Waiting time and queue length analysis of Markov-modulated fluid priority queues

Gábor Horváth

Received: 24 May 2018 / Revised: 31 October 2019
© The Author(s) 2020

Abstract
This paper considers a multi-type fluid queue with priority service. The input fluid rates are modulated by a Markov chain, which is common for all fluid types. The service rate of the queue is constant. Various performance measures are derived, including the Laplace–Stieltjes transform and the moments of the stationary waiting time of the fluid drops and the queue length distributions. An Erlangization-based numerical method is also provided to approximate the waiting time and the queue length distributions up to arbitrary precision. All performance measures are formulated as reward accumulation problems during busy periods of simple Markovian fluid flow models, for which efficient matrix-analytic solutions are provided, enabling us to solve large models with several hundred states.

Keywords Fluid queue · Priority queue · Workload process · Matrix-analytic method

Mathematics Subject Classification 60K25 · 68M20 · 90B22 · 60J25 · 65C40

1 Introduction

Multi-class queueing systems have been used for a long time to analyze systems where different kinds of jobs exist. Among the multi-type service policies, priority-based decisions are the most popular choices. In such systems, a high priority is assigned to the most important jobs, and the server always picks the highest priority job present in the system. Jobs belonging to the same class are served in order of arrival. This

This work is supported by the OTKA K-123914 Project and by the ÚNKP-17-4-III New National Excellence Program of the Ministry of Human Capacities.

Gábor Horváth
ghorvath@hit.bme.hu

1 Department of Networked Systems and Services, MTA-BME Information Systems Research Group, Budapest University of Technology and Economics, Magyar Tudósok Körútja 2, Budapest 1117, Hungary

Published online: 09 March 2020
conceptual simplicity makes priority systems easy to implement both in hardware and in software.

Queueing systems with priority service have been studied for a very long time. The first results for priority queues with exponentially distributed inter-arrival and service times have been considered in [1] more than 50 years ago. Nowadays, priority systems can be analyzed efficiently with complex arrival processes (like marked Markovian arrival processes) and more general service time distributions (see [2] or [3]).

There are queueing applications where the demands are easier to model by continuous fluid flows rather than discrete jobs. In these systems, the jobs are considered to be infinitesimally small, like fluid drops, and the queue storing the jobs is a fluid storage. The analysis of continuous (fluid based) queues has a long history as well; however, for some reason, the results for continuous queues have always been published years after their discrete counterparts. For instance, the matrix-analytic methods enabling the numerically efficient analysis of many discrete-state (non-fluid) queues have been adapted to continuous queues only in 1999 (see [4]).

Not many results exist for fluid queues with multi-type jobs. Among the existing results, the fluid queues with priority service have been studied the most, due to their practical relevance in the performance evaluation of various telecommunication systems. The authors in [5] and [6] assume simple on–off modulated rate for the high priority demands and constant fluid rate for the low priority demands, and obtain the Laplace–Stieltjes transform (LST) of the joint queue length distribution for this two-class system. The procedure proposed in [7] assumes simple on–off sources with exponentially distributed on and off periods for both the high and the low priority traffic. The novelty of that approach is that the authors could avoid the characterization of the joint queue length distribution. The idea is to create a simplified process to obtain the marginal distribution of the buffer content. In the simplified low priority queue length process, however, there are non-Markovian periods, representing the periods when the high priority class is busy. In the solution proposed in this paper, a somewhat similar reduction is introduced, too, but the reduction results in a completely Markovian system. The assumptions on the input process in the aforementioned papers, however, can be too restrictive for practical applications.

In this paper, a more general system is considered, where the fluid input rates of the arrival process are modulated by an arbitrary Markov chain. The same model has been studied in [8], where the partial differential equations for the joint distribution of the queue lengths are derived. However, these differential equations, especially the boundary functions, are difficult to solve. [8] is able to provide the mean, the variance and the covariance of the queue lengths only in case of two-state Markov chains. In [9], the LST of the stationary joint distribution of the buffer contents is obtained in a closed form based on a spectral decomposition approach, from which the tail distributions and the queue length moments are derived. Using Little’s law, the mean waiting times are also presented. [10] follows a different approach to obtain the LST of the joint distribution, based on the analysis of the idle and busy periods of the high priority queue.

The (matrix-analytic) approach presented in this paper is based on the workload process analysis, just like in [3] for the discrete case. While the main steps of the analysis are the same, adapting the method of [3] to continuous queues is not straightforward at all. The specialties of continuous systems require different solutions in many steps.
of the procedure. Additionally, several ingredients of the procedure, which have been available for discrete systems for a long time (like the relation between the queue length distributions at departure instants and at random points in time) are not yet available for the continuous case.

The drawback of the presented approach is that the joint behavior of the queues cannot be characterized this way. (Note that the characterization of the joint queue length was the starting point of most related past papers.)

The advantage of the approach presented in this paper is that it can provide the LST and the moments for both the individual queue lengths and the waiting times of fluid drops in a nice and elegant form. (Note that the waiting time of fluid drops in a fluid priority queue has never been considered in the past, apart from the expected value.) Furthermore, numerical methods for the waiting time and the queue length distributions based on Erlangization are also provided, which are created with probabilistic interpretations, and have better numerical behavior than a generic inverse Laplace transform algorithm.

It is particularly important for us to develop algorithms that behave well numerically, even for large models. All performance measures are formulated as reward accumulation problems during busy periods of simple Markovian fluid flow models, for which matrix-analytic solutions are provided. The computation bottlenecks are the solutions of Riccati- and Sylvester-type equations, for which efficient implementations exist. As a result, the presented procedure can solve large models up to many hundreds of phases, while the past solutions mentioned above are less tractable, they cannot go above a couple of tens of states (except for [9], where an efficient solution is provided if the background Markov chain is constructed by a Kronecker composition).

The rest of the paper is organized as follows: Section 2 describes the system, introduces the notation and lists the main steps of the computation of the performance measures. Markov-modulated fluid flow models are considered in Sect. 3. The first part of this section gives an overview of known results. In the second part of this section, the reward accumulated during the busy period is characterized. The waiting time of the fluid drops is derived in Sect. 4. The queue length distribution of a given fluid class is studied in Sect. 5. Finally, some numerical examples conclude the paper in Sect. 6.

2 The Markov-modulated fluid priority queue and the concept of the solution

2.1 The description of the system

In the system considered in the paper, $K$ different (fluid) job types are distinguished, where class 1 has the lowest, and class $K$ the highest priority. The priority of the fluid can be interpreted as “weight,” and can correspond to the importance or the urgency of the jobs.

The rates at which the fluid belonging to various job types enters the system are modulated by a continuous-time Markov chain (CTMC, also referred to as the background process) with generator denoted by $Q$. (The irreducibility of the background process is assumed throughout the paper.) The stationary probability vector of the
CTMC is \( \pi \); hence, \( \pi Q = 0, \pi \mathbb{1} = 1 \) hold. (\( \mathbb{1} \) denotes the column vector of ones of appropriate size.) The rate at which class \( k \) fluid flows into the queue in state \( i \) is \( r_i^{(k)} \geq 0 \). The matrix \( R^{(k)} \) is a diagonal matrix composed of the class \( k \) incoming fluid rates; \( R^{(k)} = \text{diag}(r_i^{(k)}) \). For simplicity, we denote the input rates of classes having priority equal to or higher than \( k \) by \( r_i^{(k+)} = \sum_{\ell=k}^{K} r_i^{(\ell)} \), and the corresponding matrix by \( R^{(k+)} = \text{diag}(r_i^{(k+)} \). The rate at which fluid can leave the queue is denoted by \( d \).

Using the notation introduced above, the mean arrival rate of class \( k \) fluid can be obtained by \( \lambda^{(k)} = \pi R^{(k)} \mathbb{1} \), and the total input rate is \( \lambda = \sum_{k=1}^{K} \lambda^{(k)} \). The system is assumed to be stable; hence, \( \lambda < d \).

In those states of the system where the fluid arrival rate is greater than the service rate available, the fluid is directed into a buffer. The buffer space is assumed to be infinite. Due to the priority service, lower priority fluid cannot be drained when any of the higher priority queues contain fluid. When the higher priority queues are empty and the input fluid rates of the higher priority classes are less than \( d \), the lower priority queues are drained with the remaining service capacity. (Hence, the system is work conserving.)

### 2.2 Concept of the solution

In this paper, all performance measures will be derived from the analysis of the workload process. The workload, also called the unfinished work, or the virtual waiting time, is the amount of work present in the system, not in fluid level, but in time [11]. The fluid levels of different priorities are correlated; their joint analysis is rather complex. The workload process, however, makes it possible to obtain several performance measures from a Markov chain having only a single continuous variable, which is much more tractable numerically.

The solution is based on the “tagged customer” approach. When a class \( k \) fluid drop arrives into the queue, it finds a certain amount of class \( k \) workload in the system. This amount of workload has to be served before the tagged fluid drop can leave the system. The workload of lower priority classes can be neglected. While the tagged fluid drop waits for its service in the queue, further class \( (k+1) \) fluid can arrive, which has to be served before the tagged one. After a certain amount of time, the tagged fluid drop reaches the head of the queue and can leave the system.

- The waiting time of the tagged fluid drop is the workload found in the system at arrival, increased by the higher priority workload brought to the system while waiting.
- The class \( k \) queue length at the fluid drop departure instant is the amount of class \( k \) fluid arriving while the tagged fluid drop waits in the system.

It is shown in Sect. 4.1 that the workload process of a fluid priority queue can be characterized by a Markovian fluid flow model. In Sect. 4.2, we are going to show that the time until the workload ahead of the tagged fluid drop is processed by the system is equal to the busy period duration of a special Markovian fluid flow model defined appropriately. Finally, the queue length analysis problem is translated to the analysis...
of the amount of reward accumulated over the busy period of this special Markovian fluid flow model.

