Facilitating Load-Dependent Queueing Analysis Through Factorization

Giuliano Casale[a], Peter G. Harrison[a], Wai Hong Ong[a]

[a] *Department of Computing, Imperial College London, UK*

Abstract

We propose novel exact and approximate solutions for mean-value and probabilistic analysis of closed queueing networks with limited load-dependent nodes. The main result is the derivation of an *exact correction factor* between the equilibrium solution of a load-dependent model and the one of a related model with fixed-rate queueing stations. This enables the reuse of state-of-the-art methods for fixed-rate stations in the computation of performance metrics for load-dependent systems. As many such algorithms are available, our findings significantly increase the range of techniques available to study load-dependence. We further interpret the correction factor as a load-dependent normalizing constant and propose two novel integral forms, in the real and in the complex domains, which can be used for its efficient computation. These integral forms are also shown to be applicable to evaluate more general limited load-dependent systems, as we illustrate in a sojourn time distribution analysis problem. Lastly, the proposed algorithms are numerically examined through thousands of experiments, which reveal accuracy in the range 1%-6% mean absolute relative error.

Keywords: Normalizing constant, load-dependence, queueing network, closed system, multi-server

1. Introduction

The analysis of queueing networks with limited load-dependent (LLD) stations is a challenging problem, aimed at understanding the performance impact of service rates that depend on the number of resident jobs in a station, up to a finite population level. A common example of an LLD station is a multi-server queue, which has service rates that scale proportionally to the population at the station, until reaching a maximum value that depends on the number of servers.

Load-dependent modeling applies to capacity planning [1], symmetric multi-threading [2], cellular networks [3], wireless systems [4], logistics [5], service rate control [6], among others. Load-dependent service rates are also of theoretical interest for analyzing moments of performance metrics [7], to evaluate networks with state-dependent routing [8], non-product form systems [9], fork-join networks [10], and to apply the flow-equivalent server method [11].

Email addresses: `g.casale@imperial.ac.uk` (Giuliano Casale), `p.harrison@imperial.ac.uk` (Peter G. Harrison), `wai.ong11@imperial.ac.uk` (Wai Hong Ong)

Unfortunately, despite a long history of work on this subject [12], when LLD stations are considered in multiclass closed networks, which are used to abstract systems with finite levels of parallelism, they become expensive to analyze. Indeed, commonly employed evaluation techniques for models with fixed-rate stations, such as the convolution algorithm (CA) [13, 14, 15] and mean value analysis (MVA) [16] or its approximate versions [17], are much more expensive to apply when extended to the LLD setting. These methods can also incur numerical instabilities [18], as we further discuss in Section 8.

Motivated by these shortcomings, we develop novel computational algorithms to solve product-form LLD closed queueing networks. Our main result, developed in Section 3, is the determination of an *exact correction factor* between the solution of a LLD model, expressed in terms of the normalizing constant of the equilibrium state probabilities, and the solution of a similar model with fixed-rate stations. This finding paves the way to analyze LLD models by solving simpler fixed-rate models, which can be efficiently analyzed using established algorithms, for example approximate MVA methods [19]. We show that this correction factor may also be directly used to compute mean performance metrics, such as throughputs, from the corresponding metrics obtained in the fixed-rate models. Other mean performance metrics are studied in Appendix A.1, showing that expressions for mean queue-lengths and utilizations in LLD models may also be formulated in terms of normalizing constants and thus leverage correction factors for their computation.

The above developments further motivate the need for methods to efficiently compute the correction factor. The starting point of this investigation is that the correction factor is itself found to be a special multiclass normalizing constant for a related model defined on a very much reduced state space compared to the original LLD queueing network. Stemming from this observation the following results are obtained:

- We develop in Section 4 an approximate variant of the correction factor that has much lower computational requirements than the exact one developed in Section 3. In essence, the method involves determining the correction factor by summing a collection of single-class LLD normalizing constants. The resulting proposed approximation, called *reduction heuristic* (RD), is based on this result.

- In Section 5, we show that the multiclass LLD normalizing constant can be expressed as a finite difference of single-class LLD normalizing constants. We develop a multivariate integral form for the multiclass normalizing constant that leverages the Norlund-Rice integral representation of the finite difference operator [20]. The integrand is shown to be related to a queueing network model with complex-valued parameters, for which we develop several characterization results.

- Section 5 also proposes a novel integral form for the multiclass LLD normalizing constant expressed as an integral over the unit simplex. This extends to the LLD setting some evaluation schemes for such integrals recently proposed for fixed-rate models [21].

To further show the practical benefits of our results, we report an extensive experimentation, evaluating more than $100,000$ models with varying sizes and parameters. Our results indicate average errors as low as 1.21% for mean performance metrics in large models, as well as normalizing constant approximation errors around 6.46%. We also give examples that demonstrate how our results can be applied to non-product-form multiclass LLD stations. We show in this setting that mean and percentiles of response times in multiclass first-come first-served (FCFS) queues may be more accurately estimated with the proposed RD heuristic than with baseline methods.

Summarizing, our work provides novel exact insights in the analysis of limited load-dependence, leading to novel ways to efficiently analyze LLD models. The rest of this paper is organized as follows. Definitions and background are given in Section 2. Sections 3-5 develop the contributions we have listed above. A quantitative study to validate the accuracy of our approximations is reported in Section 6. Section 7 we illustrate the application of the proposed analysis methods to non-product-form models. A summary of the state-of-the-art in computational methods for LLD models is given in Section 8. The paper concludes in Section 9. The appendix gives proofs, further background, and additional validation experiments.

Lastly, we remark that a short abstract showing a preliminary version of this work has appeared in [22]. The present paper includes many more results, as well as proofs not included therein. In particular, this work develops the RD heuristic, the LLD correction factors, a large-scale numerical validation, and the complex-domain derivations related to Norlund-Rice integral form.

2. Reference model

2.1. Closed product-form queueing networks

We focus on closed product-form queueing networks consisting of M load-dependent stations and R job classes. Such models have been extensively investigated in prior work, we point the reader for an introduction to [14, 23]. A table summarizing the model notation is given in Table 1. Scheduling policies compatible with the product-form assumptions include processor sharing, first-come first-served with identical service rates per class, and infinite server, among others [24]. Indexes i, j, k are used to denote queues ($i, j, k = 1, \dots, M$), whilst indexes r, s are used for classes ($r, s = 1, \dots, R$). Each class r is populated by N_r jobs. After receiving service, jobs in class r depart from node i joining station j with probability p_{rij}; self-loops are allowed. Jobs are assumed to remain of the same class at all times, since models with class switching can be reduced to this case without approximation error [25, 23]. We also define the population vector $\boldsymbol{N} = (N_1, \dots, N_R)$ and the total number of jobs in the network $N = |\boldsymbol{N}| = N_1 + \dots + N_R$, where $|\cdot|$ denotes vector sum.

