A Robust Queueing Network Analyzer Based on Indices of Dispersion

Ward Whitt* Wei You†

ww2040@columbia.edu weiyou@ust.hk

October 16, 2020

Abstract

We develop a robust queueing network analyzer algorithm to approximate the steady-state performance of a single-class open queueing network of single-server queues with Markovian routing. The algorithm allows non-renewal external arrival processes, general service-time distributions and customer feedback. The algorithm is based on a decomposition approximation, where each flow is partially characterized by its rate and a continuous function that measures the stochastic variability over time. This function is a scaled version of the variance-time curve, called the index of dispersion for counts (IDC). The required IDC functions for the external arrival processes can be calculated from the model primitives or estimated from data. Approximations for the IDC functions of the internal flows are calculated by solving a set of linear equations. The theoretical basis is provided by heavy-traffic limits for the flows established in our previous papers. A robust queueing technique is used to generate approximations of the mean steady-state performance at each queue from the IDC of the total arrival flow and the service specification at that queue. The algorithm effectiveness is supported by extensive simulation studies.

Keywords: *queueing networks, non-Markov queueing networks, robust queueing, index of dispersion, queueing approximations, heavy traffic*

*Department of Industrial Engineering and Operations Research, Columbia University

†Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology

1 Introduction

This paper contributes to analytical methods for designing and optimizing service systems. Such systems appear in a broad and diverse range of settings, including customer contact centers, hospitals, airlines, online marketplaces, ride-sharing platforms and cloud computing networks. The design and operation of these systems is challenging, largely because there is uncertainty about when customers will arrive and their service requirements.

Fortunately, useful guidance can often be provided by exploiting mathematical models using stochastic processes. Prominent among these are stochastic queueing network models, because service is often provided in a sequence of steps; e.g., see [5, 8]. There is an extensive literature on the applications of queueing network models to service systems. For example, see [43] for a review of applications in computer networks, see [4, 19, 39] for examples in ride-sharing economies and see [7, 10, 12, 32, 61] for healthcare-related applications.

Service operation policies often rely on quantitative descriptions of the system performance, called *performance measures*, such as the waiting time, the queue length, and the workload in the system. Decision support for service operations relies on accurate characterization of these performance measures.

A standard way to analyze the performance of complex queueing models is to employ computer simulation, e.g., see [46, 62]. However, as noted in [16], a great disadvantage of simulation-based optimization methods is the often prohibitive computation time required to obtain optimal solutions for service operation problems involving a multidimensional stochastic network. Thus, analytical analysis of the models can be very helpful. However, the class of queueing networks that can be solved analytically requires strong assumptions that are rarely satisfied, whereas more realistic models are prohibitively hard to analyze exactly. Hence, analytical performance approximation of queueing networks remains an important tool.

In this paper, we provide a new efficient algorithm to approximate the steady-state performance measures in a single-class open queueing network (OQN) with Markovian routing, unlimited waiting space and the first-come first-served (FCFS) service discipline. We focus on non-Markov OQNs where the external arrival processes need not be Poisson or renewal and the service-time distributions need not be exponential. Our algorithm is a decomposition approximation, which combines three methodologies in operations research and stochastic models: (i) robust optimization as in [3, 57], (ii) indices of dispersion and stationary point processes as in [9, 15, 45] and (iii) heavy-traffic limits as in [11, 22, 55]. However, the paper has been written to emphasize the efficient effective algorithm that is obtained in the end by synthesizing these methodologies.

1.1 Approximation Algorithms

In this section, we briefly review existing approximation algorithms for non-Markov OQNs; additional literature review appears in an online appendix.

1.1.1 Decomposition Approximations

Under the assumption of Poisson arrival processes and exponential service-time distributions, our OQN is a Markov model, called a Jackson network, which is easy to analyze, primarily because the steady-state distribution of the queue lengths has a product form; i.e., the steady-state queue lengths are independent geometric random variables, just as if each queue were independent $M/M/1$ queues. The arrival rate at each queue can be obtained by solving a system of linear equations called the traffic rate equations. Motivated by that product-form property of Markov OQNs, decomposition approximations for non-Markov OQNs have been widely investigated. In this approach, the network is decomposed into individual single-server queues, and the steady-state queue length processes are assumed to be approximately independent. For example, in [33] and [50] each queue is approximated by a $GI/GI/1$ model, where the arrival and service processes are approximated by a renewal process partially characterized by the mean and *squared coefficient of variation* (scv, variance divided by the square of the mean) of an interarrival or service time.

While the decomposition approximations do often perform well, it was recognized that dependence in the arrival processes of the internal flows can be a significant problem. The approximation for superposition processes used in the QNA algorithm [50] attempts to address the dependence. Nevertheless, significant problems remained, as was dramatically illustrated by comparisons of QNA to model simulations in [17, 47, 48], as discussed in [54].

To address the dependence in arrival processes, decomposition methods based on Markov Arrival Processes (MAPs) have been developed. The MAP was introduced by Neuts [37]; see Ch. XI of [2]. Since it is not a renewal process, so that it can model the autocorrelation in the arrival and service processes. Horváth et al. [25] approximated each station by a MAP/MAP/1 model. Kim [30, 31] approximated each queue by a MMPP(2)/GI/1 model, where the arrival process is a Markov-modulated Poisson process with two states (a special MAP).

1.1.2 Heavy-Traffic Limit Approximations

The early decomposition approximation in [50] drew heavily on the central limit theorem (CLT) and heavy-traffic (HT) limit theorems. Approximations for a single queue follow from [26, 27]. With these tools, approximations for general point processes and arrival processes were developed in [49, 51]. Heavy-traffic approximation of queues with superposition arrival processes in [52] helped capture the impact of dependence in such queues.

Another approach is to apply heavy-traffic limit theorems for the entire network. Such HT limits were established for feedforward OQNs in [26, 27] and Harrison [20, 21], and then for general OQNs by Reiman [40]. These works showed that the queue length process converges to a multidimensional reflected Brownian motion (RBM) as every service station approaches full saturation simultaneously.

These general heavy-traffic results for OQNs lead to approximations using the limiting RBM processes. The QNET algorithm in Harrison and Nguyen [22] provides such an approximation. Theoretical and numerical analysis of the stationary distribution of the multi-dimensional RBM was studied in [13, 23, 24].

As a crucial step of the QNET algorithm, Dai and Harrison [13] proposed a numerical algorithm to calculate the steady-state density of an RBM, but it require considerable computation time. The computational accuracy of that algorithm improves as the number of iteration n grows, and the author's there note that $n = 5$ generally gives satisfactory answers. For a OQN with d stations, the computational complexity is $O(d^{2n})$, see Section 6 of [13]. A further limitation is that the underlying theorem is for a sequence of OQNs in which the associated sequences of traffic intensities at all queues approaches the critical value. For practical application to large-scale systems or small systems with a wide range of traffic intensities, hybrid methods that combine a decomposition approximation and heavy-traffic theory were proposed in Reiman [41] and Dai et al. [11]. The version [11] has been shown to be remarkably effective, but it requires the numerical solution of RBMs.

The present paper also relies heavily on heavy-traffic limit theorems, but here we exploit our recent heavy-traffic limits for the flows in [56, 60].

1.1.3 Robust Queueing Approximations

Recently, a novel Robust Queueing (RQ) approach to analyze queueing performance in single-server queues was proposed by Bandi et al. [3]. The key idea in RQ is to replace the underlying probability law by a suitable uncertainty set, and analyze the (deterministic) worst case performance. The authors there relied on the discrete-time Lindley's recursion to characterize the customer waiting times as a supremum over partial sums of the interarrival times and service times. Uncertainty sets for the sequence of partial sums are proposed based on central limit theorem and two-moment partial traffic descriptions of the arrival process and service process.

Although the general RQ idea is simple and good, there remain challenges in identifying proper uncertainty sets and making connection to the original queueing system. These challenges were addressed in [57], which lay the foundation for this paper. In [57], the authors proposed a new non-parametric RQ formulation for approximating the continuous-time workload process in a single-server queue, and proved asymptotic exactness of their approximations light and heavy traffic. We briefly review this new RQ formulation in Section 2.2.

1.1.4 Non-Parametric Traffic Descriptions

As a trade-off for mathematical tractability, all approximation approaches so far rely on incomplete traffic descriptions. For example, the approximation approaches reviewed above can be characterized as parametric approaches, typically involving only means and variances of random variables, The general stochastic system is then mapped into one of a parametric family of highly structured models. Such approaches rely on a small finite set of parameters as traffic descriptions and a key step is to understand how these parameters for each arrival process evolve in the network.

Another stream of research models the temporal dependence in the stochastic processes by non-parametric traffic descriptions. In Jagerman et al. [28], the authors approximate a general stationary arrival process by a Peakedness Matched Renewal Stream (PMRS). The key ingredient is the peakedness function, which is determined by the arrival point process and the first two moments of the service-time distribution; see [34] for additional discussion. However, [28] relied on a two-parameter approximation for the peakedness function of a stationary point process, where the parameters are estimated by simulation. Similar non-parametric traffic descriptions have been studied in [28, 35, 36], but they only focus on single-station single-server queues.

We adopt a non-parametric approach to describe the arrival and service processes in an OQN. Let A be a *stationary* counting process, e.g. the arrival counting process at a queue. We partially characterize A by its rate and its *Index of Dispersion for Counts* (IDC), a function of non-negative real numbers $I_A : \mathbb{R}^+ \to \mathbb{R}^+$ defined as in §4.5 of [9],

$$I_A(t) \equiv \frac{\mathrm{Var}(A(t))}{E[A(t)]}, \quad t \geq 0. \tag{1}$$

A reference case is the Poisson process, where the IDC is a constant function $I_A(t) \equiv 1$. As regularity conditions, we assume that $E[A(t)]$ and $\mathrm{Var}(A(t))$ are finite for all $t \geq 0$. For renewal processes, it suffices to assume that the inter-renewal time distribution has finite second moment.

Being a function of time t, the IDC captures the variability in a point process over all timescales. The IDC encodes much more information about the underlying process than traditional parametric descriptions. The RQ algorithm in [57] established a bridge between the IDC traffic description and the performance measures in a single-server queue.

With the aid of the HT limits established in [56, 60], we now develop a network calculus to characterize the IDCs of the customer flows in an OQN. Similar non-parametric traffic descriptions have been studied in [28, 35, 36], but they focused on single-server queues. To the best of our knowledge, we are the first to study the non-parametric traffic descriptions in a network setting.

1.1.5 The Overall Robust Queueing Network Analyzer (RQNA)

We exploit the remarkably strong connection between the arrival IDC and the normalized workload in a single-server queue. This connection, was first exposed by Fendick and Whitt [18], but they did not produce the systematic approximations we obtained through robust queueing in [57]. We advance that approach further by showing that all these approximations can be combined to produce a *Robust Queueing Network Analyzer* (RQNA).

Our method is a decomposition approximation, because the algorithm decomposes the network into individual $G/GI/1$ models, where the arrival process and service process at each queue is partially specified by its rate and IDC, defined in (1). As in other decomposition methods, three network operations become essential: first, the *departure operation* as customers flow through a service station and an arrival process turns into a departure process; second, the *splitting operation* as a departure process split into multiple sub-processes and feed into different subsequent queues; and third, the *superposition operation* as departure flows from different queues combine together and feed into a queue.

In Section 3, we introduce a set of linear equations, which we refer to as the *IDC equations*, to describe the combined effect of these three network operations. These IDC equations are derived from the HT limits in [56, 60]. We discuss the remaining technical details in the online appendix. The IDC of the total arrival flows at each queue is approximated by the solution to the IDC equations. The RQ algorithm (13) is then applied to generate approximations of the mean steady-state performance measures at each $G/GI/1$ queue in the network.

The RQNA algorithm has a remarkably concise analytical formulation, given in (13) and (34) below, which makes it easy to implement. We discuss the computation complexity of our proposed algorithm in Remark 8. We also conduct simulation experiments to evaluate the effectiveness of the new RQNA and compare it to previous algorithms in [11, 22, 25, 50]. Our experiments indicate that RQNA performs as well or better than previous algorithms.

1.2 Network Structure and Our Contributions

In this section we briefly describe the contribution of each of our previous papers [56, 57, 58, 59, 60] and indicate how the present paper goes beyond them. To do so, it is helpful to classify OQN's according to structural complexity. We indicate the paper contributions in this taxonomy.

1. **a single $G/GI/1$ queue**

   This is an OQN with one node, where the service times are i.i.d and independent of the arrival process, but the arrival process can be general (assuming stationarity). The arrival process may be a superposition of other external arrival processes.

   Robust queueing based on the IDC is developed for this model in our first paper [57]. Indeed, since a decomposition approximation is used, the robust optimization method was established in this first paper. This paper should be the starting point for reading. The main contributions are outlined in §1.2 of [57]. A highlight is Theorem 5 there showing that the new robust queueing approximation is asymptotically exact in both light and heavy traffic. While this result provides important insight, we emphasize that the robust queueing approximation in [57] for a single $G/GI/1$ queue is not obtained directly from the heavy-traffic limit; it is not itself a heavy-traffic approximation.