Since each step of the solution relies on Markovian fluid flow models, the following section gives an overview of their stationary distribution and the behavior of the busy period.

3 Markovian fluid flow models

In this section, we recap the definition of the “standard” Markovian fluid flow models (without priorities) and those properties that will be necessary for the performance analysis of the fluid priority queue. The main purpose of this section is to introduce the notation and to review the related matrix-analytic solution methodology. The second part of this section presents some new contributions as well, related to the busy period analysis.

Markovian fluid flow models (or, in short, fluid models) consist of a continuous-time background Markov chain and a fluid storage (buffer). The rate at which fluid flows into or out from the buffer depends on the state of the background Markov chain. The buffer has a lower boundary (i.e., the minimal fluid level is 0), and its capacity is considered to be infinite.

Formally, fluid models are two-dimensional Markov processes \( \{A(t), J(t), t > 0\} \), where \( A(t) \) is the fluid level and \( J(t) \) is the state of a CTMC (the background process) at time \( t \).

The Markov chain is assumed to be irreducible with state space \( S \); the number of states is \( N = |S| \). Its generator matrix is denoted by \( F = [f_{ij}, i, j \in S] \); the stationary distribution vector is \( \vartheta \), which satisfies \( \vartheta F = 0, \vartheta 1 = 1 \). A fluid rate is associated with each state of the background process, \( c_i, i \in S \), that determines the rate at which the level of the fluid buffer changes.

Since the analysis of the fluid priority queue involves the solution of various fluid flow models, we use different notation to make it easier to distinguish between the two systems. The generator matrix and the fluid input rates of the fluid priority queue are characterized by the matrices \( Q \) and \( R \), while the generator and the fluid rates of fluid flow models are denoted by \( F \) and \( C \) throughout the paper.

The evolution of the fluid level \( A(t) \) can be described by

\[
\frac{d}{dt} A(t) = \begin{cases} c J(t), & \text{if } A(t) > 0, \\ \max\{0, c J(t)\}, & \text{if } A(t) = 0. \end{cases}
\]

The size \( N \) row vector \( \pi(x) \) denotes the stationary density of the fluid level, whose \( i \)th entry is defined by \( \pi_i(x) = \lim_{t \to \infty} \frac{d}{dx} P(A(t) < x, J(t) = i) \). At level 0, probability mass can accumulate as well. The \( i \)th entry of the row vector \( p \) represents the probability mass at level 0, \( p_i = \lim_{t \to \infty} P(A(t) = 0, J(t) = i) \). The vector \( \pi(x) \) is the solution of the matrix differential equation \([12]\)

\[
\frac{d}{dt} \pi(x) C = \pi(x) F,
\]
where $C$ is a diagonal matrix composed of the fluid rates $c_i$. The boundary equations providing $\pi(0)$ and $p$ are

$$\pi(0)C = p F,$$

$$p_i = 0, \quad \forall i : c_i > 0. \quad (3)$$

### 3.1 The stationary distribution of the fluid level

Several numerical methods are available to solve (2) and (3), including the eigenvalue decomposition-based method [12], the Schur decomposition-based method [13] and the matrix-analytic method [4]. The subsequent sections of this paper rely heavily on the matrix-analytic solution of fluid models; thus, the rest of the section provides a short summary of this approach.

For the matrix-analytic method, the state space of the Markov chain has to be partitioned into three sets, $S = S_+ \cup S_- \cup S_0$, according to the sign of the fluid rates, i.e., $S_+ = \{i \in S, c_i > 0\}$, $S_- = \{i \in S, c_i < 0\}$ and $S_0 = \{i \in S, c_i = 0\}$. From now on, it is assumed that the matrices $F$ and $C$ are partitioned, and thus,

$$F = \begin{bmatrix} F_{++} & F_{+-} & F_{+0} \\ F_{-+} & F_{-} & F_{-0} \\ F_{0+} & F_{0-} & F_{00} \end{bmatrix}, \quad C = \begin{bmatrix} C_+ \\ C_- \\ 0 \end{bmatrix}, \quad (4)$$

where all diagonal entries of $C_+ = \text{diag}(c_i)_{i \in S_+}$ are positive and all diagonal entries of $C_- = \text{diag}(c_i)_{i \in S_-}$ are negative. Furthermore, let us introduce the matrix $F^\bullet$ as the generator of $\mathcal{J}(t)$ restricted to nonzero states, and hence,

$$F^\bullet = \begin{bmatrix} F_{++}^\bullet & F_{+-}^\bullet \\ F_{-+}^\bullet & F_{--}^\bullet \end{bmatrix} = \begin{bmatrix} F_{++} & F_{+-} \\ F_{-+} & F_{--} \end{bmatrix} + \begin{bmatrix} F_{+0} \\ F_{-0} \end{bmatrix} (-F_{00})^{-1} \begin{bmatrix} F_{0+} & F_{0-} \end{bmatrix}. \quad (5)$$

The stationary solution of the differential equation (2) and (3) has a matrix-exponential form, as proven in [4]. If the drift of the queue is negative, $\vartheta C 1_1 < 0$, the density function of the stationary fluid level can be expressed as:

$$\pi(x) = \kappa e^{Kx} \begin{bmatrix} I \Psi \begin{bmatrix} C_+ \\ \end{bmatrix} \begin{bmatrix} C_- \end{bmatrix} \end{bmatrix}^{-1} \begin{bmatrix} I 0 & \begin{bmatrix} F_{+0} & (-F_{00})^{-1} \end{bmatrix} 0 & \begin{bmatrix} F_{-0} & (-F_{00})^{-1} \end{bmatrix} \end{bmatrix}, \quad (6)$$

where the matrix $\Psi$ is the minimal nonnegative solution of the non-symmetric algebraic Riccati equation (NARE)

$$\Psi|C_-|^{-1}F_{-+}^\bullet + \Psi|C_-|^{-1}F_{-}^\bullet + C_+^{-1}F_{++}^\bullet + C_+^{-1}F_{+-}^\bullet = 0. \quad (7)$$

and the matrix $K$ is obtained by

$$K = C_+^{-1}F_{++}^\bullet + \Psi|C_-|^{-1}F_{+-}^\bullet. \quad (8)$$
The probability mass at level zero is \( p = [0 \ p_- \ p_0] \). The vectors \( \kappa, p_- \) and \( p_0 \) are the solutions of the linear equations

\[
\begin{bmatrix}
\kappa & p_- & p_0
\end{bmatrix}
\begin{bmatrix}
-A & 0 & 0 \\
F_{-+} & F_{--} & F_{-0} \\
F_{0+} & F_{0-} & F_{00}
\end{bmatrix} = 0,
\]

(9)

with the normalization condition

\[
\begin{bmatrix}
\kappa & p_- & p_0
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 1 \\
\end{bmatrix} = 1.
\]

(10)

### 3.2 Reward accumulation during the busy period

The duration of the busy period of fluid flow models, \( \tau = \inf(t > 0 : A(t) = 0) \), has been studied extensively in the literature: in [14] the Laplace–Stieltjes transform (LST) of its distribution is provided, and in [15] an Erlangization-based algorithm is described to approximate its distribution in the time domain. Both of these papers are based on matrix-analytic methods. (We note that the busy period of Markovian fluid flows can be analyzed with other methodologies as well, like with the spectral decomposition method, as in [16].)

In this section, we consider a more general problem. We assign a reward rate to each state of the background process and characterize the accumulated reward over the busy period of the fluid model. It turns out that the main performance measures of the fluid priority queue can be related to the accumulated reward over the busy period.

The introduction of reward accumulation makes the system three-dimensional, \( \{A(t), B(t), J(t), t \geq 0\} \), where \( B(t) \) represents the reward accumulated up to time \( t \). The reward accumulation is linear; thus, \( B(t) = \int_0^t dJ(u)du \), where \( d_i \) denotes the reward rate associated with state \( i \). Let us also introduce the diagonal matrix of reward rates \( D = \text{diag} \langle d_i \rangle \). The joint distribution of the accumulated reward and the state transitions over the busy period is given by the matrix \( \Phi(y) \), defined as

\[
[\Phi(y)]_{i,j} = P(B(\tau) < y, J(\tau) = j | A(0) = 0, B(0) = 0, J(0) = i),
\]

(11)

for \( i \in S_+, j \in S_- \). Observe that in the special case of \( D = I \), the matrix \( \Phi(y) \) equals the distribution of the busy period duration starting with zero initial fluid level.

The following theorem, from [17], provides the LST of \( \Phi(y) \), defined by \( \Phi^*(v) = \int_0^\infty e^{-vy} d\Phi(y) \).

**Theorem 1** [17, Theorem 3] The matrix \( \Phi^*(v) \), describing the LST of the accumulated reward over a busy period starting from an empty buffer, satisfies the NARE

\[
0 = F^*_{++}(v)\Phi^*(v) + \Phi^*(v)F^*_{--}(v) + \Phi^*(v)F^*_{0+}(v)\Phi^*(v) + F^*_{+-}(v),
\]

(12)
where the matrices $F_{++}^*(v), F_{--}^*(v), F_{-+}^*(v)$ and $F_{+-}^*(v)$ are defined by

\[
\begin{align*}
F_{++}^*(v) &= C_+^{-1}(F_{++} - vD_+ + F_{0+}H_{00}^*(v)F_{0+}), \\
F_{--}^*(v) &= -C_-^{-1}(F_{--} - vD_- + F_{-0}H_{00}^*(v)F_{0-}), \\
F_{-+}^*(v) &= -C_-^{-1}(F_{-+} + F_{-0}H_{00}^*(v)F_{0+}), \\
F_{+-}^*(v) &= C_+^{-1}(F_{+-} + F_{0+}H_{00}^*(v)F_{0-}).
\end{align*}
\] (13)

with $H_{00}^*(v) = (vD_0 - F_{00})^{-1}$ being the LST of the reward accumulated in the zero states.