Table 1: Summary of main notation

Notation	Description
i, j, k	station indexes
r, s	class indexes
M	Number of queueing stations
R	Number of service classes
N_r	Population of jobs in class r
N	Total population of jobs in the model
$\boldsymbol{N}$	Population vector
β_r	Fraction of class r jobs in the mix ($\beta_r = N_r/N$)
n_k	Total number of jobs at node k in a given state
$\boldsymbol{\theta}$	Demand vector
θ_k	Mean single-class service demand at node k
$\theta_{k,r}$	Mean service demand of class-r jobs at node k
$\theta_{0,r}$	Think time of class r
$\boldsymbol{\sigma}$	Scaled demand vector
σ_k	Scaled single-class mean service demand at node k ($\sigma_k = \theta_k/\alpha_k(s_k)$)
$\sigma_{k,r}$	Scaled mean service demand of class-r jobs at node k ($\sigma_{k,r} = \theta_{k,r}/\alpha_k(s_k)$)
$V_{k,r}$	Mean number of visits of class-r jobs at node k
$\alpha_k(n_k)$	LLD scaling function for node k
s_k	LLD limit ($\alpha_k(n_k) = \alpha_k(s_k)$, $\forall n_k \geq s_k$)
$H_{\boldsymbol{\theta}}(\boldsymbol{N})$	multiclass LLD normalizing constant with demands $\theta_{k,r}$
$h_{\boldsymbol{\theta}}(N)$	single-class LLD normalizing constant with demands θ_k
$G_{\boldsymbol{\sigma}}(\boldsymbol{N})$	multiclass fixed-rate normalizing constant with demands $\sigma_{k,r}$
$g_{\boldsymbol{\sigma}}(N)$	single-class fixed-rate normalizing constant with demands σ_k
$C_r(\boldsymbol{N})$	system response time for class r in LLD models
$X_r(\boldsymbol{N})$	system throughput for class r in LLD models
$X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$	system throughput for class r in fixed-rate models with demands $\sigma_{k,r}$
$Q_{k,r}(\boldsymbol{N})$	mean queue-length of class-r at node k in LLD models
$Q_{k,r}^{\boldsymbol{\sigma}}(\boldsymbol{N})$	mean queue-length of class-r at node k in fixed-rate models with demands $\sigma_{k,r}$

The state space of the Markov process underlying the queueing network may be defined as $S(\boldsymbol{N}) = \{(n_{1,1}, \dots, n_{M,R}) \mid \sum_{k=1}^{M} n_{k,r} = N_r, n_{k,r} \geq 0\}$, where $n_{k,r}$ is the number of class-r jobs residing at station k, either queueing or receiving service. We denote by $\theta_{k,r} = V_{k,r}/\mu_{k,r}$ the mean service demand of class r at station k, defined as the ratio of the mean number of visits $V_{k,r}$ of class r jobs to queue k over the mean service rate $\mu_{k,r}$ of class r jobs at queue k. The $V_{k,r}$ value can be obtained from the routing probability matrix p_{rij} by means of the network traffic equations [26]. The sum of the demands of class r jobs at infinite server stations is denoted by $\theta_{0,r}$ and referred to as the class-r think time. Due to known equivalence results, we may assume without loss of generality that all infinite server stations are merged into a single one having class-r demand $\theta_{0,r}$ [27, Thm. 3.1.1].

2.2. LLD scaling factors

In a load-dependent queueing station k, the service rate changes instantaneously with the number of jobs presently at the station. In a state where $n_k = \sum_{r=1}^{R} n_{k,r}$ jobs are resident in the station, including the job in service, the instantaneous service rate is $\alpha_k(n_k)\mu_{k,r}$, $n_k \geq 0$, with $\alpha_k(0) = 1$. For example, a multi-server station with s_k homogeneous servers is known to be a load-dependent station with $\alpha_k(n_k) = \min(s_k, n_k)$. Single-server fixed-rate station are instead captured by the assignment $\alpha_k(n_k) = 1$. Several other load-dependent scaling factors $\alpha_k(n_k)$ have been used in the literature [28, 29].

Limited load-dependence (LLD) adds to the above the mild restriction that each scaling function $\alpha_k(n_k)$ must become constant after a large enough n_k. That is, we can associate with each LLD station k a constant s_k such that

$$\alpha_k(n_k) = \alpha_k(s_k), \qquad \forall n_k \geq s_k \tag{1}$$

For example, a multi-server node meets this condition since its service rate becomes constant whenever $n_k \geq s_k$. LLD is common in prior work [30, 31], as in practice only restricts the ability to study the asymptotic behavior of the system when s_k varies as $N \to +\infty$. For any finite value of N, and in all real-world applications, the class of models we consider is entirely general. The s_k constants are collected in a vector $\boldsymbol{s} = (s_1, \dots, s_M)$, and hence fixed-rate models are those with $\boldsymbol{s} = (1, \dots, 1)$. Throughout, we shall thus parameterize models both in terms of the original demands $\boldsymbol{\theta} = (\theta_{k,r})$ and of the *scaled demands* $\sigma_{k,r} = \theta_{k,r}/\alpha_k(s_k)$, grouped in matrix $\boldsymbol{\sigma} = (\sigma_{k,r})$.

2.3. Equilibrium distribution

From the previous definitions, it can be shown that the equilibrium state probability mass function of the network is given by [24]:

$$\pi(\boldsymbol{n}) = \frac{1}{H_{\boldsymbol{\theta}}(\boldsymbol{N})} \prod_{k=1}^{M} F_k(\boldsymbol{n}_k), \qquad F_k(\boldsymbol{n}_k) \equiv \frac{n_k!}{\prod_{t=1}^{n_k} \alpha_k(t)} \prod_{r=1}^{R} \frac{\theta_{k,r}^{n_{k,r}}}{n_{k,r}!} \tag{2}$$

where $\boldsymbol{n} \in S(\boldsymbol{N})$ and $\boldsymbol{n}_k = (n_{k,1}, \dots, n_{k,R})$. Here, the normalizing constant $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ ensures that the state probabilities sum to one and may be expressed as

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \sum_{\boldsymbol{n} \in S(\boldsymbol{N})} \prod_{k=1}^{M} F_k(\boldsymbol{n}_k) \tag{3}$$

Throughout, single-class normalizing constants are distinguished by a lower-case font, class indexes absent, and a scalar population N rather than a vector $\boldsymbol{N}$, e.g., $h_{\boldsymbol{\theta}}(N)$.

2.4. LLD mean performance metrics

We now introduce standard expressions for computing performance metrics, the interested reader can find further justification of these metrics in [13, 32, 14] In the considered class of models, the system throughput for class-r jobs, which represents the completion rates of class-r jobs per unit visit at an arbitrarily chosen reference station, may be computed using the standard expression

$$X_r(\boldsymbol{N}) = \frac{H_{\boldsymbol{\theta}}(\boldsymbol{N} - 1_r)}{H_{\boldsymbol{\theta}}(\boldsymbol{N})} \tag{4}$$

where 1_r is a vector of all zeros except for a one in position r and $H_{\boldsymbol{\theta}}(\boldsymbol{N} - 1_r)$ stands for the normalizing constant of a model with a class-r job less. The previous quantity can be applied to computing throughputs at each station in the model. At a station i that is visited by class-r jobs V_{ir} times on average before completion, the station i throughput is $X_{ir}(\boldsymbol{N}) = V_{ir} X_r(\boldsymbol{N})$.

The mean queue-length in the general LLD case is instead given by the R-dimensional convolution

$$Q_{k,r}(\boldsymbol{N}) = \sum_{\boldsymbol{0} \le \boldsymbol{n}_k \le \boldsymbol{N}} \frac{F_k(\boldsymbol{n}_k) H_{\boldsymbol{\theta}}^{-k}(\boldsymbol{N} - \boldsymbol{n}_k)}{H_{\boldsymbol{\theta}}(\boldsymbol{N})} \tag{5}$$

Note that the time and space complexity of the last expression are both exponential in the number of classes R.

A different approach to compute mean LLD queue-lengths is proposed in [33, Thm. 3], which defines an algorithm to construct modified service rate functions $\alpha_k(\boldsymbol{n})$ so that, by adding a special station to the model parameterized with such functions, it is possible to recover from this new model the mean-queue length for station k in the original system. To compute all mean queue-lengths, this process needs to be repeated M times, by solving one auxiliary model for each station k in the original queueing network.

