively. The total amount of fluid to complete service in the interval $[0, t]$ is

$$S(t) \equiv \int_0^t \sigma(y)\,dy = \int_0^t B(y)\mu(y)\,dy, \quad t \geq 0. \tag{5}$$

Because fluid in service (queue) that is not served (does not abandon or enter service) remains in service (queue), the fluid content densities b and q must satisfy the equations

$$b(t+u, x+u) = b(t,x)\frac{\bar{G}_{t-x}(x+u)}{\bar{G}_{t-x}(x)} = b(t,x)e^{-M(t,\,t+u)}, \tag{6}$$

$$q(t+u, x+u) = q(t,x)\frac{\bar{F}_{t-x}(x+u)}{\bar{F}_{t-x}(x)}, \quad 0 \leq x+u < w(t), \tag{7}$$

for $t \geq 0$, $x \geq 0$, and $u \geq 0$, where M is defined in (1), and $w(t)$ is the *boundary waiting time* (BWT) at time t,

$$w(t) \equiv \inf\{x > 0\colon q(t,y) = 0 \text{ for all } y > x\}. \tag{8}$$

(By Assumptions 7–9 of Liu and Whitt 2011a, we never divide by zero in (6) and (7). Because the service discipline is FCFS, fluid leaves the queue to enter service from the right boundary of $q(t,x)$.)

Let $A(t)$ be the total amount of fluid to abandon, and let $E(t)$ be the total amount of fluid to enter service in $[0, t]$. For each t, we have the *flow conservation equations*

$$Q(t) = Q(0) + \Lambda(t) - A(t) - E(t) \quad \text{and} \quad B(t) = B(0) + E(t) - S(t). \tag{9}$$

The abandonment satisfies

$$A(t) \equiv \int_0^t \alpha(y)\,dy, \quad \alpha(t) \equiv \int_0^\infty q(t,y) h_{F_{t-y}}(y)\,dy \tag{10}$$

for $t \geq 0$, where $\alpha(t)$ is the abandonment rate at time t and $h_{F_t}(y)$ is the hazard rate associated with the

patience cdf F_t. (Recall that F_t is defined for t extending into the past.) The flow into service satisfies

$$E(t) \equiv \int_0^t b(u,0)\,du, \quad t \geq 0, \tag{11}$$

where $b(t,0)$ is the rate fluid enters service at time t. If the system is OL, then the fluid to enter service is determined by the *rate that service capacity becomes available* at time t:

$$\eta(t) \equiv s'(t) + \sigma(t) = s'(t) + B(t)\mu(t), \quad t \geq 0. \tag{12}$$

Then $\eta(t)$ coincides with the *maximum possible rate that fluid can enter service* at time t:

$$\gamma(t) \equiv s'(t) + s(t)\mu(t). \tag{13}$$

To describe waiting times, let the BWT $w(t)$ be the delay experienced by the quantum of fluid at the head of the queue at time t, already given in (8), and let the *potential waiting time* (PWT) $v(t)$ be the virtual delay of a quantum of fluid arriving at time t under the assumption that the quantum has infinite patience. Proper definitions of q, w, and v are somewhat complicated, because w depends on q, and q depends on w, but that has been done in §7 in Liu and Whitt (2012a).

The initial conditions are specified via the initial fluid densities $b(0,x)$ and $q(0,x)$, $x \geq 0$. Then $\hat{B}(0,y)$ and $\hat{Q}(0,y)$ are defined via (2), and $B(0) \equiv \hat{B}(0,\infty)$ and $Q(0) \equiv \hat{Q}(0,\infty)$ as before. Let $w(0)$ be defined in terms of $q(0,\cdot)$ as in (8). We assume that $B(0)$, $Q(0)$ and $w(0)$ are finite. In summary, the sextuple $(\lambda(t),\ s(t),\ \mu(t),\ F_t(x),\ b(0,x),\ q(0,x))$ of functions of the variables t and x specifies the *model data* that we assume is suitably smooth; see Assumption 5 of Liu and Whitt (2011a). The system performance is characterized by $(b(t,x),\ q(t,x),\ w(t),\ v(t),\ \alpha(t),\ \sigma(t))$.

We analyze the fluid queue by considering alternating intervals over which the system is either UL or OL, where these intervals include what is usually regarded as critically loaded. In particular, an interval starting at time t_0 with (i) $Q(t_0) > 0$ or (ii) $Q(t_0) = 0$, $B(t_0) = s(t_0)$ and $\lambda(t_0) > s'(t_0) + \sigma(t_0)$, is OL. Let $\mathscr{R}$ denote the current system regime; e.g., we write $\mathscr{R}(t_0) \equiv \mathrm{OL}$. The OL interval ends at the *OL termination time*:

$$T_{\mathrm{OL}}(t_0) \equiv \inf\{u \geq t_0\colon Q(u) = 0 \text{ and } \lambda(u) \leq s'(u) + \sigma(u)\}. \tag{14}$$

Case (ii), where $Q(t_0) = 0$ and $B(t_0) = s(t_0)$, is often regarded as critically loaded, but because the arrival rate $\lambda(0)$ exceeds the rate that new service capacity becomes available, $s'(t_0) + \sigma(t_0)$, we must have the right limit $Q(t_0+) > 0$, so that there exists $\epsilon > 0$ such that $Q(u) > 0$ for all $u \in (0, 0+\epsilon)$. Hence, we necessarily have $T_{\mathrm{OL}}(t_0) > 0$.

An interval starting at time t_0 with (i) $Q(t_0) < 0$ or (ii) $Q(t_0) = 0$, $B(t_0) = s(t_0)$, and $\lambda(t_0) \leq s'(t_0) + \sigma(t_0)$ is UL, designated by $\mathscr{R}(t_0) = \mathrm{UL}$. The UL interval ends at *UL termination time*:

$$T_{\mathrm{UL}}(t_0) \equiv \inf\{u \geq t_0\colon B(u) = s(u) \text{ and } \lambda(u) > s'(u) + \sigma(u)\}. \tag{15}$$

As before, case (ii), in which $Q(t_0) = 0$, and $B(t_0) = s(0)$, is often regarded as critically loaded, but because the arrival rate $\lambda(t_0)$ does not exceed the rate that new service capacity becomes available, $\eta(t_0) \equiv s'(t_0) + \sigma(t_0)$, we must have the right limit $Q(t_0+) = 0$. The UL interval may contain subintervals that are conventionally regarded as critically loaded; i.e., we may have $Q(t) = 0$, $B(t) = s(t)$, and $\lambda(t) = s'(t) + \sigma(t)$. For the fluid models, such critically loaded subintervals can be treated the same as UL subintervals. However, unlike an overloaded interval, we cannot conclude that we necessarily have $T_{\mathrm{UL}}(t_0) > 0$ for a UL interval. Moreover, even if $T_{\mathrm{UL}}(t_0) > 0$ for each UL interval, we could have infinitely many switches between OL intervals and UL intervals in a finite interval. Thus we make assumptions to ensure that those pathological situations do not occur; see §3 of Liu and Whitt (2011a). In general, the termination time of the current interval is defined by

$$T_{\mathscr{R}}(t_0) \equiv T_{\mathrm{OL}}(t_0)\mathbf{1}_{\{\mathscr{R}(t_0)=\mathrm{OL}\}} + T_{\mathrm{UL}}(t_0)\mathbf{1}_{\{\mathscr{R}(t_0)=\mathrm{UL}\}}. \tag{16}$$

2.2. The Performance Formulas

From the basic performance vector $\hat{\mathscr{P}}(t) \equiv (b(t,\cdot),\ q(t,\cdot))$ and the definitions in §2.1, we can easily compute the performance vector

$$\mathscr{P}(t) \equiv \big(\hat{\mathscr{P}}(t), w(t), v(t), B(t), Q(t), X(t), \sigma(t), S(t), \alpha(t), A(t), E(t)\big). \tag{17}$$

We now review the way the basic functions (b, q, w, v) can be computed from the model data $\mathscr{D} \equiv (\lambda, s, \mu, F, \hat{\mathscr{P}}(0))$. For the fluid model with unlimited service capacity starting at time 0,

$$b(t,x) = e^{-M(t-x,\,t)}\lambda(t-x)1_{\{x\leq t\}} + e^{-M(0,\,t)}b(0,x-t)1_{\{x>t\}}, \tag{18}$$

$$B(t) = \int_0^t e^{-M(t-x,\,t)}\lambda(t-x)\,dx + B(0)e^{-M(0,\,t)}, \quad t \geq 0,$$

for M in (1). The same formulas apply to a UL finite-capacity system over $[0, T)$, where $T \equiv \inf\{t \geq 0\colon B(t) > s(t)\}$, with $T = \infty$ if the infimum is never obtained. In an OL interval, $B(t) = s(t)$ and

$$b(t,x) = \big(s'(t-x) + s(t-x)\mu(t-x)\big)e^{-M(t-x,\,t)}1_{\{x\leq t\}} + b(0,x-t)e^{-M(0,\,t)}1_{\{x>t\}}. \tag{19}$$