   While the general framework for our robust queueing follows Bandi et al. [3], there are significant differences even for one queue. Advantages over the initial robust queueing algorithm in [3] are discussed in Remark 1 in [57].

   Further insight is provided to the performance of the $G/GI/1$ queue when the arrival process is partially characterized by the IDC in [58]. Theorem 2.1 in [58] shows that a renewal process is fully characterized by the IDC of the associated equilibrium (stationary) renewal process. As a first consequence, for a renewal process, the IDC can be computed from the Laplace transform of the interarrival-time distribution by numerical transform inversion. (That is one good way to get the required model data.) As a second consequence, a $GI/GI/1$ model is fully characterized by the IDC of the interarrival times and the IDC of the service times. That implies that any error in approximations of performance measures for a $GI/GI/1$ queue must be due to the robust queueing approximation step, because there is no model error in that case. In summary, the IDC function encodes much more information about the underlying distribution than traditional traffic descriptions.

The paper [59] is mostly unrelated to the present paper because it focuses on a single time-varying queue with a time-varying arrival-rate function. Nevertheless, that paper contributes even for one stationary $G/GI/1$ model because it shows how to develop approximations for the percentiles of the steady-state workload distribution instead of just the mean.

2. **a tree network**

This class includes queues in series, which are already very challenging OQNs. This class also allows splitting of departure processes, which necessarily is independent splitting because of the Markovian routing assumption. However, superposition of internal processes is not allowed. Even two queues in series presents challenging new problems.

The new problem presented by this class of OQNs is developing an effective approximation for the IDC of a departure process from a $G/GI/1$ queue where the arrival process is partially characterized by its IDC. Significant progress was obtained by establishing a new heavy-traffic limit theorem for the stationary departure process from a $G/GI/1$ queue in [56]. In addition, drawing on this limit theorem, an algorithm to approximate the IDC of a departure process was developed and tested in [56]. Again we emphasize that the robust queueing approximation in [56] for queues in series is not obtained directly from the heavy-traffic limit; the algorithm is not itself a heavy-traffic approximation.

We have indicated that there are significant differences between the robust queueing approximations for one queue in [3] and [57]. The full IDC-based RQNA here is even more different from the candidate full RQNA in [3]. The differences are highlighted in the comparisons for the queues in series in Tables 1 and 2 in §4 of the online supplement to [58]. These comparisons are for the same model considered in Tables 1 and 2 of the online appendix to this paper. The comparison in the case of a high-variability in Table 2 of the two appendices is especially dramatic. The errors in the total waiting time in this difficult network are 25% for QNA from [50], 19% for QNET from [22], 10% for SBD from [11] and 2% − 11% for RQNA depending on the tuning function used. In contrast, Table 2 in §4 of the online supplement to [58] shows that the corresponding errors for three candidate algorithms from [3] are 126%, 180% and 549%.

3. **feedforward network**

   This class allows superpositions of other previous arrival processes. The component arrival processes in the superposition may be dependent. Nevertheless, the feedforward property guarantees that each queue is a $G/GI/1$ model, where the service times are i.i.d and independent of the arrival process, so that each queue is of the form assumed for a single queue.

4. **general OQN allowing feedback**

   This is the general case, allowing internal feedback and thus allowing dependence among all interarrival times and service times. Each successive class in this heirarchy allows greater complexity. We had provided no algorithms for these last two classes of OQNs prior to the present paper.

The present paper develops and evaluates an algorithm based on IDCs and robust queueing to compute approximate performance measures for each queue in a general OQN, focusing especially on the two more general classes above, for which there was no previous algorithm. The algorithm requires solving a system of linear equations, so that the complexity is algorithm complexity is similar to that for QNA in [50], see Remark 9 for details.

To establish a theoretical basis for the algorithm, we established heavy-traffic limits for the stationary flows in a general OQN in [60]. That paper contributes significantly to the algorithm developed in the present paper, but just as with the previous classes of OQNs, the heavy-traffic limit itself does not directly provide the algorithm.

In summary, the robust optimization component of the new algorithm is contained in the first paper [57], with the extension to percentiles added in [59]. The remaining papers develop approximations for the IDC of the arrival processes in the network. The supporting heavy-traffic theory is contained in [56, 57, 60].

1.3 Organization

The rest of the paper is organized as follows. In §2 we define the indices of dispersion, discuss the connection between the index of dispersion for work and the mean steady-state workload, and briefly review the robust queueing algorithm for a single $G/GI/1$ queue. We also discuss how to obtain the IDC's of the external arrival processes, as required in the model data. In §3 we develop a framework for approximating the IDC's of the flows. In §3.5 we develop a relatively elementary version of the RQNA algorithm for tree-structured networks. In §4 we discuss feedback elimination. In §5 we present the full RQNA algorithm. In §6 we discuss numerical experiments. In §7 we draw conclusions. In §7.2 we indicate when the approximations are likely to be reliable or not. We present additional supporting material in an online appendix, including more experimental results.

2 The Indices of Dispersion and Robust Queueing

In this section we provide brief reviews of the IDC function in (1) and the robust queueing algorithm from [57]. In §2.1 we define another continuous-time index of dispersion: the Index of Dispersion for Work (IDW). We discuss a useful decomposition of the IDW and its connection to the IDC and the mean steady-state workload. In §2.1.2 we indicate how to calculate the IDC from a model of the arrival process; in §2.1.3 we indicate how to estimate the IDC from data. In §2.2 we review the RQ algorithm from [57], which links the IDW to approximations of the steady-state queueing performance.

2.1 The Indices of Dispersion

Consider a general single-server queue with a general arrival process A, i.e. $A(t)$ counts the total number of arrival in the time interval $[0, t]$. We assume that A is a stationary point process; see [14, 45]. The IDC defined in (1) is a continuous-time function associated with A. Being the variance function scaled by the mean function, the IDC exposes the variability over time, independent of the scale. For this reason, the IDC can be viewed as a continuous-time generalization of the squared coefficient of variation (scv, variance divided by the square of the mean) of a nonnegative random variable. The IDC captures the way that the covariance in a point process changes over time, which extends the natural practice of including lag-k covariances in modeling the dependence in a point process.

The reference case is a Poisson arrival process, for which $I_a(t) = 1$, $t \geq 0$. However, for general arrival processes, the IDC is more complicated. Even the IDC for a determinsitic D arrival process is complicated, because the IDC is for the stationary version of the arrival process, which lets the initial point be uniformly distributed over the constant interarrival time. Much of this paper is devoted to the analysis and approximation of the IDC for the arrival process at each station of the OQN.

Remark 1 (Time scaling convention) In [57] we defined the IDC and IDW in terms of rate-1 processes, so that the actual rate of the process had to be inserted as part of the time argument. In contrast, here as in [56] we let the underlying processes A and Y have any given rate, so no further scaling is needed. That changes the formulas for the IDC of a superposition process, e.g., compare (36) of [57] to (27) here. To illustrate the idea, consider $A(t)$ with rate-1 and $A_\lambda(t) \equiv A(\lambda t)$ with rate-λ. Let $I_A(t)$ denote the IDC of $A(t)$, then we have $I_{A_\lambda}(t) \equiv \mathrm{Var}(A(\lambda t))/E[A(\lambda t)] = I_A(\lambda t)$. □

Now, consider a general sequence of service times $\{V_i : i \geq 1\}$, where V_i is the service requirement of the i-th customer. Let

$$Y(t) \equiv \sum_{i=1}^{A(t)} V_i \tag{2}$$

denote the *cumulative work input process*. This process connects to the workload of a single-server queue by (9) and (10) below.

Paralleling the IDC, the *Index of Dispersion for Work* (IDW) describes the variability associated with the cumulative input process Y in (2). The IDW is defined as in (1) of [18] by

$$I_w(t) \equiv \frac{\mathrm{Var}(Y(t))}{E[V_1]E[Y(t)]}, \quad t \geq 0. \tag{3}$$

The IDW captures the cumulative variability of the total service requirement brought to the system as a function of time t, which is a key component of the new RQ approximation in [57] as we review in §2.2.

Since we are interested in the steady-state performance of the OQN, we assume that the processes A and Y have stationary increments. Given that arrival process and service times have constant determined rates, the mean functions $E[A(t)]$ and $E[Y(t)]$ are linear in time. Hence, much of the remaining behavior of the A and Y is determined by the variance-time function or index of dispersion. We are interested in the variance-time *function*, because it captures the dependence through the covariances; the processes (A, Y) have independent increments for the $M/GI/1$ model, but otherwise not.

To connect the IDC to the IDW, consider the special case where the service times V_i are i.i.d, independent of the arrival process $A(t)$. The conditional variance formula gives a useful decomposition of the IDW

$$I_w(t) = I_a(t) + c_s^2, \quad t \geq 0, \tag{4}$$

where $c_s^2 = \mathrm{Var}(V_i)/E[V_i]^2$ is the scv of the service-time distribution.

2.1.1 The IDW and the Mean Steady-State Workload

The IDC and IDW are important because of their close connection to the mean steady-state workload $E[Z_\rho]$. Here we make the performance measure explicitly depend on the traffic intensity ρ to expose the joint impact of dependence in flows and the traffic intensity on it. Under regularity conditions, the workload $Z(t)$ converges to the steady-state workload Z_ρ as t increases to infinity. In [18] it was shown that the IDW I_w is intimately related to a scaled mean workload $c_Z^2(\rho)$, defined by

$$c_Z^2(\rho) \equiv \frac{E[Z_\rho]}{E[Z_\rho; M/D/1]}, \tag{5}$$

where $E[Z_\rho; M/D/1]$ is the mean steady-state workload in a M/D/1 model given by

$$E[Z_\rho; M/D/1] = \frac{E[V_1]\rho}{2(1-\rho)}. \tag{6}$$

As (6) suggests, the mean steady-state workload converges to 0 as $\rho \downarrow 0$ and diverges to infinity as $\rho \uparrow 1$. The normalization in (5) exposes the impact of variability separately from the traffic intensity.

In great generality as discussed in [18], we have

$$c_Z^2(0) = 1 + c_s^2 = I_w(0) \quad \text{and} \quad c_Z^2(1) = c_A^2 + c_s^2 = I_w(\infty), \tag{7}$$

where c_A^2 is the asymptotic variability parameter, i.e., the normalization constant in the central limit theorem (CLT) for the arrival process; see §4 in [57] and §5 in the associated e-companion. For a renewal process, c_A^2 coincides with the scv c_a^2 of an interarrival time. The reference case is the classical $M/GI/1$ queue, for which we have

$$c_Z^2(\rho) = 1 + c_s^2 = I_w(t) \quad \text{for all} \quad \rho,\, t, \quad 0 < \rho < 1,\, t \geq 0.$$

The limits in (7) imply that, when c_A^2 is not nearly 1, $c_Z^2(\rho)$ varies significantly as a function of ρ. Hence, the impact of the variability in the arrival process upon the queue performance clearly depends on the traffic intensity. This important insight from [18] is the starting point for our analysis. In well-behaved models, $c_Z^2(\rho)$ as a function of ρ and $I_w(t)$ as a function of t tend to change smoothly and monotonically between those extremes, but OQNs can produce more complex behavior when both the traffic intensities at the queues and the levels of variability in the arrival and service processes at different queues vary; e.g., see the examples for queues in series in §§5.2, EC.8.2 and EC8.3 of [57].

2.1.2 Calculating the IDC from Models

For renewal processes, the variance $\mathrm{Var}(A(t))$ and thus the IDC $I_a(t)$ can either be calculated directly or can be characterized via their Laplace transforms and thus calculated by inverting those transforms or approximated by performing asymptotic analysis. Because we are interested in the steady-state behavior of the OQN, we are primarily interested in the equilibrium renewal process, as in §3.5 of [42].

In turns out that the variance of the equilibrium arrival renewal process $V(t) \equiv \mathrm{Var}(A(t))$ can be expressed in terms of the renewal function $m(t) \equiv E[A_0(t)]$, where A_0 is the corresponding ordinary renewal process. For a function f, let $\hat{f}$ denote the Laplace transform of f, defined by

$$\hat{f}(s) \equiv \mathcal{L}(f)(s) \equiv \int_0^\infty e^{-st} f(t)dt.$$

The following formula is taken from §2 of [56]

$$\hat{V}(s) = \frac{\lambda}{s^2} + \frac{2\lambda}{s}\hat{m}(s) - \frac{2\lambda^2}{s^3} = \frac{\lambda}{s^2} + \frac{2\lambda}{s}\frac{\hat{g}(s)}{s\,(1-\hat{g}(s))} - \frac{2\lambda^2}{s^3}, \tag{8}$$

where g is the density function of the interarrival-time distribution. The variance function can then be obtained numerically, which is discussed in §13 of [1]. The hyperexponential (H_2) and Erlang (E_2) special cases are described in §III.G of [18].

It is also possible to carry out similar analyses for much more complicated arrival processes. [38] applies matrix-analytic methods to give explicit representations of the variance $\mathrm{Var}(A(t))$ for the versatile Markovian point process or Neuts process; see §5.4, especially Theorem 5.4.1. Explicit formulas for the Markov modulated Poisson process (MMPP) are given on pp. 287-289.