The approach is generalized in the present paper and obtains multiclass mean queue-lengths in the LLD setting more efficiently than with (5), at the cost of solving MR independent LLD models. We show this novel generalization in Appendix A.1. The computation of these quantities can leverage the results developed in sections 3 and 4-5.

2.5. LLD normalizing constants

Closed-form expressions for fixed-rate normalizing constants are derived in [30]. Similarly, [34, Eq. (29)] obtains the following closed-form solution for single-class LLD models with multi-server stations

$$h_{\boldsymbol{\theta}}(N) = \sum_{\mathbf{0} \le \boldsymbol{v} < \boldsymbol{s}} \sum_{i=1}^{M} \frac{(\theta_i/s_i)^{N+M-v-1}}{\prod_{j \neq i}(\theta_i/s_i - \theta_j/s_j)} \prod_{k=1}^{M} \frac{\theta_k^{v_k}}{v_k!} \left(1 - \frac{v_k}{s_k}\right) \tag{6}$$

This expression is valid only under the assumption that each θ_i/s_i is distinct and thus does not include the general LLD case. Using scaling properties of the normalizing constant [35] and known asymptotic limits for networks with multi-server nodes [36], it can be shown that the leading term of the exact asymptotic expansion of (6) is given by

$$h_{\boldsymbol{\theta}}(N) \sim \frac{\rho_{max}^N N^{B-1}}{(B-1)!} \prod_{k=1}^{B} \frac{\theta_k^{s_k}}{s_k!} \prod_{j=B+1}^{M} \left(1 + \theta_j + \frac{\theta_j^2}{2!} + ... + \frac{\theta_j^{s_j-1}}{(s_j-1)!} + \frac{\theta_j^{s_j}}{s_j!(1-\rho_j)}\right) \tag{7}$$

where $\rho_{max} = \max_j \rho_j$, with $\rho_j = \theta_j/s_j$, and B is the number of stations with $\rho_j = \rho_{max}$ that are also assumed to have indexes $i = 1, ..., B$. The exact scaling factor ρ_{max}^N in the expression ensures that the service demands are normalized in $[0, 1]$ as in [36].

In this paper, we derive and leverage a more general explicit expression for the normalizing constant for a single-class model, which contrary to (6) applies to every LLD model. The result, stated here and derived in Appendix A.2, gives an algebraic equivalent of a result concerning generating functions obtained in [37, Eq. (8)].

Theorem 1. In a single-class LLD closed queueing network with M stations

$$h_{\boldsymbol{\theta}}(N) = \sum_{\mathbf{0} \le \boldsymbol{v} < \boldsymbol{s}} g_{\boldsymbol{\sigma}}(N - v)\Phi_{\boldsymbol{\theta}}(\boldsymbol{v}) \tag{8}$$

where $\boldsymbol{v} = (v_1, \dots, v_M)$, $v = |\boldsymbol{v}|$, $\boldsymbol{s} = (s_1, \dots, s_M)$, and we define $\Phi_{\boldsymbol{\theta}}(\boldsymbol{v}) = \prod_{k=1}^{M} \phi_k(v_k)$, and

$$\phi_k(v_k) = \frac{\theta_k^{v_k}}{\prod_{t=1}^{v_k} \alpha_k(t)} \left(1 - \frac{\alpha_k(v_k)}{\alpha_k(s_k)}\right)$$

for $0 \le v_k < s_k$.

In the above theorem, $\alpha_i(0)$ defined to be 0 and the convention that $\prod_{t=1}^{0} \cdot = 1$; the term $g_{\boldsymbol{\sigma}}(N-v)$ is the single-class normalizing constant of a fixed-rate model with demands $\sigma_k = \theta_k/\alpha_k(s_k)$ and a population of $N - v$ jobs ($g_{\boldsymbol{\sigma}}(n) = 0$ for $n < 0$). Note that in the special case of a network of multi-server stations with distinct scaled demands the last result specializes into the earlier result shown in (6). An example that validates this property is given in Appendix A.3.

For large values of N, the exact asymptotics of $h_{\boldsymbol{\theta}}(N)$ in the general LLD case is obtained in [37] assuming that the model has a strictly positive think time $\theta_0 > 0$. To our knowledge, no similar derivation is available for the general case. We here obtain this result, extending (7) to the general LLD case.

Theorem 2. The single-class LLD normalizing constant $h_{\boldsymbol{\theta}}(N)$ admits the exact asymptotics

$$h_{\boldsymbol{\theta}}(N) \sim \frac{\rho_{max}^{N} N^{B-1}}{(B-1)!} \prod_{j:\rho_j<\rho_{\max}} \frac{1}{1-\rho_j} \cdot \prod_{i=1}^{M} \sum_{n_i=0}^{s_i-1} \phi_i(n_i) \tag{9}$$

where $\rho_{max} = \max_j \rho_j$, with $\rho_j = \theta_j/\alpha_k(s_j)$, and B is the number of stations with $\rho_j = \rho_{max}$.

A proof of this result is given in Appendix A.4.

Few exact closed-form expressions for the LLD normalizing constant are available in the multiclass case. For networks with multi-server nodes, [28, (A14)] proposes an integral that utilizes Bessel functions of the first kind of order zero. The result, besides not addressing the LLD case in full generality, is difficult to evaluate numerically due to the rapid oscillations of the Bessel function, which reduce the efficiency of quadrature methods and Monte Carlo sampling commonly used for numerical integration. A formula in [28, Eq. 2.18] is also available for more general models, however, the paper addresses only networks with a single LLD station and the evaluation of the integrand is expensive as it requires to calculate high-order derivatives.

2.6. Motivating example

To further motivate the need for novel analysis methods for LLD models, we illustrate with a toy example some numerical difficulties that affect two state-of-the-art methods for load-dependent analysis: MVA-LD [16] and an AMVA algorithm for load-dependent models with default settings [29]. The example consists of a single-class queueing network with $R = 1$ class, a single multi-server station with demand $\theta_1 = 10$ and $s_1 = 100$ servers, think time $\theta_0 = 10$, and job population $N = 256$. Numerical solution is done with the underlying CTMC, discrete-event simulation based on JMT [38] (10^6 samples), MVA-LD and AMVA returns the results shown in Table 2.

The table indicates that MVA-LD and AMVA incur very large errors due to numerical issues. For MVA-LD, this is because high station utilization results in numerical instabilities in the computation of idle probabilities, with errors recursively propagating to the marginal queue-length probabilities. The problem faced by AMVA is instead a numerical issue associated to the internal representation of the scaling factors. This method needs to interpolate scaling factors with continuous functions, which for the present example is the soft-min, which depends on a κ parameter, see Appendix A.12 for more details. This parameter value is by default $\kappa = 20$, however we find that only when this is decreased below about $\kappa = 6$, AMVA converges to a correct result, as shown in the corrected AMVA column in Table 2. Unfortunately, the correction to the κ value is *model-dependent*, leaving unable to determine upfront whether AMVA will be reliable or not for a given model. This numerical issue of AMVA, to our knowledge, has not observed before in the literature. Moreover, the lower the κ value, the looser the approximation of the multi-server scaling is, besides violating in some cases the monotonicity assumption required in QD-AMVA for convergence [29].

Table 2: Toy example illustrating inaccurate solutions obtained with existing methods

Metric	Symbol	CTMC	SIM	MVALD	AMVA	AMVA (corrected)
Mean queue-length	Q_1	140.06	139.47	219.09	256.99	140.37
Utilization	U_1	0.9995	0.9994	0.3250	0.0000	0.9968
System throughput	X_1	11.694	11.725	3.7906	0.0008	11.663

We have also run similar experiments with larger multiclass models, observing similar numerical instabilities. Overall, this suggests that both MVA and AMVA methods incur numerical problems that require further research to address and motivate the investigations in the present paper.