To obtain the moments of the performance measures of the fluid priority queue, the derivatives of the matrix $\Phi^*(v)$ will be required at $v \to 0$. Let us introduce the matrices $F_{ij}^{(m)}$ defined by $F_{ij}^{(m)} = \frac{d^m}{dv^m} F_{ij}^*(v)|_{v \to 0}$ for $i, j \in \{+, -\}$. The next theorem provides the derivatives of $\Phi^*(v)$.

**Corollary 1** The $m$th derivative of $\Phi^*(v)$ at $v \to 0$, denoted by $\Phi^{(m)}(v) = \frac{d^m}{dv^m} \Phi^*(v)|_{v \to 0}$, is obtained recursively by solving the Sylvester equations

\[
(F_{++}^{(0)} + \Phi F_{-+}^{(0)} \Phi^{(m)} + \Phi^{(m)} F_{-+}^{(0)} + F_{++}^{(m)} - \Phi F_{+-}^{(m)} \Phi + U_{m-1}) = 0
\]

for $m > 0$, and for $m = 0$ we have that $\Phi^{(0)} = \Phi = \Psi$. The matrix $U_{m-1}$ depends only on the derivatives of $\Phi^*(v)$ up to order $m - 1$:

\[
U_{m-1} = \sum_{i=0}^{m-1} \binom{m}{i} (F_{++}^{(m-i)} + \Phi F_{--}^{(m-i)}) \Phi^{(i)} + \sum_{\ell=0}^{m-1} \binom{m}{\ell} \Phi^{(\ell)} (F_{--}^{(m-\ell)} + F_{-+}^{(m-\ell)}) \Phi^{(m-i) \Phi^{(i)}} + \sum_{\ell=1}^{m-1} \sum_{i=1}^{m-\ell} \binom{m}{\ell} (m - \ell) \Phi^{(\ell)} F_{-+}^{(m-\ell-i)} \Phi^{(i)}.
\]

**Proof** Taking the $m$th derivative of (12) and letting $v \to 0$ leads to

\[
\begin{align*}
\sum_{i=0}^{m} \binom{m}{i} F_{++}^{(m-i)} \Phi^{(i)} + \sum_{i=0}^{m} \binom{m}{i} \Phi^{(i)} F_{--}^{(m-i)} + F_{-+}^{(m)} + \sum_{\ell=0}^{m-\ell} \sum_{i=0}^{m-\ell} \binom{m}{\ell} (m - \ell) \Phi^{(\ell)} F_{-+}^{(m-\ell-i)} \Phi^{(i)} &= 0,
\end{align*}
\] (14)

which, after some manipulations, provides the theorem. \qed

 Springer
Appendix A describes how to obtain the matrix derivatives $F^{(m)}_{ij}$, $i, j \in \{+, -\}$ in detail.

The rest of this section focuses on the distribution of the reward accumulated over the busy period. Unfortunately, there are no exact solutions available to obtain $\Phi(t)$. One can either rely on a generic numerical inverse Laplace transform (ILT) method to calculate it from $\Phi^*(v)$, or, one can apply the Erlangization method to approximate it. Most numerical ILT algorithms have several well-known drawbacks: they require complex and high precision arithmetic, have convergence problems and are prone to over- and undershooting, meaning that they can result in negative or higher than 1 values instead of valid probabilities. The Erlangization method can be viewed as a special kind of ILT method as well. Compared to generic ILT methods, Erlangization is slower, in the sense that it needs more evaluations of $\Phi^*(v)$ to achieve reasonable accuracy. On the other hand, it can be implemented using real, machine precision arithmetic and has no over- and undershooting problems; the result is always a valid probability. Additionally, Erlangization has an elegant stochastic interpretation, which can be exploited when tailored to specific problems. Below we present an Erlangization-based numerical method to approximate $\Phi(t)$, following the approach of [15]. The basic algorithm in [15] is extended in two aspects: first, that paper studies the duration of the busy period, while we need the accumulated reward in this paper; second, zero rates are excluded in [15], while we need them here.

According to the Erlangization algorithm, the order-$n$ approximation of the matrix $\Phi(t)$ is

$$\Phi_n(t) = \int_0^\infty f_{E(n,n/t)}(u) \cdot \Phi(u) \, du,$$

(15)

where $f_{E(n,n/t)}(u)$ is an order-$n$ Erlang density with rate parameter $\nu = n/t$. As $n$ tends to infinity, $f_{E(n,n/t)}(u)$ tends to a Dirac delta function $\delta(u - t)$; hence, $\Phi_n(t)$ tends to $\Phi(t)$. The key idea behind Erlangization is that, by probabilistic interpretation, the integral in (15) is often easier to compute than $\Phi(t)$ itself. In our case, the integral can be interpreted as the probability that the reward accumulated during the busy period is less than an Erlang$(n, \nu)$ variable.

**Theorem 2** The order-$n$ approximation of the joint distribution of the accumulated reward and the state transitions over the busy period is

$$\Phi_n(t) = \sum_{\ell=0}^{n-1} \tilde{\Psi}_\ell(v) |_{v=n/t},$$

(16)

where we have $\tilde{\Psi}_0(v) = \Phi^*(v)$, and for $\ell > 0$ the matrices $\tilde{\Psi}_\ell(v)$ are obtained recursively by the solutions of the Sylvester equations

$$0 = (F_{++}(v) + \tilde{\Psi}_0(v)F_{++}(v)) \tilde{\Psi}_\ell(v) + \tilde{\Psi}_\ell(v)(F_{--}(v) + F_{++}(v) \tilde{\Psi}_0(v))$$

$$+ \nu C^{-1}_+ D_+ \tilde{\Psi}_{\ell-1}(v) + \nu \tilde{\Psi}_{\ell-1}(v)(-C_-)^{-1}D_- + C^{-1}_+ F_{+0} Z_{\ell}(v) F_{0-}$$

$$- \tilde{\Psi}_0(v)(-C_-)^{-1}F_{-0} Z_{\ell}(v) F_{0-} + \tilde{\Psi}_0(v)$$

\(\circ\) Springer
\[
\ell - 1 \sum_{i=1}^{\ell - i} \Psi_i(v)(-C_-)^{-1} F_{-0} Z_{\ell - i - j}(v) F_{0+} \hat{\Psi}_j(v)
\]

\[
+ \sum_{i=1}^{\ell - 1} \Psi_i(v)(-C_-)^{-1} F_{-+} \hat{\Psi}_{-i}(v)
\]

\[
+ \sum_{i=0}^{\ell - 1} \Psi_i(v)(-C_-)^{-1} F_{-0} Z_{\ell - i}(v)(F_{0-} + F_{0+} \hat{\Psi}_0(v))
\]

\[
+ \sum_{i=0}^{\ell - 1} (C_+^{-1} F_{+0} + \Psi_0(v)(-C_-)^{-1} F_{0-}) Z_{\ell - i}(v) F_{0+} \hat{\Psi}_i(v),
\]

(17)

where the matrices \( F_{++}(v), F_{+-}(v), F_{-+}(v) \) and \( F_{+-}(v) \) are defined by (13) and the matrices \( Z_{\ell}(v) \) are defined by \( Z_{\ell}(v) = H_{00}^*(v)(vD_0^*H_{00}^*(v))^\ell \).

**Proof** The proof of the theorem is presented in Appendix B. \( \square \)

### 4 Waiting time analysis of the fluid priority queue

This section presents the LST, the moments and the distribution function of the waiting time of class \( k \) fluid drops, for \( k = 1, \ldots, K \).

#### 4.1 The distribution of the workload at fluid drop arrivals

As the first step of the analysis, the distribution of the total workload (also called backlog, unfinished work, or virtual waiting time in the literature) of classes \( \geq k \) is determined at class \( k \) fluid drop arrival instants. This is the amount of workload (and possibly more, due to further higher priority arrivals) that has to be accomplished before the tagged fluid drop can leave the system.

In non-fluid queues, where the jobs are discrete, the difference between the discrete-state queue length process and the continuous-state workload process is more obvious, but in the case of fluid queues both processes have continuous state space, and we will show that their dynamics are rather similar, too. Still, for the analysis of fluid priority queues, using the workload process is essential: it allows us to derive the performance measures from a process having only one continuous variable. Basing the analysis on the queue length process, we would need to solve a system with two continuous variables (the lengths of priority \( k \) and the priority \( (k + 1) \) queues—since they are correlated), which is much more involved.

Let us denote the workload of classes \( \geq k \) at time \( t \) by \( \mathcal{V}^{(k)}(t) \). It is important to note that the workload is measured in time, and not in fluid level. \( \mathcal{V}^{(k)}(t) \) is the time needed by the system to accomplish all the work and become idle, given that no class \( \geq k \) fluid enters the queue after time \( t \). \( \mathcal{V}^{(k)}(t) = 0 \) holds if and only if the class \( \geq k \) fluid level is zero, but in general the evolution of the workload process differs from the evolution of the fluid level process.
The joint density function of the workload and the state of the background process at time $t$ is denoted by the row vector $v^{(k+)}(t, x)$, whose elements are $v_i^{(k+)}(t, x) = \frac{d}{dx} P(Y_i^{(k+)}(t) \leq x, \mathcal{J}(t) = i)$. The joint probability that the workload of the system is 0 and the state of the background process is $i$ is stored by the row vector $\alpha_i^{(k+)}(t) = [\alpha_i^{(k+)}(t)]$, where $\alpha_i^{(k+)}(t) = P(Y_i^{(k+)}(t) = 0, \mathcal{J}(t) = i)$.