Let $\tilde{q}(t,x)$ be $q(t,x)$ during an OL interval $[0,T]$ under the assumption that no fluid enters service from queue. During an OL interval,

$$\begin{aligned}\tilde{q}(t,x) &= \lambda(t-x)\bar{F}_{t-x}(x)1_{\{x\le t\}} \\ &\quad + q(0,x-t)\frac{\bar{F}_{t-x}(x)}{\bar{F}_{t-x}(x-t)}1_{\{t<x\}}; \\ q(t,x) &= \tilde{q}(t-x,0)\bar{F}_{t-x}(x)1_{\{x\le w(t)\wedge t\}} \\ &\quad + \tilde{q}(0,x-t)\frac{\bar{F}_{t-x}(x)}{\bar{F}_{t-x}(x-t)}1_{\{t<x\le w(t)\}}; \qquad (20)\\ &= \lambda(t-x)\bar{F}_{t-x}(x)1_{\{x\le w(t)\wedge t\}} \\ &\quad + q(0,x-t)\frac{\bar{F}_{t-x}(x)}{\bar{F}_{t-x}(x-t)}1_{\{t<x\le w(t)\}}.\end{aligned}$$

We characterize the BWT w appearing in the formula for q above by equating the quantity of new fluid admitted into service in the interval $[t,t+\delta)$ to the amount of fluid removed from the right boundary of $q(t,x)$ that does not abandon in $[t,t+\delta)$. By careful analysis (Liu and Whitt 2012a, Theorem 3), that leads to the nonlinear first-order ODE

$$w'(t) = \Omega(t,w(t)) \equiv 1 - \frac{\gamma(t)}{\tilde{q}(t,w(t))} \qquad (21)$$

for γ in (13). (By Assumptions 6–9 of Liu and Whitt 2011a, there is no division by 0 in (20) and (21). Overall, w is continuously differentiable everywhere except for finitely many t.) The end of an OL interval is the first time t that $w(t)=0$ and $\lambda(t)\le s'(t)+s(t)\mu(t)$. During an OL interval, the PWT v is finite and is characterized as the unique solution of the equation

$$v(t-w(t)) = w(t) \quad \text{for all } t\ge 0. \qquad (22)$$

2.3. The Fluid Algorithm for Single Queues

The previous results yield an efficient algorithm to compute the basic performance four-tuple (b,q,w,v) over a finite interval $[0,T]$ that we call the *fluid algorithm for single queues* (FASQ). First, for each UL interval, we compute b directly via (18), terminating the first time we obtain $B(t) > s(t)$. Second, for each OL interval, we compute b via (19), $\tilde{q}$ via (20), and then the BWT w by solving the ODE (21). We consider terminating the OL interval when $w(t)=0$. We actually do terminate the OL interval if $\lambda(t)\le s'(t)+s(t)\mu(t)$. The proof of Theorem 5 in Liu and Whitt (2012a) provides an elementary algorithm to compute v during an OL interval from (22) once w has been computed. Theorem 6 of Liu and Whitt (2012a) shows that v satisfies its own ODE under additional regularity conditions.

The key step beyond direct computation is to control the switching between UL and OL intervals. This can be done by selecting a fixed *switching step size* ΔT over which to perform all calculations before checking to see if there is a regime change. Starting at time t in regime $\mathscr{R}(t)$, the calculations are performed over the interval $[t,t+\Delta T]$. Then the algorithm finds the first time s in $(t,t+\Delta T]$ at which there is a regime change, if any, and that becomes the new initial time t. If the switching step size ΔT is too large, then there can be much wasted computation. Otherwise, the algorithm tends to be insensitive to the choice of ΔT, as we show in §C of the online supplement.

A formal statement of the single-queue algorithm appears in §C of the online supplement. For a time interval $[0,T]$ with $\mathscr{S}$ regime switches, examples show that the running time of the FASQ tends to be linear in both T, for fixed $\mathscr{S}$, and $\mathscr{S}$, for fixed T, and independent of ΔT, provided that ΔT is suitably small, e.g., if $\Delta T \le T/\mathscr{S}$, assuming that the switching points are approximately uniformly distributed throughout the interval $[0,T]$. Thus, for a fixed density of switches per time, the run time should be $O(T^2)$, because $\mathscr{S}$ would be proportional to T. These observations are illustrated by a numerical example in §C of the online supplement.

3. The $(G_t/M_t/s_t+GI_t)^m/M_t$ Fluid Network

We now review the $(G_t/M_t/s_t+GI_t)^m/M_t$ FQNet introduced by Liu and Whitt (2011a) and the FPE-based algorithm to compute all transient performance functions proposed there.

3.1. Model Properties

There are m queues, where each queue has model parameters as given in §2.1. In addition, a proportion $P_{i,j}(t)$ of the fluid output from queue i at time t is routed immediately to queue j, and a proportion $P_{i,0}(t) \equiv 1-\sum_{j=1}^m P_{i,j}(t) \le 1$ is routed out of the network. Consistent with the terminology, we assume that $P(t)$ is substochastic for each t.

If two input streams are combined to form a single input (superposition), then the arrival rate functions are added. If one stream with arrival rate function λ is split, such that a proportion $p(t)$ of that stream goes into a new split stream at time t, then the arrival rate function of the split stream is $\lambda_p(t) \equiv \lambda(t)p(t)$. Similarly, if the departure flow from one queue becomes input to another, then the resulting arrival rate function is σ. (We do not let the abandonment flow from one queue become input to another.) We next discuss converting departure rate into new input rate.

As in open queueing networks, there is an external exogenous arrival rate function to each queue (from outside the network, which could be null at some queues), denoted by $\lambda_j^{(0)}$, and there is a *total arrival rate* (TAR) function to each queue (which we simply

call the arrival rate function), taking into account the flow from other queues, denoted by λ_j. The external arrival rate functions are part of the model data. The arrival rate functions satisfy the system of *traffic rate equations*

$$\lambda_j(t) = \lambda_j^{(0)}(t) + \sum_{i=1}^{m} \sigma_i(t) P_{i,j}(t), \tag{23}$$

where

$$\sigma_i(t) = B_i(t)\mu_i(t), \quad t \geq 0. \tag{24}$$

Equations (23) and (24) produce a system of equations, with λ_j depending upon σ_i for $1 \leq i \leq m$, whereas σ_i in turn depends on λ_i for each i, because B_i depends on λ_i. The formulas for B_i as a function of λ_i have been given in §2.2, provided that we know whether the queue is OL or UL. That requirement is the major source of complexity.

Because (23) is a linear equation, it can be written in matrix notation as $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(0)} + \boldsymbol{\sigma}\boldsymbol{P}$ by omitting the argument t as below, provided that the product σP is interpreted as in (23). Moreover, we can combine (23) and (24) to express λ as the solution of a fixed point equation. Hence the vector $B(t) \equiv B_1(t), \ldots, B_m(t)$ is a function of λ over $[0, t)$ and the model data. Hence, we can express (23) and (24) abstractly as $\boldsymbol{\lambda} = \Psi(\boldsymbol{\lambda})$, where $\Psi(x)(t)$ depends on its argument x only over $[0, t]$ for each $t \geq 0$. Here the function Ψ depends on all the model data $(\lambda_i^{(0)}, s_i, \mu_i, F_{i,\cdot}, b_i(0,\cdot), q_i(0,\cdot), P)$, $1 \leq i \leq m$.

We assume that there are only finitely many switches between OL and UL intervals in each finite interval $[0, T]$. Under that assumption, the operator Ψ mentioned above is a monotone contraction operator, by Liu and Whitt (2011a, Theorem 10). Therefore, a recursive algorithm can be developed. If the recursion starts with initial vector $\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^{(0)}$, the vector of external arrival rate functions, then the kth iterate $\lambda_j^{(k)}$ is the arrival rate of fluid that has previously experienced k transitions in the fluid network. With this notation, we can write the recursive formulas

$$\begin{aligned}\lambda_j^{(n)}(t) &= \Psi^{(n)}(\lambda^{(0)})_j(t) \\ &= \lambda_j^{(0)}(t) + \sum_{i=1}^{m} \sigma_i^{(n-1)}(t) P_{i,j}(t), \quad n \geq 1, \end{aligned} \tag{25}$$

where $\sigma_i^{(n)}(t) = B_i^{(n)}(t)\mu_i(t)$, $n \geq 0$. Because necessarily $\lambda_i^{(1)} \geq \lambda_i^{(0)}$ for each i, this recursion converges monotonically to the fixed point $\boldsymbol{\lambda}$.