3. A Novel Exact LLD Mean-Value Analysis

In this section, we propose our new exact method to efficiently compute LLD mean performance metrics, which relates the solution of a LLD model to the one of a fixed-rate queueing network. In particular, we first derive a novel general expression for the normalizing constant of a LLD model. We then show that this result allows us to express the normalizing constant as a product between a correction factor and a related normalizing constant of a fixed-rate model. These results can be directly applied to the computation of other performance metrics using the formulas in Appendix A.1.

Let us begin by considering a fixed-rate model with scaled demands $\boldsymbol{\sigma} = (\sigma_{kr})$, with $\sigma_{k,r} = \theta_{k,r}/\alpha_k(s_k)$. We then can write

$$G_{\boldsymbol{\sigma}}(\boldsymbol{N} - 1_r) = G_{\boldsymbol{\sigma}}(\boldsymbol{N}) X_r^{\boldsymbol{\sigma}}(\boldsymbol{N}) \tag{10}$$

by equation (4), where $X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$ is the class-r throughput in the fixed-rate model. Given an arbitrary integer vector $\boldsymbol{d}$ such that $\boldsymbol{0} \le \boldsymbol{d} \le \boldsymbol{N}$, we may now recursively expanding the expression on the left-hand side, obtaining

$$G_{\boldsymbol{\sigma}}(\boldsymbol{N} - \boldsymbol{d}) = G_{\boldsymbol{\sigma}}(\boldsymbol{N}) \prod_{(\boldsymbol{s},r)\in P(\boldsymbol{d},\boldsymbol{N})} X_r^{\boldsymbol{\sigma}}(\boldsymbol{s}) \tag{11}$$

where $P(\boldsymbol{d},\boldsymbol{N})$ is the collection of choices of r and resulting population vectors considered during the re-iterated application of (10). For example, a valid choice of $P(\boldsymbol{d},\boldsymbol{N})$ may be

$$P(\boldsymbol{d},\boldsymbol{N}) = \big\{(\boldsymbol{s},r) \,|\, \boldsymbol{s} = (N_1,\dots,N_{r-1},n_r,N_{r+1}-d_{r+1},\dots,N_R-d_R) : \forall r = 1,\dots,R; \forall n_r = N_r-d_r,\dots,N_r\big\} \tag{12}$$

The last expression can now be related to the LLD model under study through the next result, which is also a consequence of (8).

Theorem 3. The normalizing constant for a model with M LLD queueing stations and R classes, having demands $\theta_{k,r}$ and scaling functions α_k, admits the following *reduced convolution expression*

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \sum_{v=0}^{V} \sum_{\substack{\boldsymbol{d}\geq 0: \\ |\boldsymbol{d}|=v}} G_{\boldsymbol{\sigma}}(\boldsymbol{N}-\boldsymbol{d}) E_{\boldsymbol{\theta}}(\boldsymbol{d}) \tag{13}$$

where $\boldsymbol{d} = (d_1, \dots, d_R)$ and $V = \min(N, \sum_{k=1}^{M}(s_k - 1))$. Here, $G_{\boldsymbol{\sigma}}$ is the multiclass normalizing constant of a fixed-rate model with scaled demands $\sigma_{k,r} = \theta_{k,r}/\alpha_k(s_k)$, and we define

$$E_{\boldsymbol{\theta}}(\boldsymbol{d}) = \sum_{\boldsymbol{v}\in \boldsymbol{S}(\boldsymbol{d})} \prod_{i=1}^{M} \frac{v_i!}{\prod_{k=1}^{v_i} \alpha_i(k)} \left(1 - \frac{\alpha_i(v_i)}{\alpha_i(s_i)}\right) \prod_{r=1}^{R} \frac{\theta_{i,r}^{v_{i,r}}}{v_{i,r}!} \tag{14}$$

A proof is given in Appendix A.5. Note that the last expression refers to the state space $S(\boldsymbol{d})$ and therefore it is defined on a smaller state space than the original state space $S(\boldsymbol{N})$ whenever $V < N$. It should also be noted that, after algebraic simplifications, $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ may be itself seen as a LLD multiclass normalizing constant for a model with demands $\theta_{k,r}$ and scaling functions

$$\eta_k(n_k) = \begin{cases} 1 & \text{if } n_k = 0 \\ \alpha_k(n_k) \dfrac{(\alpha_k(s_k) - \alpha_i(n_k - 1))}{(\alpha_k(s_k) - \alpha_k(n_k))} & \text{if } 1 \leq n_k < s_k \\ +\infty & \text{if } n_k \geq s_k \end{cases} \tag{15}$$

for $k = 1, \dots, M$. Here, we use the degenerate assignment $\eta_k(n_k) = +\infty$ to restrict the summation that defines the normalizing constant over states where $n_k < s_k$. However, $\eta_k(n_k)$ is invalid if $\alpha_k(n_k) = \alpha_k(s_k)$, for some positive $n_k < s_k$. The definition of $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ avoids this issue by expressing (14) directly in terms of $\alpha_i(n_k)$ rather than using the $\eta_k(n_k)$.

By combining (11) and (13), the normalizing constant of a LLD model may now be seen as a multiple of the normalizing constant for the corresponding fixed-rate model with scaled demands $\boldsymbol{\sigma}$. This unexpected property is derived in the next theorem.

Theorem 4 (Exact LLD correction)**.** The normalizing constant for a model with M LLD queueing stations and R classes, having demands $\theta_{k,r}$ and scaling functions α_k, may be obtained from the normalizing constant of a related fixed-rate model with scaled demands $\boldsymbol{\sigma}$ as follows

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \Gamma(\boldsymbol{N}) G_{\boldsymbol{\sigma}}(\boldsymbol{N}) \tag{16}$$

where the *LLD correction factor* is the quantity

$$\Gamma(\boldsymbol{N}) = \sum_{v=0}^{V} \sum_{\substack{\boldsymbol{d}\geq 0: \\ |\boldsymbol{d}|=v}} \prod_{(\boldsymbol{s},r)\in P(\boldsymbol{d},\boldsymbol{N})} X_r^{\boldsymbol{\sigma}}(\boldsymbol{s}) E_{\boldsymbol{\theta}}(\boldsymbol{d}) \tag{17}$$

where $P(\boldsymbol{d}, \boldsymbol{N})$ is defined in (12), V is defined in Theorem 3, $X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$ is the mean system throughput of class r in the fixed-rate model, and $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ is given in (14).

As discussed in the introduction, the above determines the *exact* correction factor for a LLD model to be obtained from the solution of a fixed-rate one. In this result, it is unexpected to observe a full factorization between the load-dependent and fixed-rate terms, given that the two cannot be intuitively decomposed looking at the original normalizing constant definition (13). The result further extends to exact expressions for the class-r system throughput as follows.

Theorem 5. The exact relationship between the mean system throughput $X_r(\boldsymbol{N})$ of a LLD model and the corresponding metric $X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$ in a fixed-rate model with scaled demands $\boldsymbol{\sigma}$, is given by

$$X_r(\boldsymbol{N}) = \frac{\Gamma(\boldsymbol{N}-1_r)}{\Gamma(\boldsymbol{N})} X_r^{\boldsymbol{\sigma}}(\boldsymbol{N}) \tag{18}$$

for all classes $r = 1, \dots, R$, and in which $\Gamma(\cdot)$ is the LLD correction factor defined in (17).