**Lemma 1** The vector $v^{(k+)}(t, x)$ is determined by the differential equation

$$\frac{\partial}{\partial t} v^{(k+)}(t, x) + \frac{\partial}{\partial x} v^{(k+)}(t, x) \left( \frac{1}{d} R^{(k+)} - I \right) = v^{(k+)}(t, x)Q,$$  \hspace{1cm} (18)

with the boundary condition

$$\frac{d}{dt} \alpha_i^{(k+)}(t) = \sum_{j: r_j^{(k+)} \leq d} \alpha_j^{(k+)}(t) q_{ji} - v_i^{(k+)}(t, 0) \left( \frac{r_i^{(k+)}}{d} - 1 \right), \text{ for } i : r_i^{(k+)} \leq d, \hspace{1cm} (19)$$

and $\alpha_i^{(k+)}(t) = 0$ for $i : r_i^{(k+)} > d$.

**Proof** For $x > 0$, the behavior of $v^{(k+)}(t, x)$ in an infinitesimally small $\Delta$ time interval is

$$v_i^{(k+)}(t, x) = v_i^{(k+)}(t - \Delta, x + \Delta - \Delta r_i^{(k+)} / d) (1 + q_{ii} \Delta) + \sum_{j \neq i} q_{ji} \Delta v_j^{(k+)}(t - \Theta(\Delta), x + \Theta(\Delta)) + o(\Delta), \hspace{1cm} (20)$$

where the first term corresponds to the case when there is no state transition in $(t - \Delta, t)$, and the second one when there is a state transition from $j$ to $i$. Over a $\Delta$ long interval, the amount of class $\geq k$ workload brought into the system is $\Delta r_i^{(k+)} / d$ when the background process is in state $i$, and the amount of workload processed by the server is $\Delta$. Functions $o(\Delta)$ and $\Theta(\Delta)$ are such that $\lim_{\Delta \to 0} o(\Delta) / \Delta = 0$ and $\lim_{\Delta \to 0} \Theta(\Delta) = 0$, respectively.

Rearranging the terms of (20) and dividing both sides by $\Delta$ leads to

$$\frac{v_i^{(k+)}(t, x) - v_i^{(k+)}(t - \Delta, x)}{\Delta} + \frac{v_i^{(k+)}(t - \Delta, x) - v_i^{(k+)}(t - \Delta, x + \Delta - \Delta r_i^{(k+)} / d)}{\Delta} = \sum_{j \in \mathcal{S}} q_{ji} v_j^{(k+)}(t - \Theta(\Delta), x + \Theta(\Delta)) + o(\Delta) / \Delta,$$

which provides (18) when $\Delta \to 0$.

The boundary probability $\alpha_i^{(k+)}(t)$ is clearly zero if $r_i^{(k+)} > d$, since in these states workload starts to accumulate. For states where $r_i^{(k+)} \leq d$ holds, the behavior in a small $\Delta$ interval is
\[
\alpha_i^{(k+)}(t) = \alpha_i^{(k+)}(t - \Delta)(1 + q_{ii} \Delta) + \sum_{j \neq i, r_j^{(k+)} < d} q_{ji} \Delta \alpha_j^{(k+)}(t - \Delta) + P(\mathcal{V}^{(k+)}(t - \Delta) \in [0, \Delta - \Delta t_i^{(k+)} / d], \mathcal{J}(t - \Delta) = i) + o(\Delta),
\]

thus, either the workload was 0 in \(t - \Delta\), or it was positive, but close enough to zero such that it became zero in \((t - \Delta, t)\). Rearranging the terms, dividing by \(\Delta\) and tending \(\Delta \to 0\) gives (19).

Lemma 1 has an important corollary: the workload process behaves like an ordinary Markovian fluid flow model with parameters

\[
\mathbf{F} = \mathbf{Q}, \quad \mathbf{C}^{(k+)} = \frac{1}{d} \mathbf{R}^{(k+)} - \mathbf{I},
\]

(21)

thus, its stationary density, \(v^{(k+)}(x) = \lim_{t \to \infty} v^{(k+)}(t, x)\), and probability mass at zero, \(\alpha^{(k+)} = \lim_{t \to \infty} \alpha^{(k+)}(t)\), are given by the matrix-exponential form

\[
v^{(k+)}(x) = \kappa^{(k+)} e^{\mathbf{K}^{(k+)} x} \mathbf{A}^{(k+)}, \quad \alpha^{(k+)} = p^{(k+)},
\]

(22)

where the vectors \(\kappa^{(k+)}, p^{(k+)}\) and the matrices \(\mathbf{K}^{(k+)}, \mathbf{A}^{(k+)}\) are given by (6)–(10) with parameters (21). Hence, the density of the workload and the probability mass at zero embedded at class \(k\) fluid drop arrival instants are

\[
\hat{v}^{(k)}(x) = \frac{1}{\lambda^{(k)}} \kappa^{(k+) e^{\mathbf{K}^{(k+) x} \mathbf{A}^{(k+)} \mathbf{R}^{(k)}}, \quad \hat{\alpha}^{(k)} = \frac{1}{\lambda^{(k)}} p^{(k+) \mathbf{R}^{(k)}}.
\]

(23)

4.2 Obtaining the properties of the waiting time

As described in Sect. 2, the waiting time of a fluid drop is equal to the workload found in the system at its arrival, increased by the higher priority workload brought to the system while waiting.

Figure 1 shows an example for the evolution of the remaining waiting time process of a class \(k\) fluid drop, \(\mathcal{W}^{(k)}(t, x)\), where the arriving fluid drop found \(x\) amount of class \(k\) workload in the system. In the time periods where the curve is drawn with a solid line, the background process is in a state where \(r_J^{(k+)} = 0\) and there is no class \((k + 1)+\) workload in the system either; hence, all the server capacity is spent on class \(k\), and the remaining waiting time decreases with slope \(-1\). In the intervals where the curve is dashed, class \((k + 1)+\) fluid is also present, and the service capacity is either fully taken by the higher priority classes, or, if their incoming rate is less than \(d\) and there is no high priority fluid accumulated in the buffer, it is shared between class \(k\) and the higher priority classes. While Fig. 1 depicts the remaining waiting time of a class \(k\) fluid drop, we can observe that it follows the same trajectory as the class \((k + 1)+\) workload process, \(\mathcal{W}^{((k+)+)}(t)\), given that \(\mathcal{W}^{((k+)+)}(0) = x\).

Let \(\mathcal{T}\) denote the random variable representing the waiting time of class \(k\) fluid drops. (For the sake of simplicity, we neglect the superscript \((k)\) for \(\mathcal{T}\) in this section.)
By conditioning on the amount of class $k+$ workload at a class $k$ fluid drop arrival, we have that

$$P(T < y) = \int_0^{\infty} \hat{v}^{(k)}(x) G^{(k+1)+}(y, x) \, dx,$$

(24)

with $G^{(k+1)+}(y, x)$ being the distribution of the busy period of the class $(k + 1)+$ workload starting at level $x$, defined by

$$G^{(k+1)+}(y, x) = P(\tau^{(k+1)+} < y, J(\tau) = j | \mathcal{Y}^{(k+1)+}(0) = x, J(0) = i),$$

where $\tau^{(k+1)+} = \inf(t > 0 : \mathcal{Y}^{(k+1)+}(t) = 0)$ is the first time point when the workload process reaches level 0.

To get rid of the integral, we apply a transformation that transforms the analysis problem of the class $k$ waiting time to the analysis problem of the accumulated reward of an appropriately constructed fluid flow model over a busy period. The main idea (introduced in [18]) is to start this fluid model at level 0 and add a phase whose role is to set the fluid level according to the initial workload at class $k$ fluid arrival, having density $\hat{\upsilon}^{(k)}(x)$ (Fig. 2). The main benefit of this transformation is that it makes it possible to use the results of Sect. 3.2, where the busy period with 0 initial level is characterized.
The generator of the background process, the fluid rates and the reward rates of this special fluid flow model are

\[
F = \begin{bmatrix} K^{(k+)} & A^{(k+)}R^{(k)}/\lambda^{(k)} \\ Q \end{bmatrix}, \quad C = \begin{bmatrix} I \\ 1/\beta R^{((k+)+)} - I \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{25}
\]

The first state group of the background process is responsible for the accumulation of the workload a class \( k \) fluid drop finds in the system upon its arrival. Observe that the probability density of the time spent here equals \( e^{K^{(k+)}t}A^{(k+)}R^{(k)}/\lambda^{(k)} \), which, when started from the vector \( \kappa^{(k+)} \), equals \( \hat{v}^{(k)}(t) \), according to (23). (Remember that \( \kappa^{(k+)} \) is the initial vector of the stationary solution of the workload process; see (23).) While staying in this state group the workload increases with rate 1 (hence the \( I \) matrix in the first block of \( C \)), and during this period no reward is accumulated, since the time spent here is not part of the waiting time. (There is a 0 in the corresponding block in the matrix \( D \).)

When the appropriate workload level is reached, a transition occurs to the second group of states. In this part, the fluid level in this special model represents the workload ahead of the tagged class \( k \) fluid drop. If class \( k \) has the highest priority in the system, then the workload ahead of the tagged fluid drop decreases linearly by rate 1. If there is higher priority work, class \((k+1)^+\) fluid flows in the system, then the workload brought by them increases the workload ahead of the tagged class \( k \) fluid drop. The rate at which the workload changes during this period is given by the diagonal elements of \( 1/\beta R^{((k+)+)} - I \). The reward rate is 1 in each state of the modulating CTMC; thus, the reward measures the time spent in this part of the state space. The time spent until the workload ahead of the tagged fluid drop reaches level 0 is identical to the waiting time.