3.2. The FPE-Based Algorithm Alg(FPE)

The algorithm Alg(FPE) consists of two successive steps: (i) solving the traffic-rate Equations (23) and (24) and (ii) solving for the performance vector $(b, q, w, v, \sigma, \alpha)$ at each queue using the algorithm in §2.3. For step (i), we start with an initial vector of arrival rate functions, which can be an initial rough estimate of the final arrival rate functions or the given external arrival rate functions. We then apply the performance formulas in §2.2 to determine the performance functions B_i and σ_i at each queue to determine a new vector of arrival rate functions. We then iteratively calculate successive vectors of arrival rate functions until the difference (measured in the supremum norm over a bounded interval) is suitably small. Then we apply step (ii).

Given a desired duration T of an interval $[0, T]$, we specify the following input data: (i) the model parameter vector

$$\begin{aligned}\left(\boldsymbol{\lambda}^{(0)}, \boldsymbol{s}, \boldsymbol{G}, \boldsymbol{F}, \boldsymbol{\mathcal{P}}(0)\right) \equiv \big(&\lambda_i^{(0)}(t), s_i(t), G_i, F_i, \mathcal{P}_i(0), \\ &1 \leq i \leq m,\ t \in [0, T]\big),\end{aligned} \tag{26}$$

where the initial performance vector (at time 0) of queue i, $1 \leq i \leq m$, is

$$\begin{aligned}\mathcal{P}_i(0) \equiv \big(&b_i(0,\cdot), q_i(0,\cdot), B_i(0), Q_i(0), \\ &w_i(0), v_i(0), \alpha_i(0), \sigma_i(0)\big),\end{aligned}$$

and (ii) the algorithm parameters: the iteration *error tolerance parameter* (ETP) ϵ and the *switching step size* ΔT, both assumed to be strictly positive. (We assume that the switching step size is the same for all queues, which usually provides little loss of generality.) We give a formal statement of the algorithm in the online supplement.

From the structure of algorithm Alg(FPE), we can directly determine the *computational complexity* (computer-dependent required run time) $\mathscr{C}_{\text{FPE}} \equiv \mathscr{C}_{\text{FPE}}(\epsilon, T, m, \mathcal{S})$ as a function of the ETP ϵ, number of queues m, length of the time interval T, and the number of regime switches per queue $\mathcal{S}$, but we will also confirm it in numerical examples.

Proposition 1 (Computational Complexity of Alg(FPE)). *The computational complexity of Alg*(FPE) *is*

$$\mathscr{C}_{\text{FPE}} \equiv \mathscr{C}_{\text{FPE}}(m, T, \mathcal{S}, \epsilon) = O(mT\mathcal{S}\log(1/\epsilon)). \tag{27}$$

If we may regard $\mathcal{S} = O(T)$, *as is the case with periodic models, then* $\mathscr{C}_{\text{FPE}}(m, T, \epsilon) = O(mT^2\log(\epsilon))$.

Proof. Let $\mathcal{I} \equiv \mathcal{I}(\epsilon)$ be the number of *iterations* of the FPE as a function of the ETP ϵ. Roughly, we need to apply the FASQ for each of the m queues $\mathcal{I}$ times, although the full FASQ is not needed in the steps before the final one needed to compute the actual performance functions at each queue. Let $\mathcal{S}_i$ be the number of regime switches at queue i over

$[0, T]$. Thus the overall complexity should be $\mathscr{C}_{\mathrm{FPE}} = O(\mathscr{I}T\sum_{i=1}^{m} S_i)$. Assuming that $\mathscr{S}_i \approx \mathscr{S}$ for all i, with the switches at different queues occurring at different times, that yields $\mathscr{C}_{\mathrm{FPE}}(\mathscr{I}, m, T, \mathscr{S}) = O(\mathscr{I}Tm\mathscr{S})$. Moreover, $\mathscr{I}(\epsilon) = O(\log(1/\epsilon)$ where ϵ is the ETP, because the convergence to the fixed point in successive iterations is geometrically fast.

Unfortunately, unlike the other parameters, the number of regime switches per queue $\mathscr{S}$ cannot be directly observed from the model data. However, if the model parameters, such as λ and s, are periodic functions with periods τ_λ and τ_s, then the total number of switchings is usually bounded by $2T/\tau_\lambda + 2T/\tau_s$ so that we may regard $\mathscr{S} = O(T)$ making $\mathscr{C}_{\mathrm{FPE}}(m, T, \epsilon) = O(mT^2 \log(\epsilon))$. □

Proposition 1 is supported by the examples in §6.

4. The Alternative ODE-Based Algorithm Alg(ODE)

Now we develop the new algorithm Alg(ODE) for the $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNet. Again, the key is to compute *total arrival rates* for all queues and then treat each queue independently. In some special cases, analytic formulas are available.

4.1. Finding the Total Arrival Rate Vector

Instead of solving the fixed-point equation, as in §3, to find the TARs, we now solve an m-dimensional ODE. To do that, we need to work over subintervals where all queues are in specified regimes. So now we consider successive switching times for *any* queue in the network. We recursively solve the ODE in each of these intervals. The key is to characterize and update the system regime in different intervals and recursively advance in t. We describe the system regime at t with two sets: $\mathscr{U}(t)$ is the set of indices of queues that are UL, and $\mathscr{O}(t)$ is the set of indices of queues that are OL. In other words,

$$\mathscr{U}(t) \equiv \{1 \le i \le m\colon B_i(t) \le s_i(t),\ Q_i(t) = 0\}, \tag{28}$$

$$\mathscr{O}(t) \equiv \{1 \le i \le m\colon B_i(t) = s_i(t),\ Q_i(t) > 0\}. \tag{29}$$

Of course, $\mathscr{U}(t)$ is simply the complement of $\mathscr{O}(t)$ within the set $\{1, \ldots, m\}$.

Given $\mathscr{U}(t)$ and $\mathscr{O}(t)$, consider $1 \le i \le m$. (i) If queue i is UL, i.e., $i \in \mathscr{U}(t)$, flow conservation implies that

$$B_i'(t) = \lambda_i^{(0)}(t) + \sum_{j\in\mathscr{U}(t)} \mu_j(t)P_{j,i}(t)B_j(t) + \sum_{k\in\mathscr{O}(t)} \mu_k(t)P_{k,i}(t)s_k(t) - \mu_i(t)B_i(t).$$

If $i \in \mathscr{O}(t)$, $B_i(t) = s_i(t)$. We partition and regroup the indices of queues so that $\mathbf{B}(t) \equiv [\mathbf{B}_{\mathscr{U}}(t),\ \mathbf{B}_{\mathscr{O}}(t)]^T$, $\boldsymbol{\lambda}(t) \equiv [\boldsymbol{\lambda}_{\mathscr{U}}(t),\ \boldsymbol{\lambda}_{\mathscr{O}}(t)]^T$, $\boldsymbol{\lambda}^{(0)}(t) \equiv [\boldsymbol{\lambda}^{(0)}_{\mathscr{U}}(t),\ \boldsymbol{\lambda}^{(0)}_{\mathscr{O}}(t)]^T$, $\boldsymbol{\mu}(t) \equiv [\boldsymbol{\mu}_{\mathscr{U}}(t),\ \boldsymbol{\mu}_{\mathscr{O}}(t)]^T$, $\mathbf{s}(t) \equiv [\mathbf{s}_{\mathscr{U}}(t),\ \mathbf{s}_{\mathscr{O}}(t)]^T$, $\boldsymbol{\Gamma}_{\mathscr{U}}(t) \equiv \mathrm{diag}(\boldsymbol{\mu}_{\mathscr{U}}(t))$, $\boldsymbol{\Gamma}_{\mathscr{O}}(t) \equiv \mathrm{diag}(\mu_{\mathscr{O}}(t))$, $\boldsymbol{\Gamma}(t) \equiv \mathrm{diag}(\boldsymbol{\Gamma}_{\mathscr{U}}(t), \boldsymbol{\Gamma}_{\mathscr{O}}(t))$, and

$$\mathbf{P}(t) \equiv \begin{array}{c} \\ \mathscr{U} \\ \mathscr{O} \end{array} \begin{array}{c} \begin{array}{cc} \mathscr{U} & \mathscr{O} \end{array} \\ \left[\begin{array}{c|c} \mathbf{P}_{\mathscr{U}\mathscr{U}}(t) & \mathbf{P}_{\mathscr{U}\mathscr{O}}(t) \\ \hline \mathbf{P}_{\mathscr{O}\mathscr{U}}(t) & \mathbf{P}_{\mathscr{O}\mathscr{O}}(t) \end{array}\right] \end{array},$$

where $\mathbf{P}_{\mathscr{U}\mathscr{U}}(t)$ ($\mathbf{P}_{\mathscr{O}\mathscr{U}}(t)$, $\mathbf{P}_{\mathscr{U}\mathscr{O}}(t)$, and $\mathbf{P}_{\mathscr{O}\mathscr{O}}(t)$) denotes the transition probability from a state in $\mathscr{U}$ ($\mathscr{O}$, $\mathscr{U}$, and $\mathscr{O}$) to a state in $\mathscr{U}$ ($\mathscr{U}$, $\mathscr{O}$, and $\mathscr{O}$) at time t. Let $\mathbf{P}_{\mathscr{O}\mathscr{U}}(t) = \mathbf{P}_{\mathscr{U}\mathscr{O}}(t) = \mathbf{P}_{\mathscr{O}\mathscr{O}}(t) = \boldsymbol{\phi}$ when $\mathbf{P}_{\mathscr{U}\mathscr{U}}(t) = \mathbf{P}(t)$ (i.e., all queues are UL), and let $\mathbf{P}_{\mathscr{O}\mathscr{U}}(t) = \mathbf{P}_{\mathscr{U}\mathscr{O}}(t) = \mathbf{P}_{\mathscr{U}\mathscr{U}}(t) = \boldsymbol{\phi}$ when $\mathbf{P}_{\mathscr{O}\mathscr{O}}(t) = \mathbf{P}(t)$ (i.e., all queues are OL), where ϕ denotes an empty matrix (with rank 0).