2.1.3 Estimating the IDC from Data

Now we present an algorithm to numerically estimate the variance $V(t) = \mathrm{Var}(A(t))$ from a given realized sample path of the stationary point process $A(t)$. The main idea is based on Section 5.4 (iii) of [9].

Our goal is to estimate $V(t)$ for $0 < t < t_0$ using a realization of $A(t)$ for $0 < t < T$. The simplest way is to apply crude Monte Carlo method to estimate $V(t)$ for a fixed t and repeat over a finite grid of t's. This method divides the sample path of $A(t)$ into non-overlapping intervals of length t and counts the number of arrivals in each interval. The variance is then estimated by the sample variance of the counts. This method is simple to implement but can be slow to converge.

To accelerate the crude Monte Carlo method, we apply three techniques: (i) we use overlapping intervals instead of non-overlapping ones, which introduces bias but reduces sample variance; (ii) we calculate $V(t)$ only over a finite grid equally spaced in the logarithm scale instead of the linear scale; and (iii) we re-use the tallied number of events for shorter intervals to calculate the total number of events for longer interval, which avoids repetitive counting. We discuss the three techniques in turn:

Remark 2 (justifying the logarithmic scale) To justify the logarithm scale in (ii), we remark that the IDC of most stationary processes converges exponentially fast to a constant, as the time t increases. In particular, this holds for Markov arrival processes, which includes hyperexponential renewal process, Erlang renewal process, and Markov modulated Poisson Process as special cases; e.g.. see Ch. XI of [2], [37] or [38]. □

To use overlapping intervals, consider first $k = T/t$ non-overlapping intervals, each with length t. Now, we further divide each intervals of length t in to r intervals of the same length $\tau = t/r$. Hence we have rk number of non-overlapping intervals of length τ. Let n_i be the number of events fall in the i-th interval, consider

$$U_i \equiv A(I_i) \equiv A[i\tau, (i+r)\tau) = n_i + n_{i+1} + \cdots + n_{i+r-1}, \quad i = 0, 1, \ldots, rk - r + 1.$$

We estimate $V(t)$ with the sample variance $\bar{V}_l$ of $\{U_i\}_{i=1}^l$, where $l = rk - r + 1$. This estimator is in general biased but can achieve lower variance compared with the one obtained with crude Monte Carlo method. In §3 of the appendix we show that this estimator of $V(t)$ is asymptotically consistent under mild conditions that $V(t)$ is differentiable with derivative $\dot{V}(t)$ having finite positive limits as $t \to \infty$.

For the third technique, we now present a algorithm to simultaneously estimate $V(2^i\tau)$ for some $\tau > 0$ and $i = 0, 1, \dots, l$. Let $\{I_i\}$ be the collection of non-overlapping intervals of length τ that covers $[0, T]$. Let $n_i = A(I_i)$ be the number of events on interval I_i. Then we have the following table from [9].

	time horizon t			
sample	τ	2τ	$2^2\tau$	$\cdots$
1	n_1	$n_1 + n_2$	$n_1 + n_2 + n_3 + n_4$	$\cdots$
2	n_2	$n_2 + n_3$	$n_3 + n_4 + n_5 + n_6$	$\cdots$
3	n_3	$n_3 + n_4$	$n_5 + n_6 + n_7 + n_8$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

We find the estimation of $V(2^i\tau)$ by calculating the sample variance of the corresponding column.

Now that we have a efficient algorithm to estimate $V(2^i\tau)$ for fixed τ, we have obtained the estimations of a grid equally spaced in logarithm scale. To obtain estimations for finer grids we shift the crude grid by picking several $\tau \le \tau_j \le 2\tau$ equally spaced in log scale and, for each j, simultaneously estimate $V(2^i\tau_j)$ for all i.

2.2 Robust Queueing for Single-Server Queues

In this section, we review the RQ algorithm for single-server queues and discuss approximations for other performance measures obtained as a result. The RQ algorithm serves as a bridge between the IDC of the arrival process and the approximations of the performance measures. In particular, as in (13), the RQ algorithm generates approximation of the steady-state workload for any queue using the IDC of the total arrival process at that queue.

Consider the $G/GI/1$ queue, where the arrival process is a stationary and ergodic point process and the service times are i.i.d., independent of the arrival process. We assume that the arrival process A is partially characterized by the arrival rate λ and the IDC I_a defined in (1). For a stationary point process, we always have $E[A(t)] = \lambda t$; see §2.7 of [45]. We further assume that the service time distribution has finite mean $1/\mu$ (and thus rate μ) and scv c_s^2. We also assume that $\rho \equiv \lambda/\mu < 1$ for model stability. Let Z be the steady-state workload in the $G/GI/1$ model. The RQ algorithm provides approximation for $E[Z]$ with $(\lambda, I_a, \mu, c_s^2)$ as input data.

To obtain the RQ algorithm, we start with a reverse-time construction of the workload process as in §3 of [57]. Define the net-input process $N(t)$ as

$$N(t) \equiv Y(t) - t, \quad t \geq 0. \tag{9}$$

Then the workload at time t, starting empty at time 0, is obtained from the reflection map Ψ applied to N, i.e.,

$$Z = \Psi(N)(t) \equiv N(t) - \inf_{0 \leq s \leq t} \{N(s)\}, \quad t \geq 0. \tag{10}$$

With a slight abuse of notation, let $Z(t)$ be the workload at time 0 of a system that started empty at time $-t$. Then $Z(t)$ can be represented as

$$Z(t) \equiv \sup_{0 \leq s \leq t} \{N(s)\}, \quad t \geq 0, \tag{11}$$

where N is defined in terms of Y as before, but Y is interpreted as the total work in service time to enter over the interval $[-s, 0]$. That is achieved by letting V_k be the k^{th} service time indexed going backwards from time 0 and $A(s)$ counting the number of arrivals in the interval $[-s, 0]$.

The workload process $Z(t)$ defined in (11) is nondecreasing in t and hence necessarily converges to a limit Z. For the stable stationary $G/GI/1$ model, Z corresponds to the steady-state workload and satisfies $P(Z < \infty) = 1$; see §6.3 of [45].

In the ordinary stochastic queueing model, $N(s)$ is a stochastic process and hence $Z(t)$ is a random variable. However, in Robust Queueing practice, $N(s)$ is viewed as a deterministic instance drawn from a pre-determined uncertainty set $\mathcal{U}$ of input functions, while the workload Z^* for a Robust Queue is regarded as the worst case workload over the uncertainty set, i.e.

$$Z^* \equiv \sup_{\tilde{N} \in \mathcal{U}} \sup_{x \geq 0} \{\tilde{N}(x)\}.$$

Following the setting from [57], we adopt the following uncertainty set motivated from central limit theorem (CLT)

$$\mathcal{U} \equiv \left\{ \tilde{N} : \mathbb{R}^+ \to \mathbb{R} : \tilde{N}(s) \leq E[N(s)] + b\sqrt{\text{Var}(N(s))},\ s \geq 0 \right\}, \tag{12}$$

where $N(t)$ is the net input process associated with the stochastic queue, so

$$E[N(t)] = E[Y(t) - t] = \rho t - t,$$
$$\text{Var}(N(t)) = \text{Var}(Y(t)) = I_w(t)E[V_1]E[Y(t)] = (I_a(t) + c_s^2)\rho t/\mu.$$

The RQ approximation based on this partial model characterization is

$$E[Z_\rho] \approx Z_\rho^* \equiv \sup_{\tilde{N}_\rho \in \mathcal{U}_\rho} \sup_{x\geq 0} \{\tilde{N}(x)\} = \sup_{x\geq 0} \{-(1-\rho)x + b\sqrt{\rho x(I_a(x)+c_s^2)/\mu}\}, \qquad (13)$$

which follows Theorem 2 of [57] and (4). Notice that the approximation in (13) is directly a supremum of a real-valued function, and so can be computed quite easily for any given 4-tuple $(\lambda, I_a, \mu, c_s^2)$.

Theorem 5 in [57] states that the RQ algorithm gives asymptotically exact values of the mean steady-state workload in both light-traffic and heavy-traffic limits. Through extensive simulation experiments, it has been found that the mean steady-state workload $E[Z]$ can be well approximated by the IDW-based RQ algorithm.

Remark 3 (Continuous-time stationarity) We emphasize that, in the RQ formulation, it is essential to use the continuous-time stationary version of the IDC in (1) and the IDW in (3), instead of their discrete-time Palm stationary versions; see [45] for a comprehensive discussion. The continuous-time stationary IDC we use here yields asymptotically correct light-traffic limit, whereas the Palm stationary IDC does not; see §5.2 of [57]. □

Remark 4 (Queue length and waiting time) Approximations for other steady-state performance measures can be obtained by applying exact relations for the $G/GI/1$ queue that follow from Little's law $L = \lambda W$ and its generalization $H = \lambda G$; e.g., see [53] and Chapter X of [2] for the $GI/GI/1$ special case. Let W, Q and X be the steady-state waiting time, queue length and the number in system (including the one in service, if any). By Little's law,

$$E[Q] = \lambda E[W] = \rho E[W] \quad \text{and}$$
$$E[X] = E[Q] + \rho = \rho(E[W]+1).$$

By Brumelle's formula [6] or $H = \lambda G$, (6.20) of [53],

$$E[Z] = \rho E[W] + \rho\frac{E[V^2]}{2\mu} = \rho E[W] + \rho\frac{(c_s^2+1)}{2\mu}.$$

Hence, given an approximation Z^* for $E[Z]$, we can use the approximations

$$E[W] \approx \max\{0, Z^*/\rho - (c_s^2+1)/2\mu\} \quad \text{and}$$
$$E[Q] \approx \lambda E[W].$$

Remark 5 (Network performance measures) So far we only have discussed the performance measures for a single station. The total network performance measures, on the other hand, can also be derived. For example, the expected value of the total sojourn time T_i^{tot}, i.e. the time needed to flow through the queueing network for a customer that enters the system from station i, is easily estimated from the obtained mean waiting time at each station. Assuming Markov routing with routing matrix P, a standard argument from discrete time Markov chain theory gives the mean total number of visits $\xi_{i,j}$ to station j by a customer entering the system at station i as

$$\xi_{i,j} = \left((I - P)^{-1}\right)_{i,j},$$

where $(I - P)^{-1}$ is the fundamental matrix of a absorbing Markov chain. Hence, the mean steady-state total sojourn time $E[T_i^{\text{tot}}]$ is approximated by

$$E[T_i^{\text{tot}}] \approx \sum_{j=1}^{K} \xi_{i,j}(E[W_j] + 1/\mu_j). \tag{14}$$

In real world applications, customers often experiences non-Markovian routing, where routes are customer-dependent. For ways to represent those scenarios and convert them (approximately) to the current framework, see §2.3 and §6 of [50]. □

3 Approximating the IDCs of the Network Flows

In the i.i.d. service time setting, the IDW reduces to the arrival IDC plus the service scv as in (4). To generalize the RQ algorithm in §2.2 into a RQNA algorithm for networks, the main challenge is developing a successful approximation for the IDC of the total arrival flow at each queue.

In this section we develop a framework for approximating the IDCs of the network flows in the OQN, including the total arrival flows. We start in §3.1 by reviewing the OQN model and the required model data for the RQNA algorithm. We review the standard traffic rate equations in §3.2 and develop the new IDC equations in §3.3.

3.1 The OQN Model

3.1.1 The Model Primitives

We consider a network of K queues. Each queue has a single server, unlimited waiting space and provides service in order of arrival.

For each queue i, $1 \leq i \leq K$, we have an external arrival process $A_{0,i} \equiv \{A_{0,i}(t) : t \geq 0\}$. Each external arrival process $A_{0,i}$ is assumed to be a simple (no batches) stationary and ergodic point process with finite rate $\lambda_{0,i}$ and finite second-moment process $E[A_{0,i}^2(t)]$. We assume that all these external arrival processes, as well as the service and routing processes, are mutually independent.

For each individual queue, we assume that the service times are i.i.d. Let V_i^l denote the service requirement of the l-th customer at queue i, which we assume to be distributed according to cdf G_i with finite mean $1/\mu_i$ and scv $c_{s,i}^2$. Let the associated service renewal counting process be $S_i \equiv \{S_i(t) : t \geq 0\}$, where

$$S_i(t) = \max\left\{ n \leq 0 : \sum_{l=1}^{n} V_i^l \leq t \right\}, \quad t \geq 0. \tag{15}$$

We assume that departures are routed from node to node and out of the network by Markovian routing, which is independent of the arrival and service processes. We assume that each arrival eventually leaves w.p.1. Let $p_{i,j}$ denote the probability that a departure from node i is routed to node j. Let $P \equiv \{p_{i,j} : 1 \leq i, j \leq K\}$ be the (substochastic) routing matrix. Furthermore, let $p_{i,0} \equiv 1 - \sum_j p_{i,j}$ denote the probability that a customer departs the system after completing service at from node i.