Consequently, the waiting time is equal to the amount of reward accumulated during the busy period, starting from the first state group; thus, the initial phase distribution vector is

\[
\hat{\kappa}^{(k)} = \begin{bmatrix} \kappa^{(k+)} \\ 0 \end{bmatrix}. \tag{26}
\]

Based on the above reasoning, we can state the following corollary.

**Corollary 2** The LST of the distribution function of \( T \), \( f_T^*(s) \) and its \( m \)th moment, \( E(T^m) \), are expressed by

\[
f_T^*(s) = \hat{\kappa}^{(k)} \Phi^*(s) \mathbb{1}, \tag{27}
\]

\[
E(T^m) = (-1)^m \hat{\kappa}^{(k)} \Phi^{(m)} \mathbb{1}, \quad k > 0, \tag{28}
\]

and the order-\( n \) approximation of the distribution function, \( F_T^{(n)}(t) \), is

\[
F_T^{(n)}(t) = \hat{\kappa}^{(k)} \mathbb{1} + \hat{\kappa}^{(k)} \Phi_n(t) \mathbb{1}, \tag{29}
\]

where the matrices \( \Phi^*(s) \), \( \Phi^{(m)} \) and \( \Phi_n(t) \) are given by the reward analysis of the fluid model defined by the matrices (25), as detailed in Sect. 3.2.
In (29), the first term \( \hat{\alpha}^{(k)} \) reflects that the waiting time is zero if the workload is zero when the fluid drop arrives.

5 Queue length analysis of the fluid priority queue

To analyze the stationary queue length for each fluid class of the system, we first characterize the queue length at fluid drop departure instants. Then, we provide the relation between the queue length distribution at departures and the queue length distribution at random point in time.

5.1 Queue length distribution at fluid drop departures

The approach for characterizing the amount of class \( k \) fluid in the system at class \( k \) fluid drop departures is similar to the one presented in Sect. 4.2. A special Markov-modulated fluid model is constructed such that the reward accumulated during its busy period is equal to the queue length at departures. The matrices defining this special fluid model are

\[
F = \left[ \frac{K^{(k+)} A^{(k+)} R^{(k)} / \hat{\alpha}^{(k)}}{Q} \right], \\
C = \left[ \begin{array}{c}
1 \\
\frac{1}{\beta} R^{((k+)+)} - I
\end{array} \right], \\
D = \left[ \begin{array}{c}
0 \\
R^{(k)}
\end{array} \right].
\]

(30)

Observe that these matrices are similar to (25), and the interpretation is similar as well. The first state group sets the initial workload seen by a class \( k \) fluid drop arrival. The second state group follows the workload ahead of the fluid drop until it leaves the system. With the given reward rate matrix \( D \), however, the reward accumulated over the busy period measures the amount of class \( k \) fluid that arrives until the tagged fluid drop leaves; hence, it provides the class \( k \) queue length at the departure of the tagged class \( k \) fluid drop.

Corollary 3 Denoting the class \( k \) queue length at departures by \( X \), the LST of the distribution function by \( f_X^*(s) \), its \( m \)th moment by \( E(\mathcal{X}^m) \), and the order-\( n \) approximation of its distribution function by \( F_{\mathcal{X}}^{(n)}(x) \), we have that

\[
f_X^*(s) = \hat{\kappa}^{(k)} \Phi^*(s) \mathbb{I}, \quad (31)
\]

\[
E(\mathcal{X}^m) = (-1)^m \hat{\kappa}^{(k)} \Phi^{(m)} \mathbb{I}, \quad k > 0, \quad (32)
\]

\[
F_{\mathcal{X}}^{(n)}(x) = \hat{\alpha}^{(k)} \mathbb{I} + \hat{\kappa}^{(k)} \Phi_n(x) \mathbb{I}. \quad (33)
\]

The definition and the computation of the matrices \( \Phi^*(s) \), \( \Phi^{(m)} \) and \( \Phi_n(x) \) of the fluid model characterized by (30) are detailed in Sect. 3.2.

The first term in (33), \( \hat{\alpha}^{(k)} \), is the probability that the class \( k+ \) workload is zero at a class \( k \) fluid drop arrival, implying that the queue length is zero, too.
5.2 Queue length distribution at random point in time

In the case of non-fluid queues with discrete customers, the relation between the queue length distribution at departures and at random point in time is known [19]. For fluid queues, however, such a result is not available in the literature.

Let \( \mathcal{Y} \) denote the class \( k \) queue length at random point in time. The density function, the distribution function and the probability mass at level 0 are denoted by \( f_\mathcal{Y}(x) \), \( F_\mathcal{Y}(x) \) and \( p_\mathcal{Y} \), respectively. The following theorem establishes a relationship between the density and distribution functions of \( \mathcal{X} \) and \( \mathcal{Y} \). Before stating the theorem, we introduce the vector forms of these quantities that include the state of the background process as well, characterizing \( \lim_{t \to \infty} \{ \mathcal{Y}(t), \mathcal{J}(t) \} \). These row vector quantities are defined by \( [F_\mathcal{Y}(x)]_i = \lim_{t \to \infty} P(\mathcal{Y}(t) < x, \mathcal{J}(t) = i) \), \( f_\mathcal{Y}(x) = \frac{d}{dx} F_\mathcal{Y}(x) \), and \( [p_\mathcal{Y}]_i = \lim_{t \to \infty} P(\mathcal{Y}(t) = 0, \mathcal{J}(t) = i) \).

**Theorem 3** The density and distribution functions of \( \mathcal{X} \) and \( \mathcal{Y} \) are related by

\[
 f_\mathcal{Y}(u) R^{(k)} - F_\mathcal{Y}(u) Q = \lambda^{(k)} f_\mathcal{X}(u), \tag{34}
\]

and for the masses at level 0 we have

\[
 p_\mathcal{Y} R^{(k)} = \lambda^{(k)} p_\mathcal{X}. \tag{35}
\]

**Proof** The proof is inspired by [20] and is based on simple balance equations. Assume that the size of the fluid drops is \( \delta \). Expressing the rate at which the state \( \{ \mathcal{Y} \in [0, x), \mathcal{J}(t) = j \} \) is left (for \( x > 0 \)) we get

\[
 P(\mathcal{Y} \in [x - \delta, x), \mathcal{J} = j) \cdot r_j^{(k)}/\delta + P(\mathcal{Y} < x, \mathcal{J} = j) \cdot (-q_{jj}), \tag{36}
\]

where the first term is the rate of moving from \( \{ \mathcal{Y} \in [0, x), \mathcal{J}(t) = j \} \) to \( \{ \mathcal{Y} \geq x, \mathcal{J}(t) = j \} \) due to a fluid drop arrival, and the second term is the rate of leaving \( \{ \mathcal{Y} \in [0, x), \mathcal{J}(t) = j \} \) due to the state change of the background process.

For \( x > 0 \), the rate at which the system enters state \( \{ \mathcal{Y} \in [0, x), \mathcal{J} = j \} \) is

\[
 P(\mathcal{X} \in [x - \delta, x), \mathcal{J} = j) \cdot \lambda^{(k)}/\delta + \sum_{i \neq j} P(\mathcal{Y} < x, \mathcal{J} = i) \cdot q_{ij}, \tag{37}
\]

where \( P(\mathcal{X} \in [x - \delta, x), \mathcal{J} = j) \), in the first term, is the fraction of fluid drops that leave the system behind with fluid level \( < x \). Since \( \lambda^{(k)}/\delta \) is the fluid drop departure intensity, \( P(\mathcal{X} \in [x - \delta, x), \mathcal{J} = j) \cdot \lambda^{(k)}/\delta \) is the transition rate from \( \{ \mathcal{Y} \geq x, \mathcal{J} = j \} \) to \( \{ \mathcal{Y} \in [0, x), \mathcal{J} = j \} \) caused by fluid drop departures. The second term covers the state transitions triggered by the background process, from state \( \{ \mathcal{Y} \in [0, x), \mathcal{J} = i \} \) to \( \{ \mathcal{Y} \in [0, x), \mathcal{J} = j \} \).

Equating (36) and (37) and rearranging the terms lead to
\[
\frac{P(\mathcal{Y} \in [x-\delta, x), \mathcal{J} = j)}{\delta} r_j^{(k)} - \sum_{i \in S} P(\mathcal{Y} < x, \mathcal{J} = i) \cdot q_{ij} = \lambda^{(k)} P(X \in [x-\delta, x), \mathcal{J} = j)/\delta,
\]

which, letting \( \delta \to 0 \), exploiting that \( \lim_{\delta \to 0} P(\mathcal{Y} \in [x-\delta, x), \mathcal{J} = j)/\delta = [f^{(k)}_{\mathcal{Y}}(x)]_j \) and \( \lim_{\delta \to 0} P(X \in [x-\delta, x), \mathcal{J} = j)/\delta = [f^{(k)}_{\mathcal{X}}(x)]_j \), then switching to matrix notation, gives (34).

The relation between the probability masses of \( X \) and \( Y \) at level 0 can be obtained using similar arguments. The rate of leaving state \( \{ \mathcal{Y} = 0, \mathcal{J} = j \} \) is \( [p^{(k)}_Y]_j r_j^{(k)} / \delta + [p^{(k)}_Y]_j (-q_{ii}) \), while the rate of entering \( \{ \mathcal{Y} = 0, \mathcal{J} = j \} \) is \( \lambda^{(k)}/\delta \cdot [p^{(k)}_X]_j + \sum_{i \neq j} [p^{(k)}_Y]_i q_{ij} \). By equating them, we get

\[
[p^{(k)}_Y]_j \cdot r_j^{(k)} - \sum_{i \in S} [p^{(k)}_Y]_i q_{ij} \cdot \delta = \lambda^{(k)} \cdot [p^{(k)}_X]_j, \tag{38}
\]

which, letting \( \delta \to 0 \), establishes (35).