Therefore, in matrix notation we have

$$\mathbf{B}_{\mathscr{U}}'(t) = \mathbf{C}(t)\cdot\mathbf{B}_{\mathscr{U}}(t) + \mathbf{D}(t) \quad \text{and} \quad \mathbf{B}_{\mathscr{O}}(t) = \mathbf{s}_{\mathscr{O}}(t), \tag{30}$$

where

$$\mathbf{D}(t) \equiv \boldsymbol{\lambda}^{(0)}_{\mathscr{U}}(t) + \mathbf{P}^T_{\mathscr{O}\mathscr{U}}(t)\boldsymbol{\Gamma}_{\mathscr{O}}(t)\mathbf{s}_{\mathscr{O}}(t),$$
$$\mathbf{C}(t) \equiv (\mathbf{P}^T_{\mathscr{U}\mathscr{U}}(t) - \mathbf{I})\boldsymbol{\Gamma}_{\mathscr{U}}(t).$$

If the service rates and the routing probability matrix are independent of time, $\mu_i(t) = \mu_i$ and $P_{i,j}(t) = P_{i,j}$, i.e., the model becomes the $(G_t/M/s_t + GI_t)^m/M$ network, then $\boldsymbol{\Gamma}_{\mathscr{U}} \equiv \boldsymbol{\Gamma}_{\mathscr{U}}(t) = \mathrm{diag}(\boldsymbol{\mu}_{\mathscr{U}})$, $\mathbf{C} \equiv \mathbf{C}(t) = (\mathbf{P}^T_{\mathscr{U}\mathscr{U}} - \mathbf{I})\boldsymbol{\Gamma}_{\mathscr{U}}$, and (30) has the unique solution

$$\mathbf{B}_{\mathscr{U}}(t) = e^{-\mathbf{C}t}\left(\int_0^t e^{-\mathbf{C}u}\mathbf{D}(u)\,du + \mathbf{B}(0)\right).$$

In all cases, the TAR vector can be represented as

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}^{(0)}(t) + \mathbf{P}^T(t)\boldsymbol{\Gamma}(t)\cdot\mathbf{B}(t). \tag{31}$$

4.2. Explicit Formulas for $m = 2$

The ODE-based approach yields analytic solutions when $m = 2$. Consider the following four system regimes:

(i) When queue 1 is OL and queue 2 is UL (i.e., $B_1(t) = s_1(t)$, $Q_1(t) \ge 0$, $B_2(t) < s_2(t)$),

$$B_1(t) = s_1(t),$$
$$B_2'(t) = \lambda_2^{(0)}(t) + P_{1,2}(t)\mu_1(t)s_1(t) + (P_{2,2}(t) - 1)\mu_2(t)B_2(t),$$

which has a unique solution

$$B_2(t) = e^{\int_0^t (P_{2,2}(u)-1)\mu_2(u)\,du}\left[\int_0^t e^{\int_0^u (P_{2,2}(v)-1)\mu_2(v)\,dv}(\lambda_2^{(0)}(u) + P_{1,2}(u)\mu_1(u)s_1(u))\,du + B_2(0)\right].$$

(ii) When queue 1 is UL and queue 2 is OL (i.e., $B_1(t) < s_1(t)$, $B_2(t) = s_2(t)$, $Q_2(t) \geq 0$),

$$B_1'(t) = \lambda_1^{(0)}(t) + (P_{1,1}(t) - 1)\mu_1(t)B_1(t) + P_{2,1}(t)\mu_2(t)s_2(t),$$
$$B_2(t) = s_2(t),$$

which has a unique solution

$$B_1(t) = e^{\int_0^t (P_{1,1}(u)-1)\mu_1(u)\,du}\left[\int_0^t e^{\int_0^u (P_{2,1}(v)-1)\mu_1(v)\,dv}(\lambda_1^{(0)}(u) + P_{2,1}(u)\mu_2(u)s_2(u))\,du + B_1(0)\right].$$

(iii) When both queues are OL,

$$B_1(t) = s_1(t), \quad B_2(t) = s_2(t).$$

(iv) When both queues are UL,

$$B_1'(t) = \lambda_1^{(0)}(t) + (P_{1,1}(t) - 1)\mu_1(t)B_1(t) + P_{2,1}(t)\mu_2(t)B_2(t),$$
$$B_2'(t) = \lambda_2^{(0)}(t) + P_{1,2}(t)\mu_1(t)B_1(t) + (P_{2,2}(t) - 1)\mu_2(t)B_2(t),$$

or

$$\mathbf{B}'(t) = \boldsymbol{\lambda}^{(0)}(t) + \mathbf{C}(t) \cdot \mathbf{B}(t), \tag{32}$$

where

$$\mathbf{C}(t) \equiv (\mathbf{P}^T(t) - \mathbf{I})\boldsymbol{\Gamma}(t) \quad \text{and} \quad \boldsymbol{\Gamma}(t) \equiv \begin{bmatrix} \mu_1(t) & 0 \\ 0 & \mu_2(t) \end{bmatrix}.$$

After $\mathbf{B}(t)$ is obtained, the TARs are

$$\lambda_1(t) = \lambda_1^{(0)}(t) + P_{1,1}(t)\mu_1(t)B_1(t) + P_{2,1}(t)\mu_2(t)B_2(t),$$
$$\lambda_2(t) = \lambda_2^{(0)}(t) + P_{1,2}(t)\mu_1(t)B_1(t) + P_{2,2}(t)\mu_2(t)B_2(t).$$

4.3. The Overall Algorithm and Its Complexity

Just as for FASQ in §2.3, the key step beyond direct computation is to control the switching between regimes. Because each queue can be either UL or OL, there are overall 2^m different network regimes. We say that the system changes its regime at some time if one or more of the queues changes its regime, either from UL to OL or from OL to UL. We provide the following regime termination time:

$$\begin{aligned} T_{\mathscr{R}}(t_0) &\equiv T_1(t_0) \wedge T_2(t_0), \quad \text{where} \\ T_1(t_0) &\equiv \inf\{t > t_0\text{: some } i \in \mathscr{O}(t_0) \\ &\qquad \text{s.t. } Q_i(t) = 0,\ \lambda_i(t) \leq \sigma_i(t)\}, \\ T_2(t_0) &\equiv \inf\{t > t_0\text{: some } j \in \mathscr{U}(t_0) \\ &\qquad \text{s.t. } B_j(t) = s_j(t),\ \lambda_j(t) > \sigma_j(t)\}, \end{aligned} \tag{33}$$

with t_0 being the starting time of the desired interval and the infimum of an empty set understood to be infinity.

Within each regime, we use an ODE to compute the TARs $\lambda_i(t)$ and the service content functions $B_i(t)$, based on (30) and (31). Given the TARs at all queues, we use the FASQ to calculate the performance functions. We give a formal algorithm statement in §E of the online supplement.

The computational complexity clearly depends largely on the computational complexity of the ODE solver. Fortunately the ODEs arising in the present context tend not to be computationally difficult; e.g., they are rarely stiff. Let $\mathscr{O}_{\text{ODE}}(m, t)$ be the computational complexity for solving an m-dimensional ODE over an interval of length t. For the conventional solvers we use (see §6.1), we should have approximately $\mathscr{O}_{\text{ODE}}(m, t) = O(mt)$. From the structure of algorithm Alg(ODE), we can determine the computational complexity $\mathscr{C}_{\text{ODE}} \equiv \mathscr{C}_{\text{ODE}}(T, m, \mathscr{S})$ as a function of the number of queues m, length of the time interval T, and number of regime switches per queue $\mathscr{S}$, but we will also confirm it in numerical examples.