3.1.2 The IDC's of the Flows

In order to apply the RQ algorithm, our primary focus here is to analyze and approximate the IDC's of the customer flows in a OQN. The flows can be separated into two groups, the *external flows* and the *internal flows.* The external flows are the flows associated with the model primitives in §3.1.1. For external arrival process $A_{0,i}$, we let $I_{a,0,i} \equiv \{I_{a,0,i}(t) : 0 \leq t \leq \infty\}$ denote the its IDC, as defined in (1). For service flows, let $I_{s,i} \equiv \{I_{s,i}(t); 0 \leq t \leq \infty\}$ be the IDC of the stationary renewal process associated with (15). For the case of renewal process, we necessarily have $I_{s,i}(\infty) = c_{s,i}^2$. We assume that the IDC's $I_{a,0,i}$ and $I_{s,i}$ are continuous functions with limits at 0 and $+\infty$.

The IDC's of the external flows forms an important part of the model input of our RQNA algorithm. In particular, we assume that we are given $(\lambda_{0,i}, I_{a,0,i}, \mu_i, I_{s,i})$ for each queue i and the routing matrix P.

In practice, the IDC of the external flows can be specified by one of the following ways. First, for renewal processes, it suffices to specify the inter-renewal-time cdf; then the associated IDC can be computed from the cdf as indicated in §2.1.2. If we are only given the first two moments of the inter-renewal-time cdf, then we can fit a convenient cdf the inter-renewal-time cdf to these parameters as indicated in §3 of [49], and use the previous method. Similar methods apply to non-renewal arrival process models, as indicated in §2.1.2. Finally, if we are only give sample data of the process, then we apply the numerical algorithm in §2.1.3 to estimate the rate and IDC of the process.

To implement our IDC approximations, we develop approximations for the IDC's of the internal flows. We use the following notation: Let A_i denote the total arrival process at queue i and let $I_{a,i}$ be the associated IDC; let D_i denote the departure process at queue i and let $I_{d,i}$ be the associated IDC; and let $A_{i,j}$ denote the departing customer flow from queue i that are routed to queue j and let $I_{a,i,j}$ be the associated IDC.

3.2 The Traffic Rate Equations and Traffic Intensities

Let $\lambda \equiv (\lambda_1, \ldots, \lambda_K)$ be the effective (total) arrival rate vector. We use the same traffic rate equations as in a Jackson network to determine λ. Then $\lambda_{i,j} \equiv \lambda_i p_{i,j}$ is the rate of the internal arrival flow $A_{i,j}$. Recall that $\lambda_0 \equiv (\lambda_{0,1}, \ldots, \lambda_{0,K})$ is the external arrival rate vector, then the traffic-rate equations are

$$\lambda_i = \lambda_{0,i} + \sum_{j=1}^{K} \lambda_{j,i} = \lambda_{0,i} + \sum_{i=1}^{K} \lambda_j p_{j,i}, \quad 1 \leq i \leq K, \tag{16}$$

or in matrix form

$$(I - P')\lambda = \lambda_0,$$

where I denotes the $K \times K$ identity matrix. We assume that $I - P'$ is invertible; i.e., we assume that all customers eventually leave the system. The condition for the invertibility of $I - P'$ to hold is well known, e.g. in Theorem 3.2.1 of [29]. Hence, the vector of internal arrival rates is given by

$$\lambda = (I - P')^{-1}\lambda_0. \tag{17}$$

Then the traffic intensity at queue i is defined as usual by $\rho_i \equiv \lambda_i/\mu_i$. We assume that $\rho_i < 1$ for all i to ensure that the OQN is stable.

3.3 The Traffic Variability Equations

In this section, we develop a set of IDC equations to solve for the approximations of the IDC's of the internal flows. The IDC of the total arrival process at each queue is then converted into approximations of the performances measures as in §2.2.

As in other decomposition methods, three network operations are essential: the departure operation (flow through a queue), the splitting operation (devide a flow into several sub-flows) and the superposition operation (combining multiple flows). We develop IDC equations that reveal (approximately) how the IDC's evolve under each network operation.

3.3.1 The Departure Operation

The IDC of the stationary departure process has been studied in §6.2 of [56]. We briefly review the departure IDC equation, see §5.1 of the appendix for more details.

We approximate the IDC $I_{d,i}$ by a convex combination of the arrival IDC $I_{a,i}$ and the service IDC $I_{s,i}$. In particular,

$$I_{d,i}(t) \approx w_i(t) I_{a,i}(t) + (1 - w_i(t)) I_{s,i}(\rho_i t), \quad t \geq 0. \tag{18}$$

The weight function w_i is defined as

$$w_i(t) \equiv w^* \left((1 - \rho_i)^2 \lambda_i t / \rho_i c_{x,i}^2 \right), \quad t \geq 0, \tag{19}$$

where $c_{x,i}^2 \equiv c_{a,i}^2 + c_{s,i}^2$ and $c_{a,i}^2 = I_{a,i}(\infty)$ and the *canonical weight function* w^* is

$$w^*(t) = \frac{1}{2t} \left(\left(t^2 + 2t - 1 \right) \left(1 - 2\Phi^c(\sqrt{t}) \right) + 2\phi(\sqrt{t})\sqrt{t}\,(1+t) - t^2 \right) \tag{20}$$

Note that there is a change of notation between (18) here and (74) in [56]. In particular, we have $I_{s,i}(\rho_i t)$ here instead of $I_{s,i}(t)$. In [56], we worked with a single-server queue and assumed that $I_{s,i}(t)$ is the IDC associated with the rate-λ_i service process. However, when considering a OQN here, it is natural to work with service IDC that associated with the service rate μ_i. These two approaches are equivalent, as we observed in Remark 1. Given that the given stationary service process has rate μ_i, we convert it to rate λ_i by considering $I_{s,i}(\rho_i t)$.

Remark 6 (Parallel to QNA in [50].) The convex combination in the approximation (18) is reminiscent of the convex combination for variability parameters in (38) of [50], i.e.,

$$c_{d,i} \approx (1 - \rho_i^2) c_{a,i}^2 + \rho_i^2 c_{s,i}^2, \tag{21}$$

which corresponds to a stationary-interval approximation, as discussed in [49, 50, 51].

Similar behavior can be seen in approximation (18). In particular, the canonical weight function w^* in (20) is a monotonically increasing function with $w^*(0) = 0$ and $w^*(\infty) = 1$. By the definition of $w_i(t)$, we see that for each t, (18) places less weight on $I_{a,i}(t)$ and more weight on $I_{s,i}(t)$ as ρ_i increases. This makes sense intuitively, because the queue should be busy most of the time as ρ_i increases toward 1. Thus departure times tend to be minor variations of service times. In contrast, if ρ_i is very small, then the queue acts only as a minor perturbation of the arrival process.

However, (19) reveals a more subtle interaction between ρ_i and the variability of the departure process over different time scales. □

3.3.2 The Splitting Operation

To treat splitting, we write the split process $A_{i,j}$ as a random sum. Let $\theta_{i,j}^l = 1$ if the l-th departure from queue i is directed to queue j, and let $\theta_{i,j}^l = 0$ if otherwise. Then observe that

$$A_{i,j}(t) = \sum_{l=1}^{D_i(t)} \theta_{i,j}^l, \quad t \geq 0.$$

We apply the conditional-variance formula to write the variance $V_{a,i,j}(t) \equiv \mathrm{Var}(A_{i,j}(t))$ as

$$V_{a,i,j}(t) \quad = \quad E[\mathrm{Var}(A_{i,j}(t)|D_i(t))] + Var(E[A_{i,j}(t)|D_i(t)]). \tag{22}$$

With the Markovian routing we have assumed, the routing decisions at each queue at each time are i.i.d. and independent of the history of the network. As a consequence, for feed-forward queueing networks, we can deduce that the collection of all routing decisions made at queue i up to time t is independent of $D_i(t)$. For the case in which independence holds, we can apply (22) to express $V_{a,i,j}(t)$ in terms of the variance of the departure process, $V_{d,i}(t) \equiv \mathrm{Var}(D_i(t))$; in particular,

$$V_{a,i,j}(t) = p_{i,j}^2 V_{d,i}(t) + p_{i,j}(1 - p_{i,j})\lambda_i t, \tag{23}$$

or, equivalently, since $E[D_i(t)] = \lambda_i t$ and $E[A_{i,j}(t)] = p_{i,j}\lambda_i t = p_{i,j}E[D_i(t)]$,

$$I_{a,i,j}(t) = p_{i,j} I_{d,i}(t) + (1 - p_{i,j}). \tag{24}$$

The formula (24) is an initial approximation, which parallels the approximation used for splitting in (40) of [50], i.e., $c_{a,i,j}^2 = p_{i,j}c_{d,i}^2 + (1 - p_{i,j})$.

However, the independence assumption will not hold in the presence of customer feedback, in which case there is a complicated dependence. We develop a more general formula to improve the approximation in general OQNs.

For that purpose, we apply the FCLT for split processes in §9.5 of [55] and the heavy-traffic limit theorems in [60]. We give the detailed derivation in §5.2 of the appendix.

Based on that heavy-traffic analysis, we propose the splitting IDC equation as

$$I_{a,i,j}(t) = p_{i,j} I_{d,i}(t) + (1 - p_{i,j}) + \alpha_{i,j}(t). \tag{25}$$

To account for the dependence, we include a correction term $\alpha_{i,j}$, defined as

$$\begin{aligned} \alpha_{i,j,\rho_i}(t) &\approx 2\xi_{i,j} p_{i,j}(1 - p_{i,j}) w_{\rho_i}(t) \\ &= 2\xi_{i,j} p_{i,j}(1 - p_{i,j}) w^*((1-\rho_i)^{-2}\lambda_i t/(h(\rho_i)c_{x,i}^2)), \quad t \geq 0, \end{aligned} \tag{26}$$

where $w_{\rho_i}(t)$ is the weight function for the departure IDC in (19), $c_{x,i}^2$, $c_{a,i}^2$ and $c_{s,i}^2$ are also as in (19), while $\xi_{i,j}$ is the $(i,j)^{\text{th}}$ entry of the matrix $(I - P')^{-1}$.

3.3.3 The Superposition Operation

In this section, we investigate the impact of the superposition operation on the IDC's. To start, consider the case in which the individual streams are mutually independent. In this case, we have

$$V_{a,i}(t) \equiv \text{Var}(A_i(t)) = \text{Var}\left(\sum_{j=0}^{K} A_{j,i}(t)\right) = \sum_{j=0}^{K} \text{Var}(A_{j,i}(t)),$$

so that

$$I_{a,i}(t) = \sum_{j=0}^{K} (\lambda_{j,i}/\lambda_i) I_{a,j,i}(t), \tag{27}$$

where $I_{a,j,i}(t) \equiv \text{Var}(A_{j,i}(t))/E[A_{j,i}(t)]$. Recall that (27) differs from (36) of [57] because we are not assuming rate-1 processes in our definitons of the IDC; see Remark 1.

While (27) is exact when the streams are independent, it is not exact in general cases. Even for feed-forward networks, we may have a stream that splits and then recombines later, which introduces dependence.

For dependent streams, the variance of the superposition total arrival process at queue i can be written as

$$V_{a,i}(t) \equiv \text{Var}\left(\sum_{j=0}^{K} A_{j,i}(t)\right) = \sum_{j=0}^{K} \text{Var}\left(A_{j,i}(t)\right) + \beta_i(t) E[A_i(t)]$$

where $A_{0,i}$ denotes the external arrival process at station i,

$$\beta_i(t) \equiv \sum_{j \neq k} \beta_{j,i;k,i}(t), \quad \text{and} \quad \beta_{j,i;k,i}(t) \equiv \frac{\text{cov}\left(A_{j,i}(t), A_{k,i}(t)\right)}{E[A_i(t)]}. \tag{28}$$

In terms of the IDC's, we have

$$I_{a_i}(t) = \sum_{j=0}^{K} (\lambda_{j,i}/\lambda_i) I_{a_{j,i}}(t) + \beta_i(t). \tag{29}$$

In general, an exact characterization of the correction term $\beta_i(t)$ is not available. Thus, we again apply heavy-traffic limits in [60] to generate an approximation. Detailed derivation appears in §5.3 of the appendix.

Assume without loss of generality that $\rho_j \geq \rho_i$. From the heavy-traffic analysis, we obtain the approximation

$$\beta_{j,i;k,i}(t) = \beta_{k,i;j,i}(t) \approx (\zeta_{j,i;k,i}/\lambda_i) w^*((1-\rho_j)^2 p_{j,i}\lambda_j t/\rho_j c^2_{x,j,i}), \tag{30}$$

where w^* is the weight function in (20), $c^2_{x,j,i} = p_{j,i}c^2_{a,j} + (1-p_{j,i}) + p_{j,i}c^2_{s,j}$ and $c^2_{a,j}$ is solved from the variability equations for the asymptotic variability parameters in (35). The constant $\zeta_{j,i;k,i}$ is defined as

$$\zeta_{j,i;k,i} = \nu'_j \left(\operatorname{diag}(c^2_{a,0,i}\lambda_i) + \sum_{l=1}^{K} \Sigma_l \right) \nu_k + \nu'_k \Sigma_j e_i + \nu'_j \Sigma_k e_i, \tag{31}$$

where $\nu_l \equiv p_{l,i} e'_l (I - P')^{-1}$ for $l = j, k$, e_i is the i-th unit vector, $\operatorname{diag}(c^2_{a,0,i}\lambda_i)$ is the diagonal matrix with $c^2_{a,0,i}\lambda_i$ as the i-th diagonal entry, Σ_l is the covaraince matrix of the splitting decision process at station l defined as $\Sigma_l \equiv (\sigma^l_{i,j})$ with $\sigma^l_{i,i} = p_{l,i}(1-p_{l,i})\lambda_l$ and $\sigma^l_{i,j} = -p_{l,i}p_{l,j}\lambda_l$ for $i \neq j$.