Note that the complexity of (18) is asymptotically the same as the reduced convolution expression in (13) and therefore enjoys similar properties, namely that it can be determined using efficient methods for fixed-rate models, which can solve for performance measures in log-quadratic time, up to higher-order logarithmic terms, as shown in [39]. This is a major computational gain compared to the $\mathcal{O}(N^{R+1})$ time and space of existing exact methods such as MVA-LD.

4. Approximate LLD mean-value analysis

Stemming from the exact results obtained in the previous section, we now develop an heuristic approximation for LLD mean-value analysis that is scalable on large models. We being with noting that the constant in (14) admits the following property

$$\sum_{\substack{\boldsymbol{d}\geq 0:\\ |\boldsymbol{d}|=v}} c_1^{d_1} \cdots c_R^{d_R} \cdot E_{\boldsymbol{\theta}}(\boldsymbol{d}) = E_{\boldsymbol{\theta c}}(v) \tag{19}$$

where $\boldsymbol{c} = (c_1, \dots, c_R)$ are real coefficients, $\boldsymbol{\theta c} = (\sum_r \theta_{k,r} c_r)$ is a vector of M demands, in which $\boldsymbol{\theta c}$ is a matrix-vector product, and $e_{\boldsymbol{\theta}}(d)$ is a single-class LLD normalizing constant, defined as

$$e_{\boldsymbol{\theta}}(d) = \sum_{\substack{\boldsymbol{v}\geq \boldsymbol{0}:\\ |\boldsymbol{v}|=d}} \prod_{i=1}^{M} \frac{1}{\prod_{k=1}^{v_i} \alpha_i(k)} \left(1 - \frac{\alpha_i(v_i)}{\alpha_i(s_i)}\right) \theta_i^{v_i} \tag{20}$$

Formula (19) stems from general scaling properties of the normalizing constant [35] and an application of the multinomial theorem.

The last property can now be leveraged to simplify (16) and approximate the LLD correction factor using a *single-class* LLD model, instead than a multiclass one. To do so, we consider the throughputs in front of the $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ terms in (17) and simplify their product by the interpolation[1]

$$X_r^{\boldsymbol{\sigma}}(\boldsymbol{N}-\boldsymbol{1}_r) \approx X_r^{\boldsymbol{\sigma}}(\boldsymbol{N}) \frac{N-1}{N} \tag{21}$$

[1] Interpolations of the throughput function, as opposed to interpolation of mean queue-lengths, are rare but have been used before in AMVA methods, e.g., [40, p. 649].

which yields the expression

$$\prod_{(\boldsymbol{s},r)\in P(\boldsymbol{d},\boldsymbol{N})} X_r^{\boldsymbol{\sigma}}(\boldsymbol{s}) \approx (X_1^{\boldsymbol{\sigma}}(\boldsymbol{N}))^{d_1}\cdots(X_R^{\boldsymbol{\sigma}}(\boldsymbol{N}))^{d_R}\left(\frac{N-(d-1)^+}{N}\right)$$

where $(d-1)^+ = \max(0, d-1)$. Combining the last result with (19) allows us to aggregate the multiclass normalizing constants $E_{\boldsymbol{\theta}(\boldsymbol{N})}(\boldsymbol{d})$ into a tractable *single-class* LLD normalizing constant

$$\sum_{\substack{\boldsymbol{d}\geq 0:\\ |\boldsymbol{d}|=v}} \prod_{(\boldsymbol{s},r)\in P(\boldsymbol{d},\boldsymbol{N})} X_r^{\boldsymbol{\sigma}}(\boldsymbol{s}) E_{\boldsymbol{\theta}(\boldsymbol{N})}(\boldsymbol{d}) \approx \left(\frac{N-(v-1)^+}{N}\right) e_{\boldsymbol{\rho}(\boldsymbol{N})}(v)$$

where $\boldsymbol{\rho}(\boldsymbol{N}) = \boldsymbol{\theta}\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N})$ may be seen as a vector of utilizations and $\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N}) = (X_1^{\boldsymbol{\sigma}}(\boldsymbol{N}), \ldots, X_R^{\boldsymbol{\sigma}}(\boldsymbol{N}))$ is the throughput vector in the fixed-rate model with scaled demands $\boldsymbol{\sigma}$.

Summarizing, with the above derivations we have obtained the following result.

Approximation 1 (LLD correction heuristic)**.** *The LLD correction* (16) *may be approximated as*

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) \approx \gamma(\boldsymbol{N}) G_{\boldsymbol{\sigma}}(\boldsymbol{N}) \tag{22}$$

where the correction factor is estimated by

$$\gamma(\boldsymbol{N}) = \sum_{v=0}^{V} \left(\frac{N-(v-1)^+}{N}\right) e_{\boldsymbol{\rho}(\boldsymbol{N})}(v) \tag{23}$$

where $(v-1)^+ = \max(0, v-1)$*,* $\boldsymbol{\rho}(\boldsymbol{N}) = \boldsymbol{\theta}\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N})$ *and* $e_{\boldsymbol{\rho}(\boldsymbol{N})}(v)$ *defined as in* (20)*.*

Expression (23) can be efficiently evaluated to obtain the normalizing constant $H_{\boldsymbol{\theta}}(\boldsymbol{N})$. We first evaluate $e_{\boldsymbol{\rho}(\boldsymbol{N})}(v)$ by approximating the solution of a multiclass fixed-rate model to estimate the $\boldsymbol{\rho}(\boldsymbol{N})$ vector, which may be done by using standard AMVA methods for fixed-rate models to determine the throughputs $\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N})$, e.g., AQL [19]. After computing $\boldsymbol{\rho}(\boldsymbol{N})$, we are ready to determine the V single-class LLD normalizing constants $e_{\boldsymbol{\rho}(\boldsymbol{N})}(v)$ using a normalizing constant evaluation method for single-class LD or LLD models, such as (8) or (9).

It is also useful to note that the only approximation involved in the LLD correction heuristic is the interpolation (21). In the special case $V = 1$, which arises for example when the network includes a single LLD station with $s_i = 2$, this interpolation is however not required since the correction factors depends only on the throughputs $X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$. In this special case, (22) is exact provided that the terms $X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$ are evaluated with an exact solver for fixed-rate models.

A similar approach as the one we have just discussed can be further applied to mean-value analysis without explicitly computing the normalizing constant $H_{\boldsymbol{\theta}}(\boldsymbol{N})$. The next result follows immediately from (18) and enables this development.

Approximation 2 (RD Heuristic)**.** *The class-r system throughput of a LLD closed queueing network may be approximated as*

$$X_r(\boldsymbol{N}) \approx \frac{\gamma(\boldsymbol{N}-1_r)}{\gamma(\boldsymbol{N})} X_r^{\boldsymbol{\sigma}}(\boldsymbol{N}) \tag{24}$$

for all classes $r = 1, \dots, R$, where $\gamma(\cdot)$ is defined as in (23) and $X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$ is the class-r system throughput in a model consisting only of fixed-rate stations with scaled demands $\sigma_{k,r} = \theta_{k,r}/s_k$.

We call the above approximation to the exact formula (18), the reduction heuristic, or for short, RD heuristic. This is because the result is a consequence of the reduced convolution expression given in (14).

Besides computational savings, two key benefits of (24), are that: (i) *single-class* normalizing constants used within the $\gamma(\boldsymbol{N})$ and $\gamma(\boldsymbol{N} - 1_r)$ factors are usually numerically stable to compute using the standard LD convolution algorithm after applying an appropriate scaling to their demands [35]; (ii) there exist many AMVA methods (e.g., AQL [19]) that can efficiently estimate the throughputs $\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N})$ and $\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N} - 1_r)$ used to determine the $\boldsymbol{\rho}(\boldsymbol{N})$ and $\boldsymbol{\rho}(\boldsymbol{N} - 1_r)$ vectors within the correction factors.