Proposition 2 (Computational Complexity of Alg(ODE)). *If the computational complexity of the ODE solver is* $\mathscr{O}_{\text{ODE}}(m, t) = O(mt)$, *then the computational complexity of Alg*(ODE) *is*

$$\mathscr{C}_{\text{ODE}} \equiv \mathscr{C}_{\text{ODE}}(T, m, \mathscr{S}) = O(m^2 \mathscr{S} T). \tag{34}$$

Proof. As in §3.2, the parameter pair (m, T) is directly observable, but $\mathscr{S}$ is not. Let $\mathscr{S}_i$ be the number of regime switches at queue i over $[0, T]$. Hence the total number of regime switches for any queue in the network is $\sum_{i=1}^m \mathscr{S}_i$. Assuming that $\mathscr{S}_i \approx \mathscr{S}$ for all i as before, we see that the ODE must be solved $m\mathscr{S}$ times over subintervals, whose combined length is T. In addition, there is some computational cost of carrying out the switching in each regime switch. For the ODE portion of the algorithm, the computational complexity is

$$\mathscr{O}(m, \mathscr{S}, T) = \sum_{j=1}^{m\mathscr{S}} \mathscr{O}(m, T_i), \quad \text{where } \sum_{j=1}^{m\mathscr{S}} T_i = T. \tag{35}$$

Hence, the overall computational complexity for the ODE solver is $O(mT)$. But we must factor in the regime switching, which has computational effort proportional to the number of network regime switches, $O(m\mathscr{S})$. Assuming that these components each contribute significantly, we get the overall computational complexity in (35). □

We find that Proposition 2 is consistent with numerical examples; e.g., see Figure 2.

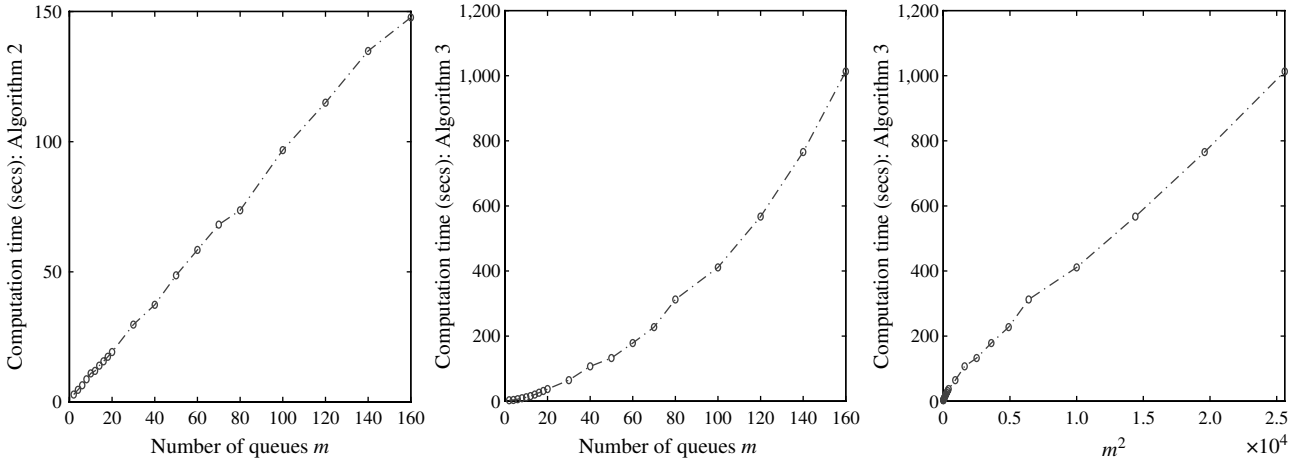

Figure 2 **Computing Times of Algorithms Alg(FPE) and Alg(ODE) for the m-Queue FQNet as a Function of m, $2 \le m \le 160$**

5. Allowing GI Service Distributions: Alg(FPE, GI)

We now generalize the model, allowing the service distribution at each queue to be GI instead of M. We need a new algorithm because neither the FPE-based algorithm Alg(FPE) in §3 nor the ODE-based algorithm Alg(ODE) in §4 is directly applicable. For simplicity, we focus on the $(G_t/GI/s_t + GI)^m/M_t$ FQNet, where the service and patience distributions are not time varying; the analysis can be easily generalized to $(G_t/GI_t/s_t + GI_t)^m/M_t$. As part of the model data, we let $(G_i, 1 \le i \le m)$ be the general service cdfs of the $(G_t/GI/s_t + GI)^m/M_t$ FQNet, and let $\bar{G}_i \equiv 1 - G_i$ be the associated ccdf; e.g., $\bar{G}_i(x) = e^{-\mu_i x}$ for M service.

5.1. A New FPE for the TAR Vector

The key is to obtain the TAR $\lambda_i(t)$ for $1 \le i \le m$ and $0 \le t \le T$. Once $\lambda_i(t)$ is obtained, the single-queue algorithm for GI service developed in Liu and Whitt (2012a) can be applied to compute all other performance measures; see §8 and Appendix G in Liu and Whitt (2012a). This single-queue algorithm for GI service is a generalization of FASQ, which requires solving another FPE to find the rate at which fluid enters service $b(t, 0)$ (which we call the *rate into service* (RIS)) during each OL interval. For M service, this FPE for RIS simplifies to (19) with $x = 0$.

We next analyze the transient dynamics of the $(G_t/GI/s_t + GI)^m/M_t$ model at arbitrary time t assuming the knowledge of the current system status. We refer to the explicit formulas for $b(t, x)$ developed in Liu and Whitt (2012a) during our analysis. The formulas for $q(t, x)$ and $w(t)$ are identical to those in §2.

Consider a queue j that is UL, i.e., $j \in \mathcal{U}(t)$. From Proposition 2 of Liu and Whitt (2012a) we have that (as a generalization of (18)),

$$b_j(t, x) = \bar{G}_j(x)\lambda_j(t-x)\mathbf{1}_{\{x \le t\}} + \frac{\bar{G}_j(x)}{\bar{G}_j(x-t)} b_j(0, x-t)\mathbf{1}_{\{x > t\}},$$

$$\begin{aligned}\sigma_j(t) &= \int_0^\infty b_j(t, x) h_{G,j}(x)\, dx \\ &= \int_0^t g_j(x)\lambda_j(t-x)\, dx \\ &\quad + \int_0^\infty \frac{g_j(x+t)}{\bar{G}_j(x)} b_j(0, x)\, dx. \end{aligned} \tag{36}$$

Note that formula (36) for queue j is in terms of the TAR λ_i, which is unknown.

Consider a queue k that is OL, i.e., $k \in \mathcal{O}(t)$. From Equations (17)–(20) of Liu and Whitt (2012a) we obtain

$$\sigma_k(t) = b_k(t, 0) - s_k'(t), \tag{37}$$

where the RIS $b_k(t, 0)$ satisfies the FPE (as a generalization of (19))

$$b_k(\cdot, 0) = \Phi(b_k(\cdot, 0)), \tag{38}$$

with

$$\Phi(y)(t) \equiv \hat{a}_k(t) + \int_0^t y(t-x) g_k(x)\, dx,$$

$$\hat{a}_k(t) \equiv s_k'(t) + \int_0^\infty \frac{b_k(0, y) g_k(t+y)}{\bar{G}_k(y)}\, dy.$$

Moreover, we have shown in Liu and Whitt (2012a, Theorem 2) that Φ is a contraction operator under mild conditions, and thus implies that the FPE (38) has a unique solution.

We note that the RIS for an OL queue depends on the rate at which the service capacity becomes available (defined in (12)) and is independent of the TAR,

unlike during a UL regime. Hence, having $\sigma_k(t)$ and $b_k(t,0)$ available (by solving the FPE (38) and (37)) for all OL queues (i.e., for all $k \in \mathcal{O}(t)$), the TAR of queue i satisfies the following traffic-rate equation:

$$\lambda_i(t) = \lambda_i^{(0)}(t) + \sum_{k\in\mathcal{O}(t)} P_{k,i}(t)\sigma_k(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\sigma_j(t)$$
$$= \hat{\gamma}_i(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\left(\int_0^t g_j(x)\lambda_j(t-x)\,dx\right), \quad (39)$$

where

$$\hat{\gamma}_i(t) \equiv \lambda_i^{(0)}(t) + \sum_{k\in\mathcal{O}(t)} P_{k,i}(t)\sigma_k(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\int_0^\infty \frac{g_j(x+t)}{\bar{G}_j(x)} b_j(0,x)\,dx,$$

with $\hat{\gamma}_i$ not depending on the TAR and determined by the FPE (38) and the second equality holding by (36).