It is worth noting that (35) is the continuous counterpart of [21, Theorem 5], while the derivative of (34) is the one of [21, Theorem 6]. The relation between \( X \) and \( Y \) in the Laplace transform domain, given by the following corollary, is similar to the result obtained for discrete-state (non-fluid) queues in [21, Equation (30)].

**Corollary 4** The LSTs of the distribution functions of \( X \) and \( Y \), denoted by \( \bar{f}_X^*(s) = \int_0^\infty e^{-st} dF_X(t) \) and \( \bar{f}_Y^*(s) = \int_0^\infty e^{-st} dF_Y(t) \), are related by

\[
\bar{f}_Y^*(s) (s R^{(k)} - Q) = \lambda^{(k)} s \bar{f}_X^*(s). \tag{39}
\]

Next, the class \( k \) queue length moments are derived based on the ones embedded at departure epochs. The moments are obtained by taking the derivatives of the LST, i.e.,

\[
E(\mathcal{Y}^m) = (-1)^m \frac{d^m}{ds^m} \bar{f}_Y^*(s)|_{s \to 0} \mathbb{I}. \tag{40}
\]

The following theorem (which is the continuous counterpart of [21, Section 4.1]) provides a recursive algorithm to compute the vectors \( y^{(m)} \).

**Theorem 4** The vectors \( y^{(m)} \) can be obtained recursively by

\[
y^{(m)} = y^{(m)} \mathbb{I} \pi - m(y^{(m-1)} R^{(k)} - \lambda^{(k)} x^{(m-1)}) (\mathbb{I} \pi - Q)^{-1}, \tag{41}\]

for \( m > 0 \), where \( y^{(0)} \mathbb{I} \) is given by

\[
y^{(m)} \mathbb{I} = x^{(m)} \mathbb{I} + m(y^{(m-1)} R^{(k)} - \lambda^{(k)} x^{(m-1)}) (\mathbb{I} \pi - Q)^{-1} R^{(k)}/\lambda^{(k)} \mathbb{I}. \tag{42}\]
The vectors $x^{(m)}$ are defined by $x^{(m)} = \frac{d^m}{ds^m} f^*_X(s)|_{s \to 0}$; hence,

$$x^{(m)} = \begin{cases} \hat{\kappa}^{(k)} \Phi^{(m)}, & m > 0, \\ \hat{\alpha}^{(k)} + \hat{\kappa}^{(k)} \Phi^{(0)}, & m = 0. \end{cases} \quad (43)$$

For $m = 0$, we have $y^{(0)} = \pi$, where $\pi$ is the stationary probability vector of $Q$.

**Proof** For $m = 0$, the elements of the vector $y^{(0)}$ are the stationary probabilities of the background process, from which $y^{(0)} = \pi$ follows.

Taking the derivative of (39) $m$ times ($m > 0$) and setting $s = 0$ leads to

$$y^{(m)}Q = m y^{(m-1)}R^{(k)} - m \lambda^{(k)} x^{(m-1)}.$$ \quad (44)

Since the matrix $Q$ is not invertible, $y^{(m)}$ cannot be expressed from (44) directly. Let us subtract $y^{(m)} 1_\pi$ from both sides of (44). We get

$$y^{(m)}(1_\pi - Q) = y^{(m)} 1_\pi - m y^{(m-1)}R^{(k)} - m \lambda^{(k)} x^{(m-1)}.$$ \quad (45)

Observing that $1_\pi - Q$ is invertible and that $1_\pi (1_\pi - Q)^{-1} = 1_\pi$ provides (41). Multiplying both sides of (41) by $R^{(k)} 1$ from the right yields

$$\frac{y^{(m)} R^{(k)} 1}{\lambda^{(k)} x^{(m)} 1} = \frac{y^{(m)} 1_\pi R^{(k)} 1 - m y^{(m-1)}R^{(k)} - \lambda^{(k)} x^{(m-1)}}{1_\pi - Q} (1_\pi - Q)^{-1} R^{(k)} 1.$$

Exploiting that multiplying (44) by $1$ from the right gives $y^{(m)} R^{(k)} 1 = \lambda^{(k)} x^{(m)} 1$ and that $\lambda^{(k)} = \pi R^{(k)} 1$, (42) is proven. \hfill \Box

With the theorems and corollaries presented above, it is possible to compute the LST and the moments of the queue length distribution. Finally, it remains to obtain the order-$n$ approximation of the class $k$ queue length distribution in the time domain, $F^{(n)} (x)$, based on Erlangization.

Assume that the order-$n$ approximation of the queue length distribution at fluid drop departures is available in the form of (33), \( F^{(n)} (x) = \tilde{\alpha}^{(k)} 1 + \tilde{\kappa}^{(k)} \Phi_n (x) 1 \), where $\Phi_n (x) = \sum_{\ell=0}^{n-1} \hat{\psi}_\ell (n/x)$ (as given in (16)). The following theorem demonstrates how to obtain the approximation of $F^{(n)} (x)$ from the same $\hat{\psi}_\ell (n/x)$ matrices.

**Theorem 5** The order-$n$ approximation of the distribution function $F_Y (x)$ is $\underline{F}_Y^{(n)} (x) = F^{(n)} (x) 1$, where the row vectors $\underline{F}_Y^{(n)} (x)$ are defined recursively by

$$\underline{F}_Y^{(n)} (x) = v \left( \lambda^{(k)} \hat{\kappa}^{(k)} \hat{\psi}_n (v) + \underline{F}_Y^{(n-1)} (x) R^{(k)} \right) (v R^{(k)} - Q)^{-1},$$ \quad (46)

for $n > 1$, and

$$\underline{F}_Y^{(1)} (x) = v \lambda^{(k)} (\hat{\alpha}^{(k)} + \hat{\kappa}^{(k)} \hat{\psi}_0 (v)) (v R^{(k)} - Q)^{-1},$$ \quad (47)

\(\Box\) Springer
for \( n = 1 \), where \( v = n/x \).

**Proof** The idea behind Erlangization is to express the distribution function as the limit of \( F_Y^{(n)}(x) = \int_0^\infty f_{\mathcal{E}(n,v)}(u) P(Y < u) \, du \). The state-dependent version of the formula is \( [F_Y^{(n)}(x)]_i = \int_0^\infty f_{\mathcal{E}(n,v)}(u) P(Y < u, J = i) \, du \).

Multiplying both sides of (34) by an Erlang density \( f_{\mathcal{E}(n,v)}(u) \) and taking the integral from 0 to \( \infty \) gives

\[
\int_0^\infty f_{\mathcal{E}(n,v)}(u) f_Y(u) \, du \, R^{(k)} - \int_0^\infty f_{\mathcal{E}(n,v)}(u) F_Y(u) \, du \, Q = \lambda^{(k)} \int_0^\infty f_{\mathcal{E}(n,v)}(u) f_Y(u) \, du
\]

which, when integrating both sides by parts, transforms to

\[
\left[ f_{\mathcal{E}(n,v)}(u)(F_Y(u) - p_Y) \right]_0^\infty R^{(k)} - \int_0^\infty v f_{\mathcal{E}(n-1,v)}(u)(F_Y(u) - p_Y) R^{(k)} \, du \\
+ \int_0^\infty v f_{\mathcal{E}(n,v)}(u)(F_Y(u) - p_Y) R^{(k)} \, du - \int_0^\infty f_{\mathcal{E}(n,v)}(u) F_Y(u) Q \, du \\
= \lambda^{(k)} \left[ f_{\mathcal{E}(n,v)}(u)(F_X(u) - p_X) \right]_0^\infty - \lambda^{(k)} \int_0^\infty v f_{\mathcal{E}(n-1,v)}(u)(F_X(u) - p_X) \, du \\
+ \lambda^{(k)} \int_0^\infty v f_{\mathcal{E}(n,v)}(u)(F_X(u) - p_X) \, du,
\]

for \( n > 1 \), where we have exploited that \( \frac{d}{du} f_{\mathcal{E}(n,v)}(u) = v f_{\mathcal{E}(n-1,v)}(u) - v f_{\mathcal{E}(n,v)}(u) \).

The terms in the square brackets are zero, since \( F_X(0) = p_X \) and \( F_Y(0) = p_Y \) hold, and \( f_{\mathcal{E}(n,v)}(u)\rvert_{u=\infty} = 0 \) holds, too. The remaining terms can be expressed as

\[
-v F_Y^{(n-1)}(x) R^{(k)} + v F_Y^{(n)}(x) R^{(k)} - F_Y^{(n)}(x) Q = \lambda^{(k)} v(F_Y^{(n)}(x) - F_Y^{(n-1)}(x)),
\]

which equals (46) since \( F_Y^{(n)}(x) - F_Y^{(n-1)}(x) = \hat{k}^{(k)} \hat{\Psi}_{n-1} \) (see (33) and (16)).

Following similar steps for \( n = 1 \) gives

\[
\left[ f_{\mathcal{E}(1,v)}(u)(F_Y(u) - p_Y) \right]_0^\infty R^{(k)} + v \int_0^\infty f_{\mathcal{E}(1,v)}(u)(F_Y(u) - p_Y) R^{(k)} \, du \\
- \int_0^\infty f_{\mathcal{E}(1,v)}(u) F_Y(u) Q \, du \\
= \lambda^{(k)} \left[ f_{\mathcal{E}(1,v)}(u)(F_X(u) - p_X) \right]_0^\infty + \lambda^{(k)} v \int_0^\infty f_{\mathcal{E}(1,v)}(u)(F_X(u) - p_X) \, du,
\]

which, since \( \lim_{v \to 0} f_{\mathcal{E}(1,v)}(u) = v \), simplifies to

\[
v F_Y^{(1)}(x) R^{(k)} - v p_Y R^{(k)} - F_Y^{(1)}(x) Q = \lambda^{(k)} v(F_Y^{(1)}(x) - p_X),
\]

(48)
proving (47), since \( p_X = \hat{\alpha}^{(k)} \), \( F_X^{(1)}(x) - p_X = \hat{\kappa}^{(k)} \hat{\Psi}_0 \) and \( p_Y R^{(k)} = \lambda^{(k)} p_X \) due to (35).