In terms of complexity analysis, the terms $e_{\boldsymbol{\rho}(\boldsymbol{N})}(v)$, $v = 0, \dots, V$, can be computed exactly in $\mathcal{O}(MV^2)$ time and space using convolution. The throughputs $\boldsymbol{X}^{\boldsymbol{\sigma}}(\boldsymbol{N})$ can instead be computed efficiently using AMVA methods, such as AQL [19] or the Linearizer [41], in constant time and space with respect to the total number of jobs N in the network. Thus, as N grows large, for fixed V, the proposed approximation in equation (24) can also be computed in constant time and space with respect to N, and without requiring marginal probabilities.

It should be noted that, for very large V, for example in the order of thousands, it is still possible that (23) exceeds the floating-point range. The latter may occur even when the individual summands can be computed in a stable fashion. Several strategies can be used to deal with these two issues. To approximate large single class normalizing constants, one may rely on a number of existing results for single-class models, such as (7) for models with multi-server nodes and [37] for more general LLD models, as well as the Gaussian approximations we develop in the next sections. All these methods allow to work directly in the log-domain effectively providing numerical stability for realistic parameterizations used in applications.

To address floating-point range exceptions due to large summation it is instead possible to either neglect all terms except the largest one, or apply a log-sum-exp approximation [42]. For the latter, we first rewrite the ratio appearing in (24) as follows

$$\frac{\gamma(\boldsymbol{N} - 1_r)}{\gamma(\boldsymbol{N})} \approx \exp(\log \gamma(\boldsymbol{N} - 1_r) - \log \gamma(\boldsymbol{N})) \tag{25}$$

Then, we let $\log \gamma(\boldsymbol{N}) = \log \sum_v \exp(\log l_v)$, where $l_v = \log((N - (v-1)^+)/N) + \log e_{\boldsymbol{\rho}(\boldsymbol{N})}(v)$ and apply the log-sum-exp approximation

$$\log \gamma(\boldsymbol{N}) \approx l_{v_{max}} + \log \sum_{\substack{v=0 \\ v \neq v_{max}}}^{V} \exp(l_v - l_{v_{max}}) \tag{26}$$

where $v_{max} = \arg\max_v l_v$. A similar expression is also used to compute $\log \gamma(\boldsymbol{N} - 1_r)$ in (25).

5. Integral forms

In this section, we develop exact integral forms for LLD normalizing constants that can be used to numerically obtain the exact correction factors we have found in the last section. While the next developments are motivated by our need to obtain the LLD correction factors, our results apply in general to arbitrary LLD normalizing constants. For this reason, in the next sections we always refer to the normalizing constant with the general notation $H_{\boldsymbol{\theta}}(\boldsymbol{N})$, rather than with the $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ notation introduced for the LLD correction factors.

5.1. *Exact integral forms*

5.1.1. **Norlund-Rice integral form**

In order to introduce a new method to compute LLD normalizing constants, we first apply a representation for LLD normalizing constants based on forward differences. Let the notation $\Delta_{\boldsymbol{n}}^{\boldsymbol{k}}\cdot = \Delta_{n_p}^{k_p}\Delta_{n_{p-1}}^{k_{p-1}}\cdots\Delta_{n_1}^{k_1}\cdot$ indicate the composition of the differences of orders $k_1,\dots,k_p$ in the discrete variables $n_1,\dots,n_p$. Additional notation and properties for the finite difference operator $\Delta_{\boldsymbol{n},\boldsymbol{0}}^{\boldsymbol{N}}$ are given in Appendix A.6. The following result now holds.

Lemma 1. In a multiclass closed queueing network with M LLD queueing stations and R classes, the normalizing constant may be written as

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{\Delta_{\boldsymbol{n},\boldsymbol{0}}^{\boldsymbol{N}} h_{\boldsymbol{\theta n}}(N)}{N_1!\cdots N_R!} = \sum_{\boldsymbol{0}\le\boldsymbol{n}\le\boldsymbol{N}} \frac{(-1)^{N-n}}{N_1!\cdots N_R!}\prod_{r=1}^{R}\binom{N_r}{n_r} h_{\boldsymbol{\theta n}}(N) \tag{27}$$

where $\boldsymbol{n} = (n_1,\dots,n_R)$, $n = \sum_{r=1}^{R} n_r$, and in which

$$h_{\boldsymbol{\theta z}}(N) = \sum_{\substack{\boldsymbol{n}\ge\boldsymbol{0}:\\ |n|=N}} \prod_{k=1}^{M} \frac{\left(\sum_{r=1}^{R} z_r\theta_{k,r}\right)^{n_k}}{\prod_{t=1}^{n_k}\alpha_k(t)} \tag{28}$$

The proof is given in Appendix A.7. Observe now that a property of forward differences of a discrete function $f(n)$ is that they may be defined equivalently as a Norlund-Rice integral [20]

$$\Delta_n^k f(n) = \frac{k!}{2\pi i}\oint \frac{f(z+n)}{z(z-1)(z-2)\cdots(z-k)}dz$$

where i denotes the imaginary unit and the contour of integration needs to encircle the poles at $0,1,\dots,k$. This follows using the definition $\Delta_n = F_n - I$ and expanding Δ_n^k by the binomial theorem.

Let us now consider a vector of real variables $\boldsymbol{t} = (t_1,\dots,t_R)$ and define $\boldsymbol{\Theta}(\boldsymbol{t}) = \boldsymbol{\theta}\cdot(e^{it_1},\dots,e^{it_R})^T$, as a column vector composed of M complex-valued demands

$$\Theta_k(\boldsymbol{t}) = \sum_{r=1}^{R}\theta_{k,r}e^{it_r} \tag{29}$$

one for each station k. We then have the following result.

Theorem 6 (Norlund-Rice Integral Form.)**.** The multiclass LLD normalizing constant can be computed by the following R-dimensional integral

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi)^R} \int_0^{2\pi} \cdots \int_0^{2\pi} h_{\boldsymbol{\Theta}(\boldsymbol{t}-\boldsymbol{\beta}^T \boldsymbol{t})}(N) d\boldsymbol{t} \tag{30}$$

where $\boldsymbol{\beta} = N^{-1}\boldsymbol{N}$ is the class mix vector and the integrand is a single-class LLD normalizing constant with the elements of the demand vectors $\boldsymbol{\Theta}(\boldsymbol{t} - \boldsymbol{\beta}^T \boldsymbol{t})$ with entries specified according to (29).

The proof is given in Appendix A.8. We refer to the integral form above as the Norlund-Rice integral for the normalizing constant $H_{\boldsymbol{\theta}}(\boldsymbol{N})$. Noting that (30) evaluates to a real number $H_{\boldsymbol{\theta}}(\boldsymbol{N})$, it is possible to equivalently rewrite the last integral form as

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi)^R} \int_0^{2\pi} \cdots \int_0^{2\pi} \Re h_{\boldsymbol{\Theta}(\boldsymbol{t}-\boldsymbol{\beta}^T \boldsymbol{t})}(N) d\boldsymbol{t} \tag{31}$$

where $\Re$ denotes the real part of its argument. The last expression is convenient for implementation purposes, as it yields an integral of a real function over a real domain.