Equation (39) expresses the TAR vector λ as the solution of an FPE, i.e.,

$$\lambda = \mathcal{J}(\lambda), \quad (40)$$

where $\mathcal{J}\colon \mathbb{D}^m \to \mathbb{D}^m$ with

$$\mathcal{J}(u)_i(t) \equiv \hat{\gamma}_i(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\left(\int_0^t g_j(x)u_j(t-x)\,dx\right), \quad 1 \le i \le m, \quad (41)$$

where $u \equiv (u_1,\ldots,u_m) \in \mathbb{D}^m$. Under regularity conditions, we can show that there exists a unique solution to Equation (39) by applying the Banach contraction theorem. We will use the complete (nonseparable) normed space $\mathbb{D}^m$ with the uniform norm over the interval $[0,T]$, i.e.,

$$\|u\|_T \equiv \sum_{i=1}^m \sup_{0\le t\le T} |u_i(t)|. \quad (42)$$

Theorem 1 (*TAR for GI Service*). *Assume the system regime does not change in a small interval* $[0,T]$, *then the operator* $\mathcal{J}$ *in* (41) *is a monotone contraction operator on* $\mathbb{D}^n$ *with norm defined in* (42).

Proof. Assume that $T > 0$ is small enough so that the system regime does not change, i.e., $\mathcal{U}(t) = \mathcal{U}$ and $\mathcal{O}(t) = \mathcal{O}$ for $0 \le t \le T$. Then

$$\|\mathcal{J}(u_1) - \mathcal{J}(u_2)\|_T$$
$$= \sum_{i=1}^m \sup_{0\le t\le T}\left|\sum_{j\in\mathcal{U}} P_{j,i}(t)\left[\int_0^t g_j(x)(u_{1,j}(t-x) - u_{2,j}(t-x))\,dx\right]\right|$$
$$\le \sum_{i=1}^m \sup_{0\le t\le T}\sum_{j\in\mathcal{U}} \|u_{1,j} - u_{2,j}\|_T P_{j,i}(t)G_j(t)$$
$$\le m \max_{1\le j\le m} G_j(T)\cdot \sup_{0\le t\le T}\sum_{j\in\mathcal{U}} \|u_{1,j} - u_{2,j}\|_T \le \tilde{C}(T)\,\|u_1 - u_2\|_T,$$

where $\tilde{C}(T) \equiv m\max_{1\le j\le m} G_j(T)$. This provides what we need, because we can make $\tilde{C}(T) < 1$ for sufficiently small $T > 0$, because $G_i(t) \to 0$ as $t \to 0$ for all $1 \le i \le m$ by our assumption on the existence of the service densities. □

5.2. The Overall FPE-Based Algorithm with *GI* Service

Algorithm Alg(FPE, GI) has two parts: (i) regime switching and (ii) the new FPE within each fixed network regime. The regime switching can be managed just as for the FASQ and Alg(ODE). As before, we work with a regime switching step size ΔT. Given a time t, we apply the new FPE in §5.1 to find a new TAR vector over the interval $[t, t+\Delta T]$. However, after doing that calculation, we must check to see if there is a regime switch at any queue in the network. If such a regime switch occurs at time $s \in [t, t+\Delta T]$, then we replace t with s and repeat. In this way, we move forward in time until we compute the TAR vector for all of $[0,T]$.

Within each interval with fixed network regime, we calculate the TAR using FPE (40). Given that TAR within each interval with fixed network regime, we apply the single-queue algorithm from Liu and Whitt (2012a) to calculate the queue performance at each queue. This is more complicated than the FASQ in §2, because it is necessary to solve the FPE (38) at each queue that is OL in that particular network regime.

For this last algorithm, the computational complexity is more difficult to determine from the algorithm structure, because the algorithm is more complicated. Just as for Alg(ODE), there are $O(m\mathcal{S})$ network regimes, so that regime switching should have complexity of order $O(m\mathcal{S})$. The new FPE is more complicated, requiring an FPE within the overall FPE at each queue. Because the first-step FPE (38) is done at each queue throughout $[0,T]$, we can estimate its complexity as $O(mT)$. The second-step FPE (40) may also have complexity of order $O(mT)$. In addition, these FPEs depend on the ETPs ϵ. Because both operators are contraction, the rate of convergence is geometric. Hence the computational complexity of both iterations as functions of ϵ are $O(\log(1/\epsilon))$. Thus, we estimate that the computational complexity should be

$$\mathcal{C}_{\mathrm{FPE,GI}}(m,T,\mathcal{S},\epsilon) = O\left((mT+mT)m\mathcal{S}\log\left(\frac{1}{\epsilon}\right)\right)$$
$$= O(m^2\mathcal{S}T\log(1/\epsilon)). \quad (43)$$

6. Examples

In this section we report the results of implementing the algorithms in §§3–5 and applying them to three examples: (i) a Markovian $(M_t/M/s+M)^2/M$ two-queue FQNet, (ii) a Markovian $(M_t/M/s+M)^m/M$

FQNet with m queues, $2 \le m \le 160$, and (iii) a non-Markovian $(G_t/LN/s + E_2)^2/M$ model. For simplicity, in these examples we make only the arrival rate time varying. The extension to time-varying staffing is of course very important and is not difficult to do as well, as we illustrate with an example in the online supplement. Adding time-varying functions to the service, abandonment and routing are less important, so we do not directly illustrate those extensions. The third algorithm applies to all three examples, but the first two algorithms only apply to the first two examples. In §6.1 we first provide details about our implementation.

6.1. Implementation Details

Before discussing the examples, we briefly explain how we implemented the numerical algorithms and conducted the simulation experiments. For both, we used MATLAB on a personal computer. To numerically solve ODEs both one-dimensional for $w(t)$ at each queue as in (21), and multidimensional for the TAR as in (30), we used the MATLAB solvers "ode23" and "ode45," which employ automatic step-size Runge–Kutta–Fehlberg integration methods. The first one, ode23, uses a pair of simple second-order and third-order formulas. The second, ode45, uses a pair of fourth-order and fifth-order formulas. See Thomas (1995) for details on finite-difference methods for numerically solving differential ODEs. As a base case for the examples, we considered a system starting empty over the time interval $[0, T]$ with $T = 20$. In that framework, we divided the continuous time interval $[0, T]$ into discrete intervals with length 0.002.

Care is needed in estimating the various time-dependent performance functions in the simulation experiments. For the mean head-of-line waiting time $E[W(t)]$, the mean queue length $E[Q(t)]$, and the mean number of busy servers $E[B(t)]$, we divide the interval $[0, T]$ into subintervals or bins. For $E[W(t)]$, we keep track of all customer arrivals in each sample path. For a customer n, we keep track of the arrival time A_n and the time that the customer enters service E_n. Therefore, one value for this sample path is $(t, \hat{W}(t)) = (E_n, E_n - A_n)$. Of course, this customer may have already abandoned by time E_n. Because we are interested in the potential waiting time, assuming infinite patience, we keep track of the time that the customer would enter service even after they abandon; i.e., our procedure includes the behavior of virtual customers. The bin size for $E[W(t)]$ is 0.1, whereas the bin size for $E[Q(t)]$ and $E[B(t)]$ is 0.05. Thus, we sampled the queue length once every 0.05 units of time.

6.2. A Two-Queue FQNet Example

We first consider the two-queue $(M_t/M/s + M)^2/M$ FQNet discussed in §1. It has sinusoidal external arrival rates

$$\lambda_i^{(0)}(t) = a_i + b_i \sin(c_i t + \phi_i), \quad i = 1, 2; \tag{44}$$

exponential service and patience distributions $\bar{G}_i(x) = e^{-\mu_i x}$ and $\bar{F}_i(x) = e^{-\theta_i x}$, $i = 1, 2$, respectively; constant staffing functions s_i, $i = 1, 2$; and a constant 2×2 Markov transition probability matrix P with elements $P_{1,2} = P_{2,1} = 0.2$ and $P_{i,i} = 0.3$, so that $P_{i,0} = 0.5$, $i = 1, 2$. Let $a_1 = a_2 = 0.5$, $b_1 = 0.25$, $b_2 = 0.35$, $c_1 = c_2 = 1$, $\phi_1 = 0$, $\phi_2 = -3$, $\mu_1 = 1$, $\mu_2 = 0.5$, $\theta_1 = 0.5$, $\theta_2 = 0.3$, $s_1 = 1$, and $s_2 = 2$. We let the network be initially empty.

We first show how the FPE-based algorithm Alg(FPE) from §3 works. It is based on an FPE for the TARs $\lambda_1(t)$ and $\lambda_2(t)$ for $0 \le t \le T$. Figure 6 in Section G.1 of the online supplement displays the arrival rates in successive iterations, dramatically showing both the monotone convergence and the geometric rate of convergence of the operator Ψ in §3.1. Alg(FPE) terminates after iteration $\mathcal{I}(\epsilon)$, where $\epsilon > 0$ is the prespecified ETP, and

$$\mathcal{I}(\epsilon) \equiv \inf\left\{ n \ge 0\colon \mathcal{E}_T(n) \equiv \max_{j=1,2} \|\lambda_j^{(n)} - \lambda_j^{(n-1)}\|_T \le \epsilon \right\},$$

yielding final TARs $\lambda_i \equiv \lambda_i^{(\mathcal{I}(\epsilon))}$, $i = 1, 2$. For this example, we show how the number of iterations $\mathcal{I}(\epsilon)$, the total run time $\mathcal{T}(\epsilon)$, and the terminating error $\mathcal{E}_T(\mathcal{I}(\epsilon))$ depend on the EPT ϵ in Table 1.