3.4 The IDC Equation System

We now assemble the building blocks into a system of linear equations (for each t) that describes the IDC's in the OQN. Combining (18), (25) and (29), we obtain *the IDC equations*. These are equations that should be satisfied by the unknown IDCs. For $1 \leq i \leq K$, the equations are

$$\begin{aligned} I_{a,i}(t) &= \sum_{j=1}^{K} (\lambda_{j,i}/\lambda_i) I_{a,j,i}(t) + (\lambda_{0,i}/\lambda_i) I_{a,0,i}(t) + \beta_i(t), \\ I_{a,i,j}(t) &= p_{i,j} I_{d,i}(t) + (1-p_{i,j}) + \alpha_{i,j}(t), \\ I_{d,i}(t) &= w_i(t) I_{a,i}(t) + (1-w_i(t)) I_{s,i}(\rho_i t). \end{aligned} \tag{32}$$

The parameters $p_{i,j}$, $\lambda_{i,j}$ and λ_i are determined by the model primitives in §3.1.1 and the traffic rate equations in §3.2. The IDC's of the external flows $I_{a_{0,i}}(t)$ and $I_{s_i}(t)$ are assumed to be calculated via exact or numerical inversion of Laplace Transforms, or estimated from data. The weight functions $w_i(t)$ is defined in (19), which involves a limiting variability parameter $c^2_{x,i} \equiv I_{a,i}(\infty) + c^2_{s,i}$.

To solve for the limiting variability parameters $I_{a,i}(\infty)$, we let $t \to \infty$ in (32) and denote $c^2_{a,i} \equiv I_{a,i}(\infty), c^2_{a,i,j} \equiv I_{a,i,j}(\infty)$ and $c^2_{d,i} \equiv I_{d,i}(\infty)$. Furthermore, we define

$$c^2_{\alpha_{i,j}} \equiv \alpha_{i,j}(\infty) = 2\xi_{i,j} p_{i,j}(1 - p_{i,j}),$$
$$c^2_{\beta_i} \equiv \beta_i(\infty) = \frac{2}{\lambda_i} \sum_{j<k} \zeta_{j,i;k,i},$$

where we used $w^*(\infty) = 1$ in (26) and (30). Hence, we have the *limiting variability equations*:

$$\begin{aligned} c^2_{a,i} &= \sum_{j=1}^{K} (\lambda_{j,i}/\lambda_i) c^2_{a,j,i} + (\lambda_{0,i}/\lambda_i) c^2_{a,0,i} + c^2_{\beta_i}, \\ c^2_{a,i,j} &= p_{i,j} c^2_{d,i} + (1 - p_{i,j}) + c^2_{\alpha_{i,j}}, \\ c^2_{d,i} &= c^2_{a,i}, \quad 1 \le i \le K. \end{aligned} \tag{33}$$

where we used the fact that $w_i(t) \to 1$ as $t \to \infty$.

For a concise matrix notation, let

$$\begin{aligned} \mathbf{I}(t) &\equiv (I_{a,1}(t), \ldots, I_{a,K}(t), I_{a,1,1}(t), \ldots, I_{a,K,K}(t), I_{d,1}(t), \ldots, I_{d,K}(t)), \\ \mathbf{b}(t) &\equiv (b_{a,1}(t), \ldots, b_{a,K}(t), b_{a,1,1}(t), \ldots, b_{a,K,K}(t), b_{d,1}(t), \ldots, b_{d,K}(t)), \\ \mathbf{M}(t) &\equiv (M_{m,n}(t)) \in \mathbb{R}^{(2K+K^2)^2}, \quad m, n \in \{a_1, \ldots, a_K, a_{1,1}, \ldots, a_{K,K}, d_1, \ldots, d_K\}, \\ \mathbf{c}^2 &\equiv (c^2_{a,1}, \ldots, c^2_{a,K}, c^2_{a,1,1}, \ldots, c^2_{a,K,K}, c^2_{d,i}, \ldots, c^2_{d,K}), \end{aligned}$$

where

$$\begin{aligned} & b_{a,i}(t) \equiv \frac{\lambda_{0,i}}{\lambda_i} I_{a,0,i}(t) + \beta_i(t), \quad b_{a,i,j} \equiv (1 - p_{i,j}) + \alpha_{i,j}(t), \\ & b_{d,i}(t) \equiv (1 - w_i(t)) I_{s,i}(t); \quad M_{a_i, a_{j,i}(t)} = \frac{\lambda_{j,i}}{\lambda_i}, \\ & M_{a_{i,j}, d_i}(t) = p_{i,j}, M_{d_i, a_i}(t) = w_i(t), \quad \text{and} \quad M_{m,n}(t) = 0 \text{ otherwise}. \end{aligned}$$

Then the IDC equations can be expressed concisely as

$$(\mathbf{E} - \mathbf{M}(t))\mathbf{I}(t) = \mathbf{b}(t), \tag{34}$$

while the limiting variability equations can be expressed as

$$(\mathbf{E} - \mathbf{M}(\infty))\mathbf{c}^2 = \mathbf{b}(\infty), \tag{35}$$

where $\mathbf{E} \in \mathbb{R}^{(2K+K^2)^2}$ is the identity matrix.

The following theorem states that these equations have unique solutions.

Theorem 1 *Assume that* $I - P'$ *is invertible. Then* $\mathbf{E} - \mathbf{M}(t)$ *is invertible for each fixed* $t \in \mathbb{R}^+ \cup \{\infty\}$*. Hence, for any given* t *and* $\mathbf{b}$*, the IDC equations in* (34) *have the unique solution*

$$\mathbf{I}(t) = (\mathbf{E} - \mathbf{M}(t))^{-1}\mathbf{b}(t)$$

and the limiting variability equations in (35) *have the unique solution*

$$\mathbf{c}^2 = (\mathbf{E} - \mathbf{M}(\infty))^{-1}\mathbf{b}(\infty).$$

Proof. Let $\delta_{i,j}$ be the Kronecker delta function. Then substituting the equations for $I_{a,j,i}(t)$ and $I_{d,i}(t)$ into the equation for $I_{a,i}(t)$, we obtain an equation set for $I_{a,i}(t)$ with coefficient matrix $(\delta_{i,j} - (\lambda_{j,i}/\lambda_i)p_{j,i}w_j(t)) \in \mathbb{R}^{K^2}$. Note that $(\lambda_{j,i}/\lambda_i)w_j(t) \leq 1$ for $t \in \mathbb{R}^+ \cup \{\infty\}$, the invertibility of $I - P'$ implies that the equations for $I_{a,i}(t)$ have an unique solution. Substituting in the solution for $I_{a,i}(t)$, we obtain solutions for $I_{a,i,j}(t)$ and $I_{d,i}(t)$. □

Remark 7 (The Kim [30, 31] MMPP(2) decomposition.) In Kim [30, 31], a decomposition approximation of queueing networks based on MMPP(2)/GI/1 queues was investigated. MMPP(2) stands for Markov modulated Poission process with 2 underlying states. The four rate parameters in the MMPP(2) are determined from the approximations of the mean, IDC and the third moment process of the arrival process at a pre-selected time t_0 and the limiting variability parameter of the arrival process. The IDC and third moment processes are approximated by the network equations with correction terms motivated from the Markovian routing settings.

At first glance, the IDC equations proposed here are quite similar to the network equations used in [30], see (20), (22) and (31) there. However, our method are different in three aspects. First, our approach does not fit the flows to special processes (MMPP in [30]), instead we partially characterize the flows by the IDC and apply the RQ algorithm reviewed in §2.2. Second, the entire IDC function is utilized in the RQ algorithm, whereas [30] used IDC evaluated at a pre-selected time t_0 to fit the parameters of the MMPP. Third, we rely on more detailed heavy-traffic limit to propose asymptotically exact correction terms, see §5.3 of the appendix. □

3.5 RQNA for Tree-Structured Queueing Networks

With the IDC equations developed in §3.4, we immediately obtain an elementary algorithm for tree-structured OQNs. A *tree-structured queueing network* is an OQN whose topology forms a directed tree. Recall that a directed tree is a connected directed graph whose underlying undirected graph is a tree. The tree-structured network is a special case of feed-forward network in which the superposed flows at each node have no common origin.

This special structure greatly simplifies the IDC-based RQNA algorithm. First, there is no customer feedback, which significantly simplify the IDC equations as well as the dependence in the queueing network. Second, for any internal flow $A_{i,j}$ that is non-zero, we must have $\alpha_{i,j} = 0$ for the correction term in (25), see discussions in §5.3 of the appendix. Finally, the tree structure implies that $\beta_i = 0$ for the correction term for superposition because all superposed processes are independent.

We summarize the procedure in Algorithm 1. To elaborate, with these simplifications of

---

Algorithm 1: The RQNA algorithm for approximating the IDC's at each time t in a tree-structured queueing network.

Require: The queueing network has tree structure.

Output : Solution to the IDC equations (34).

for $i = 1$ *to* n **do**

- $\lambda_i \leftarrow \lambda_{0,i} + \sum_{j<i} \lambda_j p_{j,i}$;
- $\rho_i \leftarrow \lambda_i/\mu_i$;
- $c_{a,i}^2 \leftarrow \sum_{j<i} \frac{\lambda_{j,i}}{\lambda_i} c_{a,j,i}^2 + \frac{\lambda_{0,i}}{\lambda_i} c_{a,0,i}^2$;
- $c_{x,i}^2 \leftarrow c_{a,i}^2 + c_{s,i}^2$;
- $w_i(t) \leftarrow w^*((1-\rho_i)^2 \lambda_i t/(\rho_i c_{x,i}^2))$;
- $I_{a_i}(t) \leftarrow \sum_{j<i} \frac{\lambda_{j,i}}{\lambda_i} (p_{j,i} (w_j(t) I_{a,j}(t) + (1 - w_j(t)) I_{s,j}(t)) + (1 - p_{j,i})) + \frac{\lambda_{0,i}}{\lambda_i} I_{a,0,i}(t)$;
- $I_{d_i}(t) \leftarrow w_i(t) I_{a,i}(t) + (1 - w_i(t)) I_{s,i}(t)$;
- **for** $j < i$ **do**
  - $I_{a,i,j}(t) \leftarrow p_{i,j} I_{d,i}(t) + (1 - p_{i,j})$;
- **end**

end

return $\mathbf{I}(t)$.

---

the correction terms, the equations in (32), yield, for $1 \le i, j \le K$,

$$
\begin{aligned}
I_{a_i}(t) &= \sum_{j=1}^{K} \frac{\lambda_{j,i}}{\lambda_i} I_{a_{j,i}}(t) + (\lambda_{0,i}/\lambda_i) I_{a_{0,i}}(t), \\
I_{a_{i,j}}(t) &= p_{i,j} I_{d_i}(t) + (1 - p_{i,j}), \\
I_{d_i}(t) &= w_i(t) I_{a_i}(t) + (1 - w_i(t)) I_{s_i}(t).
\end{aligned}
$$

The IDC equations in this setting inherit a special structure that allows a recursive algorithm. Note that the stations in the tree-structured network can be partitioned into disjoint layers $\{\mathcal{L}_1, \ldots, \mathcal{L}_l\}$ such that for station $i \in \mathcal{L}_k$, it takes only the input flows from $j \in \bigcup_{j=1}^{k-1} \mathcal{L}_j$ for $1 \leq k \leq l$. To simplify the notation, we sort the node in the order of their layers and assign arbitrary order to nodes within the same layer. If $i \in \mathcal{L}_k$, then $\bigcup_{j=1}^{k-1} \mathcal{L}_j \subset \{1, 2, \ldots, i-1\}$, so that $\lambda_{j,i} = 0$ for all $j \geq i$. Hence, by substituting in the equations for I_{d_i} and $I_{a_{i,j}}$ into that of I_{a_i}, we have

$$
\begin{aligned}
I_{a_i}(t) &= \sum_{j=1}^{K} \frac{\lambda_{j,i}}{\lambda_i} \left(p_{j,i} \left(w_j(t) I_{a_j}(t) + (1 - w_j(t)) I_{s_j}(t) \right) + (1 - p_{j,i}) \right) + \frac{\lambda_{0,i}}{\lambda_i} I_{a_{0,i}}(t), \\
&= \sum_{j<i} \frac{\lambda_{j,i}}{\lambda_i} \left(p_{j,i} \left(w_j(t) I_{a_j}(t) + (1 - w_j(t)) I_{s_j}(t) \right) + (1 - p_{j,i}) \right) + \frac{\lambda_{0,i}}{\lambda_i} I_{a_{0,i}}(t). \qquad (36)
\end{aligned}
$$

Note that (36) exhibits a lower-triangular shape so that we can explicitly write down the solution in the order of the stations.