6 Numerical examples

The proposed algorithms have been implemented in the MATLAB environment and are available to download\(^1\). This implementation uses the ADDA algorithm [22] to solve the Riccati equations, and the \( \text{lyap} \) function of MATLAB to solve the Sylvester equations (the \( \text{lyap} \) function is based on the Hessenberg–Schur algorithm [23]). All numerical experiments have been executed on an average PC with a CPU clocked at 3.4 GHz and 4 GB of RAM.

6.1 Example 1

In the first example, there are three fluid types and the background process has four states. The generator of the CTMC and the fluid input rates in various states are given by the matrices

\[
\begin{align*}
Q &= \begin{bmatrix}
-8 & 5 & 0 & 3 \\
3 & -4 & 0 & 1 \\
4 & 6 & -10 & 0 \\
2 & 3 & 10 & -15
\end{bmatrix}, \\
R^{(1)} &= \begin{bmatrix}
1 \\
0 \\
2 \\
1
\end{bmatrix}, \\
R^{(2)} &= \begin{bmatrix}
2 \\
0 \\
4 \\
1
\end{bmatrix}, \\
R^{(3)} &= \begin{bmatrix}
4.5 \\
1 \\
0 \\
2
\end{bmatrix},
\end{align*}
\]

and the service rate is \( d = 4 \). With these parameters, the utilization of the system is \( \lambda/d = 0.875 \).

The algorithms returned the waiting time and queue length moments without a measurable delay. For the numeric results, see Table 1.

Obtaining the distributions is, however, more demanding computationally. The accuracy and the execution speed depend on how many terms are taken into account during the Erlangization. Our experience is in line with the earlier reports (see [15]),

---

\(^1\) The implementation can be downloaded from http://www.hit.bme.hu/~ghorvath/software.
namely that Erlangization is able to provide relatively accurate results with a small number of terms (the parameter $n$). Figure 3 depicts the distribution functions of class 1 waiting times for different values of $n$. For $n = 10$, the approximation has some visible error, but the approximate distribution functions for $n \geq 25$ are almost indistinguishable from each other.

Profiling this example revealed that the bottleneck of the Erlangization algorithm is the computation of the twofold summation at line 5 in (17). This term, however, is zero when the workload process of class 2 and 3 jobs does not have zero rates. The workload process does have a zero rate in this example, since in state 3 we have $r_3^{(2)} + r_3^{(3)} = d$. Changing $r_3^{(2)}$ to a value different from 4 eliminated the zero state, leading to an order of magnitude faster execution times\(^2\) (Table 2).

### 6.2 Example 2

The aim of the second example is to demonstrate how the speed of the algorithms scales with the size of the input. Let us define the model as the superposition of $N$ identical on–off sources with exponentially distributed on and off periods, with transition rates denoted by $\nu$ and $\gamma$, respectively. Each source generates fluid with rate $\rho/N$ while being in the on state. The matrix parameters of the superposed fluid model are given by

\[^2\] Our implementation detects the lack of zero states automatically for faster execution.
where $N$ determines the size of the model. The matrix $Q_N$ is the generator of the background process, and the matrix $R_N$ is the diagonal matrix of the overall fluid rates. If we assume that 10% of the fluid generated by the sources is class 1 fluid, 30% is class 2 fluid and 60% is class 3 fluid, the per class fluid input rates are $R^{(1)}_N = 0.1 \cdot R_N$, $R^{(2)}_N = 0.3 \cdot R_N$ and $R^{(3)}_N = 0.6 \cdot R_N$.

In this example, the parameters are $\nu = 1/N$ and $\gamma = 1/(3N)$ (the reason for making them dependent on $N$ is to keep the values of the generator in the same range irrespective of the size), and the fluid rate parameter is $\rho = 50$. With these settings, the mean fluid arrival rate is $\lambda = 12.5$.

We have computed the first five moments of the queue length of all job classes with different model sizes (controlled by the parameter $N$) and with different utilizations $u$ (controlled by the service rate $d = \lambda/u$). The execution times are depicted in Fig. 4, which reports the average of multiple repetitions for every point, together with the bootstrap-t confidence intervals (that are very tight).

As visible in the figure, the presented algorithms are able to solve huge models with 1000 states in reasonable time, without any numerical problems. The execution times depend on the utilization as well, since this affects the number of positive and negative states. At low utilization, obtaining five moments of the queue length for all job classes requires 10 s for $N = 1000$, while at high utilization about 40 s are required for the largest model.
Appendix A: Calculating the derivatives of $F^*_{ij}(v)$

This section describes how to calculate the derivatives of matrix functions of the form $F(v) = (vD - F)^{-1}$ (where $D$ is a diagonal matrix composed of the reward rates), necessary to obtain the matrices $F^{(m)}_{ij}$ in Corollary 1.

Taking the derivatives of $F(v)$ is trivial if all entries of $D$ are nonzero. If this is not the case, the matrices involved must be re-partitioned according to the sign of the reward rates, and hence,

$$D = \begin{bmatrix} D_+ & 0 \\ 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} F_{++} & F_{+0} \\ F_{0+} & F_{00} \end{bmatrix}.$$

Note that this partitioning is not the same as the one assumed in Sect. 3. In Sect. 3, the partitioning is done according to the fluid rates; here, it is done according to the reward rates.

According to [24], block matrices can be inverted using

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1}D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix}.$$

Introducing $X(v) = (F_{++} - vD_+ + F_{+0}(-F_{00})^{-1}F_{0+})^{-1}$, this means that

$$(vD - F)^{-1} = \begin{bmatrix} -X(v) & X(v)F_{+0}F_{00}^{-1} \\ F_{00}^{-1}F_{0+}X(v) - F_{00}^{-1} - F_{00}^{-1}F_{0+}X(v)F_{+0}F_{00}^{-1} \end{bmatrix}.$$  

(50)

Since the only term in (50) which depends on $v$ is $X(v)$, the derivatives of $(vD - F)^{-1}$ can now be easily obtained since

$$\frac{d^m}{dv^m} X(v) = (-1)^{m+1} m! (vI - (F_{++} + F_{+0}(-F_{00})^{-1}F_{0+})D_+)^{-m-1}D_+^{-1}.$$  

(51)
Appendix B: The proof of Theorem 2

Examining (15), it can be observed that the entries of the matrix \( \Phi_n(t) \) are the probabilities that the accumulated reward is less than an order-\( n \) Erlang-distributed random variable with parameter \( \nu \). Hence, the matrix \( \Phi_n(t) \) is obtained by creating a process where the reward accumulation process and the Erlang distribution are running in parallel and are competing. To this end, the state space of the background process (given by the generator matrix \( F \)) has to be extended such that it does not only keep track of the state of \( J(t) \), but also counts the number of events generated by a Poisson process having parameter \( \nu \). At the end of the busy period, the probabilities of those states, where the counter is less than \( n \), contribute to \( \Phi_n(t) \). To take the state-dependent reward rates into account, the transition rate incrementing the counter must be \( \nu \cdot d_i \) if \( J(t) = i \). Thus, the time is accelerated for the counting process if the reward rate is high in the current state, and it is slowed down if the reward rate is low.

The generator matrix and the fluid rates of the modified process, which also includes the counter, have the following block structure:

\[
\tilde{F}(\nu) = \begin{bmatrix}
F - \nu D & \nu D \\
F - \nu D & \nu D \\
& & \ddots \\
& & & \ddots \\
& & & & \ddots \\
& & & & & \ddots
\end{bmatrix},
\tilde{C} = \begin{bmatrix}
C \\
C \\
& & \ddots \\
& & & \ddots \\
& & & & \ddots \\
& & & & & \ddots
\end{bmatrix},
\tag{52}
\]

where being in block \( i \) at time \( t \) means that the value of the counter is \( i \) at time \( t \).

By partitioning the state space according to the sign of the fluid rates, the matrices \( \tilde{F}(\nu) \) and \( \tilde{C} \) become

\[
\tilde{F}(\nu) = \begin{bmatrix}
F_{+,+} - \nu D_{+} & \nu D_{+} & F_{+,+} & F_{+,0} \\
F_{-,+} - \nu D_{-} & \nu D_{+} & F_{-,+} & F_{-,0} \\
& & & \ddots \\
& & & & \ddots \\
& & & & & \ddots
\end{bmatrix},
\tilde{C} = \begin{bmatrix}
C_+ \\
C_+ \\
& & \ddots \\
& & & \ddots \\
& & & & \ddots \\
& & & & & \ddots
\end{bmatrix},
\]

where each of the \( 3 \times 3 \) partitions of the matrices \( \tilde{F}(\nu) \) and \( \tilde{C} \) have size infinity.