Figure 7 in Section G.1 of the online supplement shows plots of all the standard performance functions in the fluid network using Alg(FPE), including λ_i, Q_i, w_i, B_i, X_i, and $b_i(\cdot, 0)$, $i = 1, 2$. Figure 1 compares the fluid approximations with results from a simulation experiment for a very large-scale queueing system. The queueing model has nonhomogeneous Poisson external arrival processes with sinusoidal rate functions $\lambda_{n,i}^{(0)}(t) = n\lambda_i^{(0)}(t)$, $i = 1, 2$, with $n = 4{,}000$. We compare the fluid model predictions to a single sample path of the queueing system (one simulation

Table 1 **The Number of Iterations $\mathcal{I}(\epsilon)$, Computation Time $\mathcal{T}(\epsilon)$, and Terminating Error $\mathcal{E}_T(\mathcal{I}(\epsilon))$ for Algorithm Alg(FPE) as a Function of the ETP $\epsilon \equiv 10^{-n}$, $n \ge 1$, for the Two-Queue FQNet Example Using $T = 20$ and $\Delta T = 2$**

$\log_{10}(\epsilon)$	−1	−2	−3	−4	−5	−6	−7	−8	−9
$\mathcal{I}(\epsilon)$	3	6	8	11	13	15	16	17	19
$\mathcal{T}(\epsilon)$	1.03	1.82	2.41	2.90	3.12	3.67	3.94	4.28	4.73
$\mathcal{E}_T(\mathcal{I}(\epsilon))$	0.081	0.007	9.2E–4	4.8E–5	4.9E–6	2.8E–7	5.2E–8	8.3E–9	1.4E–10

Table 2 **The Number of Iterations $\mathcal{I}(m)$ and Computation Time $\mathcal{T}(m)$ (Seconds) as a Function of m, the Number of Queues, Using Alg(FPE) with Fixed EPT $\epsilon = 10^{-5}$**

m	2	4	6	8	10	12	14	16	18	20
$\mathcal{I}(m)$	12	12	12	13	12	12	12	12	12	12
$\mathcal{T}(m)$	2.86	4.68	6.43	8.75	11.02	11.96	13.96	15.63	17.39	19.21
m	30	40	50	60	70	80	100	120	140	160
$\mathcal{I}(m)$	12	12	12	12	12	12	12	12	12	12
$\mathcal{T}(m)$	29.76	37.37	48.67	58.42	68.15	73.63	96.77	115.0	134.84	147.7

run). In Figure 1 the solid lines are the simulation estimations of single sample paths applied with fluid scaling, and the dashed lines are the fluid approximations.

When the scale of the queueing model is not large, i.e., when n is smaller, single sample paths of the queueing functions typically do not agree closely with the fluid functions because of stochastic fluctuations. However, the mean functions of these processes can be well approximated, as shown in Section G.1 of the online supplement, Figure 8, for the case $n = 50$. In this example, the two queues do not become OL (UL) at the same time because of the phase difference of the external arrival rates (i.e., $\phi_1 = 0$, $\phi_2 = -3$). We also consider different phases ϕ_i in another example in Section G.1 of the online supplement.

All three algorithms were run on this example; the resulting identical performance functions confirm all of the algorithms. For this small FQNet example, the most important characteristic is ease of implementation, for which Alg(ODE) from §4 tends to be easiest, whereas Alg(FPE, GI) from §5 is hardest. For all examples, Alg(FPE, GI) tends to have the longest run time, as expected because it involves an FPE for each queue as well as an FPE for the TARs. For two-queue examples like the one just considered, the running time of Alg(FPE, GI) tends to be twice as long as that of Alg(ODE).

6.3. A Network with Many Queues

We next evaluate the performance of algorithms Alg(FPE) and Alg(ODE) as a function of the number of queues m. To do so, we consider a simple idealized network with m queues. Each queue i has a time-varying arrival rate as in (44), exponential service and patience times with rates μ_i and θ_i, constant staffing level s_i, and constant routing probabilities $P_{i,j}$, where

$$a_i = 0.5, \quad b_i = ia_i/m, \quad \phi_i = \pi(1.5 - i/m), \quad \theta_i = 0.5,$$
$$c_i = s_i = \mu_i = 1, \quad P_{i,j} = 1/2m, \quad 1 \le i \le m,\ 1 \le j \le m.$$

Figure 13 in Section G.2 of the online supplement shows plots of the performance functions for $m = 10$.

Table 2 shows the number of iterations $\mathcal{I}(m)$ and computation time $\mathcal{T}(m)$ in seconds as a function of the number of queues m, $2 \le m \le 160$, using algorithm Alg(FPE) with fixed EPT $\epsilon = 10^{-5}$. In this example we observe that (i) the number of iterations $\mathcal{I}(m)$ does not grow with the number of queues m, and (ii) the computation time $\mathcal{T}(m)$ grows linearly in m.

We also analyzed the performance of this same model using Alg(ODE). Table 3 shows the computation times $\mathcal{T}(m)$ as a function of m. Because we used the ODE solvers ode23 and ode45, which are $O(m)$ algorithms, the running time for Alg(ODE) becomes $O(m^2\mathcal{S})$. Figure 2 dramatically shows the difference in the algorithm performance.

We conclude this section with some general observations comparing the performance of the two algorithms Alg(FPE) and Alg(ODE). For small m (e.g., $2 \le m \le 8$) and small ϵ (e.g., $\epsilon < 10^{-5}$), Alg(ODE) runs faster than Alg(FPE); for big m and medium ϵ, Alg(FPE) runs faster than Alg(ODE). Of course, the complexity of Alg(ODE) depends on the choice of the multidimensional ODE solver. The polynomial growth in m as shown in Table 3 is attributed to the specific numerical scheme (such as Runge–Kutta–Fehlberg) of the ODE solver.

6.4. A $(G_t/LN/s+E_2)^2/M$ Non-Markovian Example

We now consider an example with a nonexponential service-time distribution for which only the final algorithm Alg(FPE, GI) introduced in §5 applies. To illustrate this example, we consider the $(G_t/LN/s+E_2)^2/M$ model with lognormal service distributions at each queue (the LN) and Erlang-2 patience distributions at each queue (the E_2). Specifically, we let the service time at station i be $S_i \equiv e^{Z_i}$, where Z_i is a normal random variable with mean $\hat{\mu}_i$ and variance $\hat{\sigma}_i^2$, i.e.,

Table 3 **The Computation Time $\mathcal{T}(m)$ (Seconds) as a Function of the Number of Queues m Using Alg(ODE)**

m	2	4	6	8	10	12	14	16	18	20
$\mathcal{T}(m)$	2.77	3.67	6.16	8.92	12.03	15.46	20.35	25.95	31.30	37.37
m	30	40	50	60	70	80	100	120	140	160
$\mathcal{T}(m)$	64.72	107.05	132.65	178.7	227.64	312.61	411.09	567.15	765.55	1,013.1