4 Feedback Elimination

In this section, we discuss the case in which customers can return (feedback) to a queue after receiving service there. Customer feedback introduces dependence between the arrival process and the service times, even when the service times themselves are mutually independent. As a result, the decomposition $I_w(t) = I_a(t) + c_s^2$ in (4) is no longer valid. Indeed, assuming that it is, as we have done so far, can introduce serious errors, as we show in our simulation examples. We address this problem by introducing a feedback elimination procedure. We start with the so-called immediate feedback in §4.1 and generalize it into near-immediate feedback in §4.2.

4.1 Immediate Feedback Elimination

In Section III of [50] it is observed that it is often helpful to pre-process the model data by eliminating immediate feedback for queues with feedback. We now show how that can be done for the RQNA algorithm.

We consider a single queue with i.i.d. feedback. In this case, all feedback is *immediate feedback*, meaning that the customer feeds back to the same queue immediately after completing service, without first going through another service station. For a $GI/GI/1$ model allowing feedback, all feedback is necessarily immediate because there is only one queue.

Normally, the immediate feedback returns the customer back to the end of the queue. However, in the immediate feedback elimination procedure, the approximation step is to put the customer back at the head of the line so that the customer receives a geometrically random number of service times all at once. Clearly this does not alter the queue length process or the workload process, because the approximation step is work-conserving.

The modified system is a single-server queue with a new service-time distribution and without feedback. Let N_p denote a geometric random variable with success probability $1-p$ and support $\mathbb{N}^+$, the positive natural numbers, then the new service time can be expressed as

$$S_p = \sum_{i=1}^{N_p} S_i, \tag{37}$$

where S_i's are i.i.d. copies of the original service times. This modification in service times results in a change in the service scv. By the conditional variance formula, the scv of the total service time is $\tilde{c}_s^2 = p + (1-p)c_s^2$. The new service IDC in the modified system is the IDC of the stationary renewal process associated with the new service times. To obtain the new service IDC, we need only find the Laplace Transform of the new service distribution, then apply the algorithm in §2.1.2. We provides the details in §4 of the appendix.

For the mean waiting time, we need to adjust for per-visit waiting time by multiplying the waiting time in the modified system by $(1-p)$. Note that $(1-p)^{-1}$ is the mean number of visits by a customer in the original system.

In §4.1 of [60] it is shown that the modified system after the immediate feedback elimination procedure shares the same HT limits of the queue length process, the external departure process, the workload process and the waiting time process. Hence, the immediate feedback elimination procedure as an approximation is asymptotically exact in the heavy-traffic limit.

4.2 Near-Immediate Feedback

Now, we consider general OQNs, where the feedback does not necessarily happen immediately, meaning that a departing customer may visit other queues before coming back to the feedback queue. To treat general OQNs, we extend the immediate feedback concept to the *near-immediate feedback*, which depends on the traffic intensities of the queues on the path the customer took before feedback happens. The near-immediate feedback is defined as any feedback that does not go through any queue with higher traffic intensity.

By default, the RQNA algorithm eliminates all near-immediate feedback. To help understand near-immediate feedback, consider a modified OQN with one bottleneck queue, denoted by h. A *bottleneck queue* is a queue with the highest traffic intensity in the network. While all non-bottleneck queues have service times set to 0 so that they serve as instantaneous switches. In the reduced network, we define an external arrival $\hat{A}_0$ to the bottleneck queue to be any external arrival that arrive at the bottleneck queue for the first time. Hence, an external arrival may have visited one or multiple non-bottleneck queues before its first visit to the bottleneck queue. In particular, the external arrival process can be expressed as the superposition of (i) the original external arrival process $A_{0,h}$ at station h; and (ii) the Markov splitting of the external arrival process $A_{0,i}$ at station i with probability $\hat{p}_{i,h}$, for $i \neq h$, where $\hat{p}_{i,h}$ denote the probability of a customer that enters the original system at station i ends up visiting the bottleneck station h. For the explicit formula of $\hat{p}_{i,h}$, see Remark 3.2 of [60].

In §4.2 of [60], we showed that this reduced network is asymptotically equivalent in the HT limit to the single-server queue with i.i.d. feedback that we considered in §4.1. In particular, the arrival process of the equivalent single-station system is $\hat{A}_0$ as described above, the service times remain unchanged and the feedback probability is $\hat{p}$, which is exactly the probability of a near-immediate feedback in the original system; see (3.9) of [60] for the expression of $\hat{p}$. Hence we showed that eliminating all feedback at the bottleneck queue as described above prior to analysis is asymptotically correct in HT for OQNs with a single bottleneck queue in terms of the queue length process, the external departure process, the workload process and the waiting time process. Moreover, the different variants of the algorithm - eliminating all near immediate feedback or only the near-immediate feedback at the bottleneck queues - are asymptotically exact in the HT limit for an OQN with a single-bottleneck queue, because only the bottleneck queues have nondegenerate HT limit. In contrast, if there are multiple bottleneck queues, the HT limit requires multidimensional RBM, which is not used in our RQNA.

5 The Full RQNA Algorithm

As basic input parameters, the RQNA algorithm requires the model data specified in §3.1:

1. Network topology specified by the routing matrix P;
2. External arrival processes specified by (i) the interarrival distribution, if renewal; or (ii) rate λ and IDC; or (iii) a realized sample path of the stationary external arrival process;

3. Service renewal process specified by (i) the service distribution; or (ii) the rate and IDC; or (iii) a realized sample path of the stationary service renewal process.

Combining the traffic-rate equation, the limiting variability equation, the IDC equation and the feedback elimination procedure, we have obtained a general framework for the RQNA algorithm, which we summarize in Algorithm 2. We remark that the RQNA algorithm becomes much simpler in the case without customer feedbacks, as discussed in §3.5.

Algorithm 2: A general framework of the RQNA algorithm for the approximation of the system performance measures.

Require: Specification of the correction terms $\alpha_{i,j}(t)$ in §3.3.2 and $\beta_i(t)$ in §3.3.3, a set of stations to perform feedback elimination as specified in §4 and the flows to eliminate for each of the selected station.

Output : Approximation of the system performance measures.

1 Solve the traffic rate equations by $\lambda = (I - P')^{-1}\lambda_0$ as in §3.2 and let $\rho_i = \lambda_i/\mu_i$;

2 Solve the limiting variability equations by $\mathbf{c}^2 = (\mathbf{E} - \mathbf{M}(\infty))^{-1}\mathbf{b}(\infty)$ as in §3.4;

3 Solve the IDC equations by $\mathbf{I}(t) = (\mathbf{E} - \mathbf{M}(t))^{-1}\mathbf{b}(t)$ for the total arrival IDCs, where we use $\mathbf{c}$ from Step 2 in (19);

4 Select a set of stations to perform feedback elimination, as in §4. For each selected station, identify the flows to eliminate, then identify the corresponding feedback probability, the modified service IDC as in §4.1 as well as the reduced network. Repeat Step 1 to Step 3 on the reduced network to obtain the modified IDW (as the sum of the modified total arrival IDC and the modified service scv) at the selected station.

5 Apply the RQ algorithm in (13) to obtain the approximations for the mean steady-state workload at each station.

6 Apply the formulas in Remark 4 and 5 to obtain approximations for the expected values of the steady-state queue length and waiting time at each queue and the total sojourn time for the system.

Remark 8 (Computation complexity.) We remark that the full RQNA algorithm is light in computational complexity. Most of the calculation comes from Step 3 of the RQNA algorithm. For each t, the algorithm needs to solve for one linear systems with K equations, where K is the number of stations. By default, the algorithm solves these equations on a grid with points logarithmically apart, see Remark 2. For station i, the RQ algorithm requires the value of arrival IDC in the interval $[0, T_i]$ for $T_i = O((1-\rho_i)^{-2})$. Hence, RQNA solves for at most $O(-2\log(1-\rho_{\max}))$ linear systems, where $\rho_{\max} = \max_i \rho_i$. For each station that we apply feedback elimination, we need to run RQNA (without feedback elimination) on the reduced network. As a result, RQNA with feedback elimination solves for at most $O(-2K\log(1-\rho_{\max}))$ linear systems, each with at most K equations.

The general framework here allows different choices of (i) the correction terms $\alpha_{i,j}$ in §3.3.2 and β_i in §3.3.3 and (ii) the feedback elimination procedure. The default correction terms are given in (26) and (30). For the feedback elimination procedure, we apply near-immediate feedback elimination to all stations. In §6 of the appendix we discuss an additional tuning function to fine tune the performance of our RQNA algorithm.

6 Numerical Studies

In this section, we discuss examples of networks with significant near-immediate feedback from [11]. We show that the near-immediate feedback in these examples makes a big difference in the performance descriptions. Hence our predictions with and without feedback elimination are very different. We find that our RQNA with near-immediate feedback elimination performs as well or better than the other algorithms. Additional numerical examples appear in our previous papers and in §7 of the appendix.

6.1 A Three-Station Example

In this section, we look at the suite of three-station examples §3.1 of [11] depicted in Figure 1. This example is designed to have three stations that are tightly coupled with each other, so that the dependence among the queues and the flows is fairly complicated.

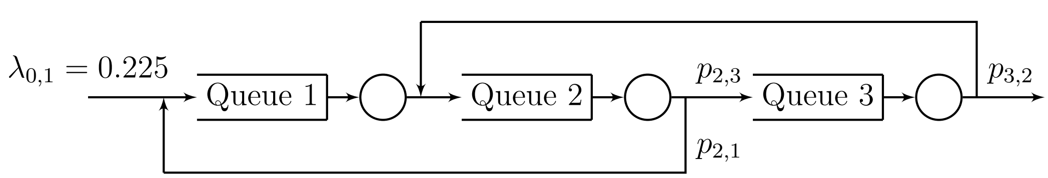

Figure 1: A three-station example.

In this example, we have three stations in tandem but also allow customer feedback from station 2 to station 1 and from station 3 to station 2, with probability $p_{2,1} = p_{2,3} = p_{3,2} = 0.5$.

The only external arrival process is a Poisson process which arrives at station 1 with rate $\lambda_{0,1} = 0.225$, hence by (16) the effective arrival rate is $\lambda_1 = 0.675$, $\lambda_2 = 0.9$ and $\lambda_3 = 0.45$.

For the service distributions, we consider the same sets of parameters as in [11], summarized in Table 1 and 2. Note that Case 2 is relatively more challenging because there are two bottlneck stations; in contrast, all the other cases have only one.

Table 1: Traffic intensity of the four cases in the three-station example.

Case	ρ_1	ρ_2	ρ_3
1	0.675	0.900	0.450
2	0.900	0.675	0.900
3	0.900	0.675	0.450
4	0.900	0.675	0.675

Table 2: Variability of the service distributions of the four cases in the three-station example.

Case	$c_{s,1}^2$	$c_{s,2}^2$	$c_{s,3}^2$
A	0.00	0.00	0.00
B	2.25	0.00	0.25
C	0.25	0.25	2.25
D	0.00	2.25	2.25
E	8.00	8.00	0.25

We now compare the RQNA approximations and four previous algorithms as in §7.3 of the appendix, with the simulated mean sojourn times at each station, as well as total sojourn time of the network. The sojourn time for each station is defined as the waiting time plus the service time at that station, whereas the total sojourn time of the network is defined as in (14). We consider two cases of the RQNA algorithm: (1) the plain RQNA algorithm without feedback elimination, as in Algorithm 2 and (2) the RQNA algorithm with feedback elimination, as discussed in §4.

For RQNA with feedback elimination, we apply feedback elimination to each station that has at least one feedback flow that only passes through stations with equal or lower traffic intensities. We eliminate all such flows in the feedback elimination procedure. Take Case 1 for example, we do not apply feedback elimination for Station 1 because all feedback customers go through Station 2, which has higher traffic intensity; we will, however, eliminate the flow from 2 to 1 as well as the flow from 3 to 2 for Station 2, since both Station 1 and 3 have lower traffic intensities. As another example, for both Station 2 and 3 in case 4, we eliminate the flow from 3 to 2, but we do not eliminate the flow from 2 to 1, since Station 2 and 3 share the same traffic intensity while Station 1 has higher traffic intensity.

Tables 3 and 4 expand Tables II and III in [11] by adding values for (1) the mean total sojourn time and (2) the RQ and RQNA approximations, with and without feedback elimination. For each table, we indicate by an asterisk in the last column the stations where

Table 3: A comparison of six approximation methods to simulation for the total sojourn time in the three-station example in Figure 1 with parameters specified in Table 1 and 2. In calculating the average absolute relative error, the diverging entry for QNET is ignored.