To obtain the state of the background process at the end of the busy period, the blocks of these matrices need to be inserted into (7), leading to
\[ \begin{align*}
0 &= \hat{C}_+^{-1}(\hat{F}_{++}(v) + \hat{F}_{+0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0+}(v))\hat{\Psi}(v) \\
&\quad + \hat{\Psi}(v)(\hat{C}_+^{-1}(\hat{F}_{++}(v) + \hat{F}_{-0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0-}(v)) \\
&\quad + \hat{\Psi}(v)(\hat{C}_+^{-1}(\hat{F}_{+-}(v) + \hat{F}_{+0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0+}(v))\hat{\Psi}(v) \\
&\quad + \hat{C}_+^{-1}(\hat{F}_{+-}(v) + \hat{F}_{+0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0-}(v)).
\end{align*} \tag{53}\]

Due to the structure of the matrices \(\hat{F}(v)\) and \(\hat{C}\), the inverse matrix \((\hat{F}_{00}(v))^{-1}\) has an upper block-triangular structure:

\[ \begin{align*}
(\hat{F}_{00}(v))^{-1} &= \begin{bmatrix}
Z_0(v) & Z_1(v) & Z_2(v) & \ldots \\
Z_0(v) & Z_1(v) & \ldots \\
Z_0(v) & \ldots \\
\vdots & \ddots & \ddots
\end{bmatrix},
\end{align*} \tag{54}\]

with \(Z_\ell(v) = H_{00}^\ell(v)(vD_0H_{00}^\ell(v))^\ell\). Investigating the structure of the coefficient matrices of (53), namely

\[
\begin{align*}
M_{++}(v) &= \hat{C}_+^{-1}(\hat{F}_{++}(v) + \hat{F}_{+0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0+}(v)), \\
M_{+-}(v) &= \hat{C}_+^{-1}(\hat{F}_{+-}(v) + \hat{F}_{+0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0-}(v)), \\
M_{-+}(v) &= (-\hat{C}_+)^{-1}(\hat{F}_{-+}(v) + \hat{F}_{-0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0+}(v)), \\
M_{--}(v) &= (-\hat{C}_+)^{-1}(\hat{F}_{--}(v) + \hat{F}_{-0}(v)(\hat{F}_{00}(v))^{-1}\hat{F}_{0-}(v)),
\end{align*}
\]

reveals that all of them are upper block-triangular, too; hence, we have

\[ M_{ij}(v) = \begin{bmatrix}
M_{ij}^{(0)}(v) & M_{ij}^{(1)}(v) & M_{ij}^{(2)}(v) & \ldots \\
M_{ij}^{(0)}(v) & M_{ij}^{(1)}(v) & \ldots \\
M_{ij}^{(0)}(v) & \ldots \\
\vdots & \ddots & \ddots
\end{bmatrix}, \tag{55}\]

where \(i \in \{+, -\}\) and \(j \in \{+, -\}\). The block structure of the coefficient matrices implies that the solution of the NARE

\[ \begin{align*}
0 &= M_{++}(v)\hat{\Psi}(v) + \hat{\Psi}(v)M_{+-}(v) + \hat{\Psi}(v)M_{-+}(v)\hat{\Psi}(v) + M_{--}(v)
\end{align*} \tag{56}\]

has an upper block-triangular structure as well, i.e.,

\[ \hat{\Psi}(v) = \begin{bmatrix}
\hat{\Psi}_0(v) & \hat{\Psi}_1(v) & \hat{\Psi}_2(v) & \ldots \\
\hat{\Psi}_0(v) & \hat{\Psi}_1(v) & \ldots \\
\hat{\Psi}_0(v) & \ldots \\
\vdots & \ddots & \ddots
\end{bmatrix}. \tag{57}\]
With the given coefficient matrices, the first block row and the $\ell+1$th block column of the NARE (56) become

\begin{equation}
0 = \sum_{i=0}^{\ell} M_{++}^{(i)}(\nu) \hat{\Psi}_{-i}(\nu) + \sum_{i=0}^{\ell} \hat{\Psi}_{-i}(\nu) M_{-+}^{(\ell-i)}(\nu) \\
+ \sum_{j=0}^{\ell} \sum_{i=0}^{j} \hat{\Psi}_{-i}(\nu) M_{-+}^{(j-i)}(\nu) \hat{\Psi}_{-j}(\nu) + M_{-+}^{(\ell)}(\nu),
\end{equation}

from which $\hat{\Psi}_{0}(\nu) = \Phi^{*}(\nu)$ follows immediately for $\ell = 0$. In order to obtain a recursion for $\hat{\Psi}_{\ell}(\nu)$, $\ell > 0$, we separate all terms involving $\hat{\Psi}_{\ell}(\nu)$ from those that do not depend on it, yielding

\begin{equation}
0 = (M_{++}^{(0)}(\nu) + \hat{\Psi}_{0}(\nu) M_{--}^{(0)}(\nu)) \hat{\Psi}_{\ell}(\nu) + \hat{\Psi}_{\ell}(\nu) (M_{--}^{(0)}(\nu) + M_{--}^{(0)}(\nu) \hat{\Psi}_{0}(\nu)) \\
+ \sum_{i=1}^{\ell} M_{++}^{(i)}(\nu) \hat{\Psi}_{-i}(\nu) + \sum_{i=0}^{\ell-1} \hat{\Psi}_{-i}(\nu) M_{-+}^{(\ell-i)}(\nu) + M_{-+}^{(\ell)}(\nu) \\
+ \sum_{j=1}^{\ell-1} \sum_{i=0}^{j} \hat{\Psi}_{-i}(\nu) M_{-+}^{(j-i)}(\nu) \hat{\Psi}_{-j}(\nu) + \sum_{i=0}^{\ell-1} \hat{\Psi}_{-i}(\nu) M_{-+}^{(\ell-i)}(\nu) \hat{\Psi}_{0}(\nu).
\end{equation}

Inserting the definitions of the coefficient matrices into the equation gives (17).

Finally, as the matrix $\hat{\Psi}_{\ell}(\nu)$ contains the probabilities that the counter value is $\ell$ when the busy period ends, (16) follows. \qed

**References**

1. Miller Jr., R.G.: Priority queues. Ann. Math. Stat. 31, 86–103 (1960)
2. Takine, T.: A nonpreemptive priority MAP/G/1 queue with two classes of customers. J. Oper. Res. Soc. Jpn. 39(2), 266–290 (1996)
3. Horváth, G.: Efficient analysis of the MMAP[K]/PH[K]/1 priority queue. Eur. J. Oper. Res. 246(1), 128–139 (2015)
4. Ramaswami, V.: Matrix analytic methods for stochastic fluid flows. In: Teletraffic Engineering in a Competitive World—Proceedings of the 16th International Teletraffic Congress (ITC 16), Elsevier, pp. 1019–1030 (1999)
5. Knessl, C., Tier, C.: A simple fluid model for servicing priority traffic. IEEE Trans. Autom. Control 46(6), 909–914 (2001)
6. Liu, Y., Gong, W.: On fluid queueing systems with strict priority. IEEE Trans. Autom. Control 48(12), 2079–2088 (2003)
7. Narayanan, A., Kulkarni, V.: First passage times in fluid models with an application to two priority fluid systems. In: Proceedings of IEEE International Computer Performance and Dependability Symposium. IEEE, pp. 166–175 (1996)
8. Zhang, J.: Performance study of Markov modulated fluid flow models with priority traffic. In: INFOCOM’93. Proceedings. Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future. IEEE, pp. 10–17 (1993)
9. Choi, B.D., Choi, K.B.: A Markov modulated fluid queueing system with strict priority. Telecommun. Syst. 9(1), 79–95 (1998)
10. Tzenova, E.I., Adan, I., Kulkarni, V.G.: A two-priority fluid flow model: joint steady state distribution of the buffer content processes. Technische Universiteit Eindhoven, Department of Mathematics and Computing Science, TU/e (2004)

11. Takine, T.: The workload in the MAP/G/1 queue with state-dependent services: its application to a queue with preemptive resume priority. Stoch. Models 10, 183–204 (1994)

12. Kulkarni, V.G.: Fluid models for single buffer systems. Front. Queueing Models Appl. Sci. Eng. 321, 338 (1997)

13. Kankaya, H.E., Akar, N.: Solving multi-regime feedback fluid queues. Stoch. Models 24(3), 425–450 (2008)

14. Ahn, S., Ramaswami, V.: Efficient algorithms for transient analysis of stochastic fluid flow models. J. Appl. Probab. 42, 531–549 (2005)

15. Ramaswami, V., Woolford, D.G., Stanford, D.A.: The Erlangization method for Markovian fluid flows. Ann. Oper. Res. 160(1), 215–225 (2008)

16. Kulkarni, V.G., Tzenova, E.: Mean first passage times in fluid queues. Oper. Res. Lett. 30(5), 308–318 (2002)

17. Bean, N.G., O’Reilly, M.M.: A stochastic two-dimensional fluid model. Stoch. Models 29(1), 31–63 (2013)

18. Dzial, T., Breuer, L., da Silva Soares, A., Latouche, G., Remiche, M.-A.: Fluid queues to solve jump processes. Perform. Eval. 62(1), 132–146 (2005)

19. Takine, T., Takahashi, Y.: On the relationship between queue lengths at a random instant and at a departure in the stationary queue with BMAP arrivals. Stoch. Models 14(3), 601–610 (1998)

20. Kim, N.K., Chang, S.H., Chae, K.C.: On the relationships among queue lengths at arrival, departure, and random epochs in the discrete-time queue with D-BMAP arrivals. Oper. Res. Lett. 30(1), 25–32 (2002)

21. Lucantoni, D.M., Meier-Hellstern, K.S., Neuts, M.F.: A single-server queue with server vacations and a class of non-renewal arrival processes. Adv. Appl. Probab. 22, 676–705 (1990)

22. Wang, W.-G., Wang, W.-C., Li, R.-C.: Alternating-directional doubling algorithm for M-matrix algebraic Riccati equations. SIAM J. Matrix Anal. Appl. 33(1), 170–194 (2012)

23. Golub, G.H., Nash, S., Van Loan, C.: A Hessenberg–Schur method for the problem AX+XB=C. IEEE Trans. Autom. Control 44(6), 909–913 (1979)

24. Bernstein, D.S.: Matrix Mathematics: Theory, Facts, and Formulas. Princeton University Press, Princeton (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.