5.1.2. **Simplex integral form**

We now derive a second method for evaluating LLD normalizing constants based on integration over the unit simplex. It is known that the single-class fixed-rate normalizing constant $g_{\boldsymbol{\sigma}}(N)$ in (8) may be written as [21]

$$g_{\boldsymbol{\sigma}}(N) = \frac{(N+M-1)!}{N!} \int_{T_M} (\sigma_1 t_1 + ... + \sigma_M u_M)^N d\boldsymbol{t} \tag{32}$$

where the integration domain $T_M = \{\boldsymbol{t} \geq 0 : |\boldsymbol{t}| = 1\}$ is the unit simplex in M dimensions. We here extend the result to the LLD case, obtaining a novel integral form for $H_{\boldsymbol{\theta}}(\boldsymbol{N})$.

Theorem 7. The multiclass LLD normalizing constant can be computed by the following M-dimensional integral

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{\prod_{r=1}^R N_r!} \int_{T_M} \sum_{\boldsymbol{0} \leq \boldsymbol{v} < \boldsymbol{s}} \alpha_{\boldsymbol{v}} \boldsymbol{\Delta}_{l_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{l}}^{\boldsymbol{v}} \prod_{r=1}^R \left(\sum_{k=1}^M \sigma_{k,r}(l_k + l_0 t_k) \right)^{N_r} d\boldsymbol{t} \tag{33}$$

where $v = |\boldsymbol{v}|$, $\boldsymbol{l} = (l_1, \dots, l_M)$, and

$$\alpha_{\boldsymbol{v}} = \frac{(N+M-1-v)!}{(N-v)!} \prod_{k=1}^M \frac{\alpha_k(s_k)^{v_k}}{\prod_{t=1}^{v_k} \alpha_k(t)} \left(1 - \frac{\alpha_k(v_k)}{\alpha_k(s_k)}\right)$$

The proof is given in Appendix A.9. The above expression is exact and, differently from the Norlund-Rice form, requires integration on $\mathbb{R}^M$, M being the number of stations. As such, the simplex form should be preferred in models with many classes, where it is more computationally efficient than the Norlund-Rice form. The latter is instead more efficient on models with many stations.

5.2. *Numerical evaluation of integral forms*

Numerical methods may be used to estimate the integral forms in (31) and (33). Here we focus on two approaches: second-order approximations based on Gaussian methods and Grundmann-Möller (GM) cubature rules [43]. We summarize the essential ideas in the main text, pointing to appendices A.10 and A.11 for more details.

5.2.1. *Gaussian approximations*

Gaussian approximations to the integral form of normalizing constants have been used before for models with fixed-rate stations [44, 21], however they are to the best of knowledge novel in the context of LLD queueing models. The closest methods in the load-dependent case are probably the saddle point techniques proposed in [37], which however require restrictive assumptions on the network structure.

In this evaluation approach, we first apply a change of variables $\boldsymbol{t}(\boldsymbol{x})$ to (31) and (33) to map the integrand over the whole Euclidean space. In doing so, we also introduce a Jacobian $J(\boldsymbol{x})$ that decays at exponential rate with the distance from the mode of the transformed integrand. Subsequently, the resulting integral is approximated by a multivariate Gaussian. Afterwards, we apply approaches such as Monte Carlo sampling or Laplace-type approximations to numerically evaluate the integral. These approximations require the first-order and second-order derivatives of the integrand, which are analytically obtained in appendices A.10 and A.11 for the proposed integral forms. A high-level summary of the key insights is given below.

Norlund-Rice form. Obtaining the gradient and Hessian corresponding to (31) is quite complex, as one needs first to undertake an analysis of normalizing constants when the service demands are defined in the complex domain. This is required since the Norlund-Rice integral form depends on the demand matrix $\boldsymbol{\Theta}(\boldsymbol{t})$ defined in (29). The detailed expressions of these derivatives are presented in Appendix A.10. In particular, our derivations of the gradient and Hessian for the integrand of (31) depends on the change of variables $\boldsymbol{t}(\boldsymbol{x}) : (-\infty, \infty)^R \to (-\pi, \pi)^R$ chosen to map the integral form to $\mathbb{R}^R$. We restrict our attention to changes of variables where the transformation along the r-th dimension depends only on x_r, i.e., $\boldsymbol{t}(\boldsymbol{x}) = (t_1(x_1), \dots, t_R(x_R))$. Let us also define the shorthand notation

$$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_R(\boldsymbol{x})) = \boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{\beta}^T \boldsymbol{t}(\boldsymbol{x}) \tag{34}$$

where $\boldsymbol{x} = (x_1, \dots, x_R)$ and $\boldsymbol{\beta} = N^{-1}\boldsymbol{N}$ is the class-mix vector. Let the Jacobian of the change of variables be $J(\boldsymbol{x}) = \det\left(\frac{\partial \boldsymbol{t}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)$, then we wish to estimate

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi)^R} \int_{\mathbb{R}^R} h_{\boldsymbol{x}}(N) J(\boldsymbol{x}) d\boldsymbol{x} \tag{35}$$

where we use the shorthand notation $h_{\boldsymbol{x}}(N) = h_{\boldsymbol{\Theta}(\boldsymbol{f}(\boldsymbol{x}))}(N) = h_{\boldsymbol{\Theta}(\boldsymbol{t}(\boldsymbol{x})-\boldsymbol{\beta}^T\boldsymbol{t}(\boldsymbol{x}))}(N)$. Two concrete examples of a suitable change of variables that may be used to map the Norlund-Rice integration domain to $\mathbb{R}^R$ are the logit and the probit transformations.

- *Logit transformation.* We consider a scaled logit transformation $x_r = \log(\frac{t_r}{2\pi - t_r})$ which implies $t_r = t_r(x_r) = 2\pi e^{x_r}/(1+e^{x_r})$. This then gives a Jacobian $J(\boldsymbol{x}) = (2\pi)^R \prod_{r=1}^R \frac{e^{x_r}}{(1+e^{x_r})^2}$.
- *Probit transformation.* The scaled probit transformation is $x_r = \sqrt{2}\,\mathrm{erf}^{-1}(\pi^{-1}t_r - 1)$ which implies $t_r(x_r) = 2\pi\Phi(x_r)$ where Φ is the cumulative distribution function of the standard normal distribution. This then gives a Jacobian $J(\boldsymbol{x}) = (2\pi)^R \prod_{r=1}^R \phi(x_r)$ where ϕ is the probability density function of the standard normal distribution.

It is possible to verify that, for both logit and probit transformations, the maximum of the transformed integrand is located at $\boldsymbol{x} = 0$. This is because, in both transformations, this choice yields $\boldsymbol{t} = [\pi, \dots, \pi]$ so that the Norlund-Rice integrand becomes $\Re h_{\boldsymbol{\Theta}(\mathbf{0})}(N) = \sum_{\substack{\boldsymbol{n}\geq 0:\\ |\boldsymbol{n}|=N}} H_{\boldsymbol{\theta}}(\boldsymbol{n}) = h_{\boldsymbol{\theta}\mathbf{1}}(N)$, where $\mathbf{1} = (1, \dots, 1)^T$ is a column vector of ones.

Knowing that the maximum is at the origin, under a Gaussian approximation centered at that location, we may approximate the Norlund-Rice integral (31) as

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi)^R} \int_{\mathbb{R}^R} h_{\boldsymbol{x}}(N) J(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{(2\pi)^{-R/2}}{\sqrt{\det(-\boldsymbol{A}(\mathbf{0}))}} h_{\boldsymbol{\theta}\mathbf{1}}(N) J(\mathbf{0}) \tag{36}$$

Here, $\boldsymbol{A}(\mathbf{0})$ is the Hessian matrix of the function $h_{\boldsymbol{x}}(N)J(\boldsymbol{x})$, evaluated at the origin in the transformed space. This expression is based on a standard closed-form for high-dimensional Gaussian integrals used in Laplace's approximation [45]. It should be noted that this is in general an heuristic, since small fluctuations are possible far from the maximum that make, in general, the integrand not perfectly Gaussian. However, this is generally not an impediment to accurate numerical evaluation, since the exponentially decaying Jacobian rapidly dampens such fluctuations.