Case		Simulation	QNA	QNET	SBD	RQ	RQNA	RQNA (elim)
A	1	40.39 (3.75%)	20.5 (-49%)	diverging	43.0 (6.4%)	73.9 (83%)	83.5 (107%)	44.8 (11.0%)
	2	59.58 (3.29%)	36.0 (-40%)	56.7 (-4.9%)	58.2 (-2.4%)	78.0 (31%)	94.3 (58%)	69.3 (16.4%)
	3	40.72 (4.78%)	24.0 (-41%)	38.7 (-5.0%)	40.2 (-1.3%)	57.2 (41%)	74.7 (83%)	43.3 (6.3%)
	4	42.12 (3.36%)	26.2 (-38%)	41.8 (-0.7%)	42.7 (1.3%)	59.3 (41%)	75.1 (78%)	41.2 (-2.2%)
B	1	52.40 (2.64%)	42.0 (-20%)	52.6 (0.4%)	50.2 (-4.2%)	72.4 (38%)	93.7 (79%)	53.1 (1.4%)
	2	91.52 (3.77%)	94.1 (2.8%)	83.7 (-8.5%)	95.3 (4.1%)	109 (20%)	169 (85%)	94.5 (3.2%)
	3	61.68 (3.44%)	72.2 (17%)	61.9 (0.4%)	60.9 (-1.3%)	79.4 (29%)	133 (115%)	60.5 (-1.9%)
	4	63.34 (2.83%)	75.8 (20%)	64.1 (1.3%)	64.7 (2.1%)	83.0 (31%)	135 (113%)	62.4 (-1.4%)
C	1	44.24 (1.96%)	31.3 (-29%)	37.0 (-16%)	47.1 (6.4%)	75.7 (71%)	91.4 (106%)	42.1 (-4.8%)
	2	92.42 (4.23%)	87.4 (-5.4%)	91.2 (-1.4%)	91.6 (-0.83%)	106 (15%)	156 (68%)	96.0 (3.8%)
	3	44.26 (4.69%)	33.2 (-25%)	44.0 (-0.7%)	45.0 (1.7%)	61.3 (38%)	84.2 (90%)	44.0 (-0.6%)
	4	50.20 (1.04%)	41.4 (-18%)	51.1 (1.7%)	52.2 (4.0%)	67.4 (34%)	91.2 (82%)	45.9 (-8.6%)
E	1	134.4 (4.77%)	265 (97%)	155 (15%)	116 (-14%)	158 (17%)	305 (127%)	120 (-11%)
	2	213.1 (3.47%)	308 (45%)	228 (7.1%)	206 (-3.3%)	234 (10%)	367 (72%)	173 (-19%)
	3	138.7 (3.97%)	244 (76%)	161 (16%)	135 (-2.5%)	163 (17%)	300 (116%)	136 (-2.0%)
	4	155.1 (4.37%)	252 (63%)	168 (8.2%)	147 (-5.0%)	178 (15%)	312 (101%)	148 (-4.8%)
Average absolute relative error			36.63%	5.82%	3.80%	33.19%	92.50%	6.15%

elimination is applied.

We observed that the plain RQNA algorithm works well for stations with moderate to low traffic intensities, but not so satisfactory for congested stations. On the other hand, the accuracy of the RQNA algorithm with feedback elimination is on par with, if not better than the best previous algorithm.

6.2 A Ten-Station Example

We conclude with the 10-station OQN example with feedback considered in §3.5 of [11]. It is depicted here in Figure 2.

The only exogenous arrival process is Poission with rate 1. For each station, if there are two routing destinations, the departing customer follows Markovian routing with equal probability, each being 0.5. The vector of mean service times is $(0.45, 0.30, 0.90, 0.30, 0.38571, 0.20, 0.1333, 0.20, 0.15, 0.20)$, so that the traffic intensity vector is $(0.6, 0.4, 0.6, 0.9, 0.9, 0.6, 0.4, 0.6, 0.6, 0.4)$. The scv's at these stations are $(0.5, 2, 2, 0.25, 0.25, 2, 1, 2, 0.5, 0.5)$, where we assume a Erlang distribution if $c_s^2 < 1$, an exponential distribution if $c_s^2 = 1$ and a hyperexponential distribution if $c_s^2 > 1$.

In particular, note that stations 4 and 5 are bottleneck queues, having equal traffic

Table 4: A comparison of six approximation methods to simulation for the sojourn time at each station of the three-station example in Figure 1 for Case D in Table 1 and 2.

Case	Station	Simulation	QNA	QNET	SBD	RQ	RQNA	RQNA (elim)
D1	1	2.476 (0.61%)	2.24 (-9.4%)	2.48 (0.3%)	2.47 (-0.1%)	2.47 (-0.28%)	2.68 (7.8%)	2.68 (7.8%)
	2	10.85 (3.21%)	14.9 (37%)	11.6 (6.5%)	11.4 (5.2%)	19.8 (83%)	28.4 (162%)	11.1* (2.7%)
	3	2.544 (0.63%)	2.53 (-0.8%)	2.54 (-0.0%)	2.59 (1.6%)	2.57 (1.2%)	2.53 (-0.7%)	2.53 (-0.7%)
	Total	55.81 (2.58%)	71.4 (28%)	58.8 (5.3%)	58.2 (4.3%)	91.8 (64%)	127 (127%)	57.6 (3.3%)
D2	1	11.35 (3.29%)	8.01 (-29%)	10.8 (-4.5%)	11.1 (-1.9%)	13.7 (20%)	16.6 (46%)	11.3* (0.1%)
	2	2.643 (1.25%)	2.96 (12%)	2.75 (4.0%)	2.82 (6.7%)	2.85 (7.8%)	3.06 (16%)	3.06 (16%)
	3	26.87 (2.04%)	32.9 (22%)	26.8 (-0.4%)	24.9 (-7.5%)	27.5 (2.2%)	36.4 (35%)	31.1* (16%)
	Total	98.36 (1.82%)	102 (3.4%)	97.2 (-1.2%)	94.4 (-4.0%)	104 (6.0%)	132 (34%)	105 (7.1%)
D3	1	11.39 (3.04%)	7.95 (-30%)	11.0 (-3.5%)	11.3 (-0.5%)	15.8 (39%)	16.5 (45%)	11.3* (-0.5%)
	2	2.290 (1.27%)	2.90 (27%)	2.53 (10%)	2.26 (-1.4%)	2.57 (12%)	3.04 (33%)	2.10* (-8.2%)
	3	2.220 (0.59%)	2.40 (7.9%)	2.38 (7.0%)	2.59 (16%)	2.39 (7.6%)	2.43 (9.6%)	2.43 (9.6%)
	Total	47.72 (2.51%)	40.2 (-16%)	47.8 (0.2%)	48.2 (1.0%)	62.6 (31%)	66.6 (39%)	47.5 (0.51%)
D4	1	11.30 (6.39%)	7.97 (-29%)	10.9 (-3.2%)	11.3 (0.3%)	14.2 (26%)	16.43 (45%)	11.3* (0.3%)
	2	2.414 (1.12%)	2.93 (21%)	2.64 (9.5%)	2.60 (7.7%)	2.65 (10%)	3.05 (26%)	2.10* (-13%)
	3	5.886 (1.05%)	6.83 (16%)	6.31 (7.3%)	6.17 (4.8%)	6.47 (10%)	6.85 (16%)	5.95* (1.1%)
	Total	55.24 (4.37%)	49.3 (-11%)	56.0 (1.4%)	56.7 (2.7%)	69.3 (25%)	75.5 (37%)	54.3 (-1.7%)
Average absolute relative error			20.24%	4.72%	4.52%	21.61%	42.60%	5.51%

intensity, far greater than the traffic intensities at the other queues. Moreover, these two stations are quite closely coupled. Thus, at first glance, we expect that SBD with two-dimensional RBM should perform very well, which proves to be correct. Moreover, this example should be challenging for RQNA because it is based on heavy-traffic limits for OQNs with only a single bottleneck, thus involving only one-dimensional RBM.

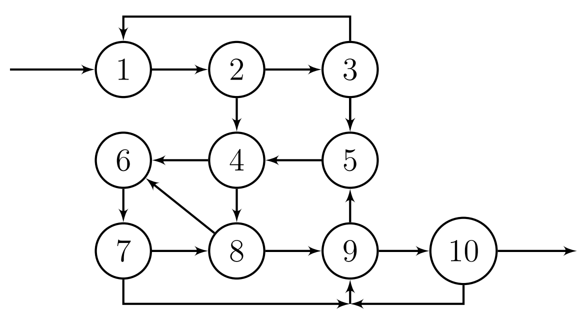

Figure 2: A ten-station with customer feedback example.

In Table 5, we report the simulation estimates and approximattions for the steady-state mean sojourn time (waiting time plus service time) at each station, as well as the total sojourn time of the system, calculated as in (14). For the approximations, we compare QNA from [50], QNET from [22], SBD from [11], RQ from [57] (with estimated IDC), as well as the RQNA algorithms here. The simulation, QNA, QNET and SBD columns are taken from

Table 5: A comparison of six approximation methods to simulation for the mean steady-state sojourn times at each station of the open queueing network in Figure 2.

Station	Simulation	QNA	QNET	SBD	RQ	RQNA	RQNA (elim)
1	0.99 (0.86%)	0.97 (-2.8%)	1.00 (0.2%)	1.00 (0.4%)	0.97 (-2.0%)	1.09 (9.2%)	1.00* (0.4%)
2	0.55 (0.69%)	0.58 (6.0%)	0.56 (2.6%)	0.55 (0.2%)	0.55 (-0.1%)	0.56 (1.3%)	0.56 (1.4%)
3	2.82 (1.93%)	2.93 (4.2%)	2.90 (3.2%)	2.76 (-2.0%)	2.96 (5.0%)	3.40 (21%)	2.75* (-2.5%)
4	1.79 (3.71%)	1.34 (-25%)	1.41 (-21%)	1.76 (-1.6%)	2.34 (31%)	3.51 (97%)	2.11* (18%)
5	2.92 (4.77%)	2.49 (-15%)	2.44 (-17%)	2.81 (-3.6%)	3.77 (29%)	9.07 (211%)	3.35* (15%)
6	0.58 (0.78%)	0.64 (10%)	0.62 (7.4%)	0.59 (2.2%)	0.60 (3.8%)	0.70 (20%)	0.49* (-16%)
7	0.24 (0.28%)	0.24 (-1.7%)	0.26 (7.1%)	0.27 (11%)	0.23 (-3.0%)	0.24 (-1.3%)	0.24 (-1.3%)
8	0.58 (0.67%)	0.64 (9.6%)	0.61 (4.6%)	0.60 (1.7%)	0.61 (3.9%)	0.70 (20%)	0.59* (0.6%)
9	0.34 (0.63%)	0.32 (-6.1%)	0.35 (2.0%)	0.43 (26%)	0.33 (-4.2%)	0.73 (111%)	0.42* (21%)
10	0.29 (0.19%)	0.30 (2.4%)	0.29 (1.4%)	0.28 (-1.7%)	0.28 (-1.5%)	0.26 (-8.7%)	0.26 (-8.7%)
Total	22.0 (2.45%)	20.3 (-7.9%)	20.4 (-7.3%)	22.4 (1.7%)	26.1 (18%)	44.5 (102%)	24.2* (9.9%)

Table XIV of [11].

Again, we consider two versions of RQNA algorithm, the first one does not eliminate feedback, while the second one (marked by 'elim') applies the feedback elimination procedure. As before, in eliminating customer feedback, for each station, we identify the near-immediate feedback flows as the flows that come back to the station after completing service, without passing through any station with a higher traffic intensity. We then eliminate all near-immediate feedback flows, apply plain RQNA algorithm on the reduced network and use the new RQNA approximation as the approximation for that station.

We make the following observations from this numerical example:

1. Particular attention should be given to the two bottleneck stations: 4 and 5. Note that QNA and QNET produce $15-25\%$ error, which is satisfactory, but SBD does far better with only $1-4\%$ error.

2. The RQNA algorithm without feedback elimination can perform very poorly with high traffic intensity and high feedback probability, presumably due to the break down of the IDW decomposition in (4).

3. With feedback elimination, the RQNA algorithm performs significantly better and is competitive with previous algorithms in this complex setting, producing $15-18\%$ error at stations 4 and 5. The performance of RQNA at the tightly coupled bottleneck queues evidently suffers because the current RQNA depends heavily on one-dimensional RBM.

7 Conclusions

7.1 Summary

In this paper we developed a new decomposition approximation for the principal steady-state performance measures of each queue in a single-class open queueing network of single-server queues with unlimited waiting space and the first-come-first served service discipline. We focus on non-Markov OQNs where the external arrival processes need not be Poisson or renewal and the service-time distributions need not be exponential. Our algorithm combines three methodologies in operations research and stochastic models: (i) robust optimization as in [3, 57], (ii) indices of dispersion and stationary point processes as in [9, 15, 45] and (iii) heavy-traffic limits as in [11, 22, 55]. The algorithm builds on our previous papers [56, 57, 58, 59, 60] as indicated in §1.2.

Given the model data, the computational effort is the same as for QNA in [50]. Efficient ways to obtain the model data, primarily the indices of dispersion of the external arrival processes, are indicated in §2.1. Just as for QNA in [50], an effective way to apply the algorithm in applications is together with simulation. The analytical algorithm can be used to rapidly explore and optimize over spaces of candidate models, while simulation can be used to confirm algorithm predictions.