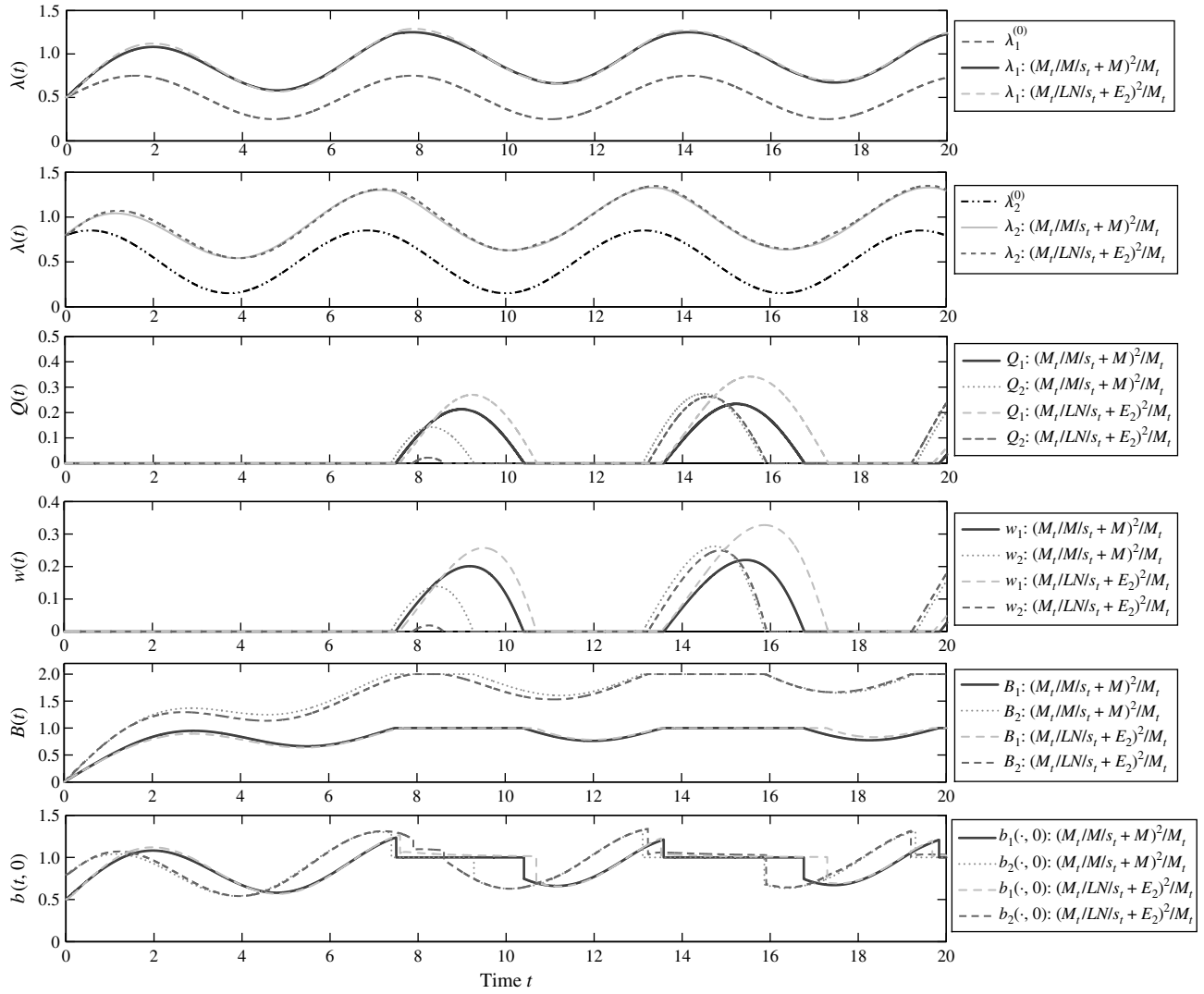

Figure 3 Computing the Fluid Performance Functions for the $(M_t/LN/s_t+E_2)^2/M_t$ Network Fluid Model

$Z_i \sim N(\hat{\mu}_i, \hat{\sigma}_i^2)$, $i = 1, 2$. The service *probability density function* (pdf) is

$$g_i(x) = \frac{1}{x\hat{\sigma}_i\sqrt{2\pi}} e^{-(\log x - \hat{\mu}_i)^2/(2\hat{\sigma}_i^2)}, \quad x \geq 0,\ i = 1, 2.$$

For $i = 1, 2$, the mean service times and the variances are

$$\mu_i^{-1} \equiv E[S_i] = e^{\hat{\mu}_i + (1/2)\hat{\sigma}_i^2} \quad \text{and}$$
$$\sigma_i^2 \equiv \mathrm{Var}(S_i) = (e^{\hat{\sigma}_i^2} - 1)e^{2\hat{\mu}_i + \hat{\sigma}_i^2}.$$

The *LN* assumption is representative because Brown et al. (2005) showed that service times in call centers follow *LN* distributions.

We let the patience distribution be Erlang-2 (E_2) with pdf

$$f_i(x) = 4\theta_i^2 x e^{-2\theta_i x}, \quad x \geq 0.$$

Letting A_i be a generic patience time of a customer at queue i, we have $E[A_i] = 1/\theta_i$, $i = 1, 2$. The E_2 distribution has a squared coefficient of variation $c^2 \equiv \mathrm{Var}(X)/E[X]^2 = 1/2$. We choose $\hat{\mu}_1 = -0.549$, $\hat{\sigma}_1 = 1.048$, $\hat{\mu}_2 = 0.144$, and $\hat{\sigma}_2 = 1.048$ such that $\mu_1 = 1$, $\mu_2 = 0.5$, $\sigma_1^2 = 2$, and $\sigma_2^2 = 8$. Thus, we have $c^2 = 2$ for the service distributions. We let $\theta_1 = 0.5$, $\theta_2 = 0.3$. In this way both the service rates (μ_1 and μ_2) and the patience rates (θ_1 and θ_2) remain the same as in the example in §6.2. For comparison, we let the external arrival rate $\lambda^{(0)}$ be sinusoidal as in (44) and the Markovian routing matrix $\mathbf{P}$ be constant with the same parameters as in §6.2. We also let the system be initially empty.

Figure 3 and Figure 14 in Section G.3 of the online supplement show plots of the standard performance functions and compares them to simulation experiments in the two cases $n = 4{,}000$ and $n = 50$. These two figures are analogs of Figures 7 and 8 in the online supplement. As before, for $n = 4{,}000$ the fluid

performance agrees with individual sample paths of the SQNet; for $n = 50$ the fluid performance agrees with the mean values of the time-varying stochastic SQNet performance. In Figure 3, we compare the fluid functions of the two-queue Markovian model (the solid line for queue 1 and dotted line for queue 2) and those of the non-Markovian $(M_t/LN/s + E_2)^2/M$ model (the dashed line for queue 1 and dashed and dotted line for queue 2). As indicated earlier, these two models have the same model parameters, including the service and patience rates μ and θ, except for the service and patience distributions.

In addition to showing that the new algorithm Alg(FPE, GI) is effective, Figure 3 shows that the service and patience distributions beyond their means play an important role in the time-dependent performance of the fluid network with time-varying model parameters. For the stationary $G/GI/s + GI$ fluid queue, Whitt (2006) showed that the patience distribution beyond its mean plays an important role, whereas the service-time distribution does not. In Liu and Whitt (2012a) we showed that the service-time distribution beyond its mean is also important in the time-dependent behavior.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ijoc.1120.0547.

Acknowledgments

The second author was supported by National Science Foundation [Grant CMMI 1066372].

References

Brown L, Gans N, Mandelbaum A, Sakov A, Shen H, Zeltyn S, Zhao L (2005) Statistical analysis of a telephone call center: A queueing-science perspective. *J. Amer. Statist. Assoc.* 100(469): 36–50.

Green LV, Kolesar PJ, Whitt W (2007) Coping with time-varying demand when setting staffing requirements for a service system. *Production Oper. Management* 16(1):13–39.

Kang W, Pang G (2011) Computation and properties of fluid models for time-varying many-server queues with abandonment. Working paper, Pennsylvania State University, University Park.

Liu Y, Whitt W (2011a) A network of time-varying many-server fluid queues with customer abandonment. *Oper. Res.* 59(4): 835–846.

Liu Y, Whitt W (2011b) Large-time asymptotics for the $G_t/M_t/s_t + GI_t$ many-server fluid queue with abandonment. *Queueing Systems* 67(2):145–182.

Liu Y, Whitt W (2012a) The $G_t/GI/s_t + GI$ many-server fluid queue. *Queueing Systems* 71(4):405–444.

Liu Y, Whitt W (2012b) A many-server fluid limit for the $G_t/GI/s_t + GI$ queueing model experiencing periods of overloading. *Oper. Res. Lett.* 40(5):307–312.

Liu Y, Whitt W (2013) Many-server heavy-traffic limits for queues with time-varying parameters. *Ann. Appl. Probab.* Forthcoming.

Mandelbaum A, Massey WA, Reiman MI (1998) Strong approximations for Markovian service networks. *Queueing Systems* 30(1–2): 149–201.

Mandelbaum A, Massey WA, Reiman MI, Rider B (1999a) Time varying multiserver queues with abandonments and retrials. Key P, Smith D, eds. *Proc. 16th Internat. Teletraffic Congress, Edinburgh, UK,* 355–364.

Mandelbaum A, Massey WA, Reiman MI, Stolyar A (1999b) Waiting time asymptotics for time varying multiserver queues with abandonment and retrials. *Proc. of 37th Annual Allerton Conf. Comm., Control Comput., Allerton, IL*, 1095–1104.

Massey WA, Whitt W (1993) Networks of infinite-server queues with nonstationary Poisson input. *Queueing Systems* 13(1–3): 183–250.

Nelson BL, Taaffe MR (2004a) The $Ph_t/Ph_t/\infty$ queueing system: Part I—The single node. *INFORMS J. Comput.* 16(3):266–274.

Nelson BL, Taaffe MR (2004b) The $[Ph_t/Ph_t/\infty]^k$ queueing system: Part II—The multiclass network. *INFORMS J. Comput.* 16(3):275–283.

Newell GF (1982) *Applications of Queueing Theory*, 2nd ed. (Chapman and Hall, London).

Thomas JW (1995) *Numerical Partial Differential Equations: Finite Difference Methods* (Springer, New York).

Whitt W (2006) Fluid models for multiserver queues with abandonments. *Oper. Res.* 54(1):37–54.