In addition to the goal of computing steady-state performance measures of interest, a major goal in this work has been to gain a better understanding of the dependence in the flows of an OQN and the impact of that dependence upon the performance of the queues. Heavy-traffic limits have traditionally aimed at exposing the performance impact by skipping this step. We have used indices of dispersion to approximately characterize the dependence. The starting point is to link the indices of dispersion to the performance of a single queue. That initial step was provided with robust queueing in [57]. Theorem 5 of [57] shows that the robust queueing based on the IDC is asymptotically correct in both light and heavy traffic.

Nevertheless, it was not evident that the approximation of one queue in [57] could be extended to yield an analog of QNA in [50] for a general OQN. With the aid of heavy-traffic limits for the flows in [56, 60], the present paper synthesizes those theoretical results and develops an efficient algorithm for a general OQN.

After reviewing the indices of dispersion and the robust queueing approximation for a single queue in §2, we developed the important variability linear equations for the IDCs of the internal arrival processes in §3. We then introduced the extra step of feedback elimination in §4. We put all this together into a full algorithm in §5, developing a simplified version for networks with tree structure in §3.5.

We then evaluated the performance of the new RQNA-IDC by making comparisons with simulations for various examples in §6 and §7 of the appendix. These experiments confirm that RQNA-IDC is remarkably effective.

7.2 When Should the IDC-Based RQNA Be Effective?

It is significant that the IDC provides a useful diagnostic tool to judge when candidate performance approximations for OQNs are likely to be effective or not. This is well illustrated by the figures in [56, 57]. They show plots of the IDC in (1) as a function of time and the normalized mean workload in (5) as a function of the traffic intensity.

The easiest case is a Poisson process when the IDC is 1. If an entire IDC is nearly 1, then the arrival process should behave much like a Poisson process. More generally, when the IDC is nearly constant, there should be relatively little ambiguity about the relevant level of variability in the arrival process; e.g., see the light traffic and heavy-traffic limits in (7). For the $GI/GI/1$ model with a renewal arrival process, the IDC and IDW approach limits as time evolves, usually with exponential decay. Thus, standard approximations are usually effective.

In an OQN, this good behavior is likely to prevail if the level of variability of in all the service times, as measured by their scv's, and in all the external arrival processes, as characterized by the IDC's, are roughly equal. Experience has shown that the difficult examples typically arise when that property is seriously violated. This is reflected by the convex combination appearing in the approximation for the departure process in equation (18). More generally, problems with the approximations are likely to arise as the complexity of the OQN increases when the level of variability is not nearly constant, as indicated in §1.2.

The traffic intensities of the queues also play a role. The RBM-based heavy-traffic QNET algorithm in [22] is likely to be especially effective if the traffic intensities are nearly equal and relatively high. The SBD decomposition in [11] is likely to be especially effective if the traffic intnsities can be separated into groups, with some high, others medium and others low. The RQNA developed here is likely to be especially effective if there is a single bottleneck node, because we exploit the heavy-traffic theory in [60] for that case. That condition is violated for the three-station examples in §6.

It is important to note that our numerical examples have deliberately been chosen from the most difficult cases exposed in previous work. The first class of notorious examples is based on the heavy-traffic bottleneck phenomenon from [48], which is studied in [11, 56] and in §7.3 of the appendix to this paper. The different levels of variability appear at different queues depending on the traffic intensity of the queue. The second class are the networks with near-immediate feedback from [11], which is studied here in Tables 3 and 4 here.

The ten-station example in Table 5 here from [11] has quite a bit of feedback, but is not so difficult. Note that, all methods produce reasonable accuracy for this example, provided feedback elimination is incorporated in the IDC-based RQNA here. For many-realistic OQNs arising in practice, such as the large manufacturing examples in [44], most methods work quite well.

7.3 Directions for Future Research

There are many excellent directions for future research, including (i) developing refined approximations for the flows that exploit multi-dimensional RBM instead of just one-dimensional RBM, (ii) extending RQNA-IDC to other OQN models, e.g., with multiple servers and other service disciplines and (iii) extending our initial robust queueing for a time-varying queue in [59] to time-varying networks of queues. In fact, we think that our work should be regarded as only one step in the serious study of dependence in stochastic point (arrival) processes, queueing networks and related stochastic models.

Acknowledgements

This work was done while the Wei You was a graduate student at Columbia University, where both authors received support from NSF grant CMMI 1634133.

References

[1] Abate J, Whitt W (1992) The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems* 10:5–88.

[2] Asmussen S (2003) *Applied Probability and Queues* (New York: Springer), second edition.

[3] Bandi C, Bertsimas D, Youssef N (2015) Robust queueing theory. *Operations Research* 63(3):676–700.

[4] Banerjee S, Johari R, Riquelme C (2015) Pricing in ride-sharing platforms: A queueing-theoretic approach. *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 639 (ACM).

[5] Boucherie RJ, van Dijk NM, eds. (2011) *Queueing Networks*. International Series in Operations Research and Management Science, Number 154 (Springer).

[6] Brumelle S (1971) On the relation between customer averages and time averages in queues. *J. Appl. Prob.* 8(3):508–520.

[7] Chan CW, Dong J, Green LV (2016) Queues with time-varying arrivals and inspections with applications to hospital discharge policies. *Operations Research* 65(2):469–495.

[8] Chen H, Yao DD (2001) *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization* (New York: Springer).

[9] Cox DR, Lewis PAW (1966) *The Statistical Analysis of Series of Events* (London: Methuen).

[10] Creemers S, Lambrecht M (2011) Modeling a hospital queueing network. *Queueing Networks*, chapter 18, 767–798 (Springer).

[11] Dai J, Nguyen V, Reiman MI (1994) Sequential bottleneck decomposition: an approximation method for generalized Jackson networks. *Operations Research* 42(1):119–136.

[12] Dai J, Shi P (2019) Inpatient overflow: An approximate dynamic programming approach. *Manufacturing & Service Operations Management* 21(4):713–748.

[13] Dai JG, Harrison JM (1992) Reflected Brownian motion in an orthant: numerical methods for steady-state analysis. *The Annals of Applied Probability* 2(1):65–86.

[14] Daley D, Vere-Jones D (2008) *An Introduction to the Theory of Point Processes: General Theory and Structure*, volume II (Oxford, U. K.: Springer), second edition.

[15] Daley DJ, Vere-Jones D (2008) *An Introduction to the Theory of Point Processes* (Oxford, U. K.: Springer), second edition.

[16] Dieker AB, Ghosh S, Squillante MS (2016) Optimal resource capacity management for stochastic networks. *Operations Research* 65(1):221–241.

[17] Fendick KW, Saksena V, Whitt W (1989) Dependence in packet queues. *IEEE Trans Commun.* 37:1173–1183.

[18] Fendick KW, Whitt W (1989) Measurements and approximations to describe the offered traffic and predict the average workload in a single-server queue. *Proceedings of the IEEE* 71(1):171–194.

[19] Freund D, Henderson SG, Shmoys DB (2017) Minimizing multimodular functions and allocating capacity in bike-sharing systems. *International Conference on Integer Programming and Combinatorial Optimization*, 186–198 (Springer).

[20] Harrison JM (1973) The heavy traffic approximation for single server queues in series. *Journal of Applied Probability* 10(3):613–629.

[21] Harrison JM (1978) The diffusion approximation for tandem queues in heavy traffic. *Advances in Applied Probability* 10(4):886–905.

[22] Harrison JM, Nguyen V (1990) The QNET method for two-moment analysis of open queueing networks. *Queueing Systems* 6(1):1–32.

[23] Harrison JM, Reiman MI (1981) Reflected Brownian motion on an orthant. *The Annals of Probability* 302–308.

[24] Harrison JM, Williams RJ (1987) Multidimensional reflected Brownian motions having exponential stationary distributions. *The Annals of Probability* 115–137.

[25] Horváth A, Horváth G, Telek M (2010) A joint moments based analysis of networks of $MAP/MAP/1$ queues. *Performance Evaluation* 67(9):759–778.

[26] Iglehart DL, Whitt W (1970) Multiple channel queues in heavy traffic, I. *Advances in Applied Probability* 2(1):150–177.

[27] Iglehart DL, Whitt W (1970) Multiple channel queues in heavy traffic, II: Sequences, networks and batches. *Advances in Applied Probability* 2(2):355–369.

[28] Jagerman DL, Balcıoglu B, Altıok T, Melamed B (2004) Mean waiting time approximations in the G/G/1 queue. *Queueing Systems* 46(3-4):481–506.

[29] Kemeny JG, Snell JL (1976) *Finite Markov Chains* (New York: Springer).

[30] Kim S (2011) Modeling cross correlation in three-moment four-parameter decomposition approximation of queueing networks. *Operations Research* 59(2):480–497.

[31] Kim S (2011) The two-moment three-parameter decomposition approximation of queueing networks with exponential residual renewal processes. *Queueing Systems* 68:193–216.

[32] Kim SH, Whitt W, Cha WC (2018) A data-driven model of an appointment-generated arrival process at an outpatient clinic. *INFORMS Journal on Computing* 30(1):181–199.

[33] Kuehn PJ (1979) Approximate analysis of general queuing networks by decomposition. *IEEE Transactions on Communications* 27(1):113–126.

[34] Li A, Whitt W (2014) Approximate blocking probabilities for loss models with independence and distribution assumptions relaxed. *Performance Evaluation* 80:82–101.

[35] Li SQ, Hwang CL (1992) Queue response to input correlation functions: discrete spectral analysis. *IEEE INFOCOM'92: The Conference on Computer Communications*, 382–394 (IEEE).

[36] Li SQ, Hwang CL (1993) Queue response to input correlation functions: continuous spectral analysis. *IEEE/ACM transactions on networking* 1(6):678–692.

[37] Neuts MF (1979) A versatile Markovian point process. *Journal of Applied Probability* 16(4):764–779.

[38] Neuts MF (1989) *Structured Stochastic Matrices of $M/G/1$ Type and their Application* (New York: Marcel Dekker).

[39] Ozkan E, Ward A (2017) Dynamic matching for real-time ridesharing, working paper, University of Sourthern California.

[40] Reiman MI (1984) Open queueing networks in heavy traffic. *Math. Oper. Res.* 9(3):441–458.

[41] Reiman MI (1990) Asymptotically exact decomposition approximations for open queueing networks. *Operations Research Letters* 9(6):363–370.

[42] Ross SM (1996) *Stochastic Processes* (New York: Wiley), second edition.

[43] Sauer CH, Chandy KM (1981) Computer systems performance modeling. *SIAM Rev.* 124—-125.

[44] Segal M, Whitt W (1989) A Queueing Network Analyzer for Manufacturing. Bonatti M, ed., *Teletraffic Science for New Cost-Effective Systems, Networks and ServicesProceedings: ITC 12, Proceedings of the 12th International Teletraffic Congress*, 1146–1152 (Elsevier, North-Holland).

[45] Sigman K (1995) *Stationary Marked Point Processes: An Intuitive Approach* (New York: Chapman and Hall/CRC).

[46] Sinreich D, Marmor Y (2005) Emergency department operations: the basis for developing a simulation tool. *IIE transactions* 37(3):233–245.

[47] Sriram K, Whitt W (1986) Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications* SAC-4(6):833–846.

[48] Suresh S, Whitt W (1990) The heavy-traffic bottleneck phenomenon in open queueing networks. *Operations Research Letters* 9(6):355–362.

[49] Whitt W (1982) Approximating a point process by a renewal process: two basic methods. *Oper. Res.* 30:125–147.

[50] Whitt W (1983) The queueing network analyzer. *Bell Laboratories Technical Journal* 62(9):2779–2815.

[51] Whitt W (1984) Approximations for departure processes and queues in series. *Naval Research Logistics (NRL)* 31(4):499–521.

[52] Whitt W (1985) Queues with superposition arrival processes in heavy traffic. *Stochastic Processes and Their Applications* 21:81–91.

[53] Whitt W (1991) A review of $L = \lambda W$. *Queueing Systems* 9:235–268.

[54] Whitt W (1995) Variability functions for parametric-decomposition approximations of queueing networks. *Management Science* 41(10):1704–1715.

[55] Whitt W (2002) *Stochastic-Process Limits* (New York: Springer).

[56] Whitt W, You W (2018) Heavy-traffic limit of the GI/GI/1 stationary departure process and its variance function. *Stochastic Systems* 8(2):143–165.

[57] Whitt W, You W (2018) Using robust queueing to expose the impact of dependence in single-server queues. *Operations Research* 66(1):184–199.

[58] Whitt W, You W (2019) The advantage of indices of dispersion in queueing approximations. *Operations Research Letters* 47(2):99–104.

[59] Whitt W, You W (2019) Time-varying robust queueing. *Operations Research* 67(6):1766–1782.

[60] Whitt W, You W (2020) Heavy-traffic limits for stationary network flows. *Queueing Systems* 95:53–68.

[61] Zacharias C, Armony M (2016) Joint panel sizing and appointment scheduling in outpatient care. *Management Science* 63(11):3978–3997.

[62] Zeltyn S, Marmor YN, Mandelbaum A, Carmeli B, Greenshpan O, Mesika Y, Wasserkrug S, Vortman P, Shtub A, Lauterman T, et al. (2011) Simulation-based models of emergency departments: Operational, tactical, and strategic staffing. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 21(4):24.