Simplex form. In the case of the simplex form (33), we transform the integrand using an additive logistic transformation [46, 21]. Appendix A.11 develops derivations up to finding exact expressions for the partial derivatives of the integrand. We then sample the transformed integrand using Monte Carlo integration, centered at the maximum and parameterized with the Hessian of the integrand.

5.2.2. **GM cubature rules.**

The application of GM rules to fixed-rate normalizing constants is known [43, 21] and our result here is only to generalize their use to LLD models via (33). For polynomial integrands, cubature rules can also produce exact results, for large enough degree. In particular, a GM cubature rule of degree $2S+1$, involves sampling the integrand at a fixed collection points, so that we may write

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{\prod_{r=1}^R N_r!} \sum_{i=0}^{S} w_i \sum_{\substack{\boldsymbol{b}\geq\mathbf{0}:\\ |b|=S-i}} \sum_{\mathbf{0}\leq\boldsymbol{v}<\boldsymbol{s}} \alpha_{\boldsymbol{v}} \boldsymbol{\Delta}_{l_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{l}}^{\boldsymbol{v}} \prod_{r=1}^{R} \left(\sum_{k=1}^{M} \sigma_{k,r}(l_k + l_0 b_{ik}) \right)^{N_r} \tag{37}$$

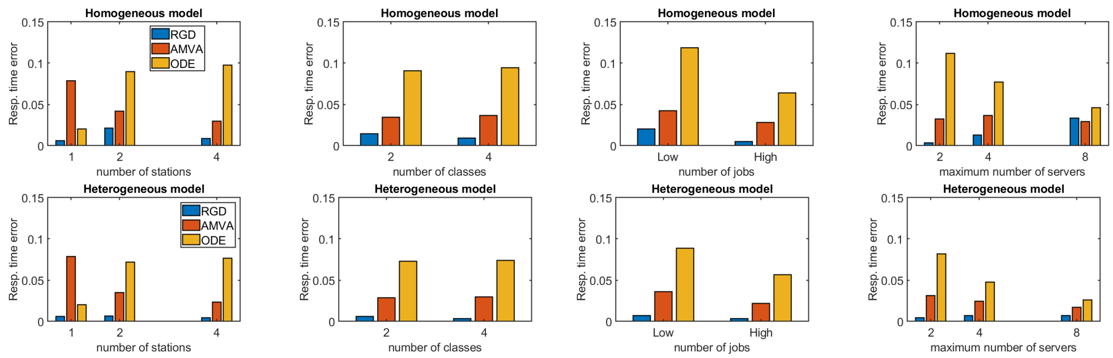


Figure 1: Percentage error of the proposed RD heuristic on tractable, exactly solvable, models.

where $b_{ik} = (2b_k + 1)/(2S + M - 2i)$ and in which we use the weights

$$w_i = (-1)^i 2^{-2S} \frac{(2S + K - 2i)^{2S+1}}{i!(2S + K - i)!}$$

For a polynomial of order N, as in the case of the integrand of (33), the GM method returns exact estimates if $S = \lceil (N-1)/2 \rceil$, in $\mathcal{O}(N^{M+1})$ time and $\mathcal{O}(1)$ space, slightly improving over the complexity of the RECAL algorithm for LLD models [47].

6. Numerical results

6.1. Methodology

In this section, we evaluate in separate studies the accuracy in approximating mean value indexes and normalizing constants for the methods we have proposed in the earlier sections.

For mean value analysis, we focus on the system response time $C_r(\boldsymbol{N}) = N_r/X_r(\boldsymbol{N}) - \theta_{0,r}$, as this is less sensitive than the system throughput $X_r(\boldsymbol{N})$ to the think times $\theta_{0,r}$, and thus challenging to approximate under large think time values. We obtain estimates of $C_r(\boldsymbol{N})$ with the methods shown in Table 3. For an approximation A, and a baseline method B (in our case, either EX or SIM), the error metric used in the validation is the mean absolute percentage error (MAPE) over the job classes: $\varepsilon = R^{-1} \sum_{r=1}^{R} \left|1 - C_r^A(\boldsymbol{N})/C_r^B(\boldsymbol{N})\right|$.

Since the normalizing constant grows rapidly beyond the floating-point range, in the validation we attempt to approximate $\log H_{\boldsymbol{\theta}}(\boldsymbol{N})$. Estimates are obtained using the methods shown in Table 4. For an approximation method A we evaluate the absolute percentage error $\varepsilon = \left|1 - \log H_{\boldsymbol{\theta}}^A(\boldsymbol{N})/\log H_{\boldsymbol{\theta}}^B(\boldsymbol{N})\right|$, and set EX as the baseline B.

Additional implementation remarks for the methods listed in Table 3 and Table 4 are given in Appendix A.12.

6.2. Random models

We have generated two large sets of random queueing network models of varying sizes, as detailed in Table 5. In total, this resulted in over 100,000 models. Each network consists of

Table 3: Mean value analysis methods in Section 6

RD	(24), where $X_r^{\boldsymbol{\sigma}}$ is evaluated with AQL [19]
AMVA	Queue-dependent AMVA algorithm [29]
ODE	Kurtz's mean field approximation [48]
EX	Exact solution [16]
SIM	Discrete-event simulation

Table 4: Normalizing constant methods in Section 6.3.3

NRL	(36), with $\boldsymbol{t}(\boldsymbol{x})$ defined as a logit transformation
NRP	(36), with $\boldsymbol{t}(\boldsymbol{x})$ defined as a probit transformation
RDA	Asymptotic approximation based on RD throughput (c.f Appendix A.12)
RDH	Heuristic in (22)
EX	Exact solution [13]

M multi-server stations, each with s_k homogeneous servers. Note that if $M = 1$, the network consists also of a think time captured by an infinite server station, so there are always at least two stations. Stations are visited by a population of N jobs equally split across R classes, possibly up to a small remainder that we allocate to class R. The population is constrained to feature a job in each class at least and it is assigned one of two values associated with low and high load (N_{lo}, N_{hi}). A value $s^{\max}$ is assigned to each model so that either we choose at random the number of servers $s_k \in [1, \min(N, s^{\max})]$ at each station, termed the *heterogeneous* case, or we set $s_1 = ... = s_M = \min(N, s^{\max})$ at all nodes, termed the *homogeneous* case. For each combination of parameters, the demand matrix $\boldsymbol{\theta}$ is varied within a fixed pool of pre-generated random matrices, sampled from a uniform distribution. Think times are also sampled uniformly at random.

Table 5: Random models used for validation. The population levels are $N_{lo} = R+M\cdot s^{\max}$ and $N_{hi} = R+2M\cdot s^{\max}$.

Parameter	*Tractable Models*	*Large Models*
Demand matrix pool size ($\boldsymbol{\theta}$)	1000	100
Number of stations (M)	1,2,4	8,16,32
Number of classes (R)	2,4	8,16,32
Total population (N)	$N_{lo}, N_{hi},$	N_{lo}, N_{hi}
Maximum LLD limit ($s^{\max}$)	1,2,4,8	1,2,4,8
Random s_k?	true, false	true, false
Baseline solution	EX	SIM
Number of models	*96000*	*14400*

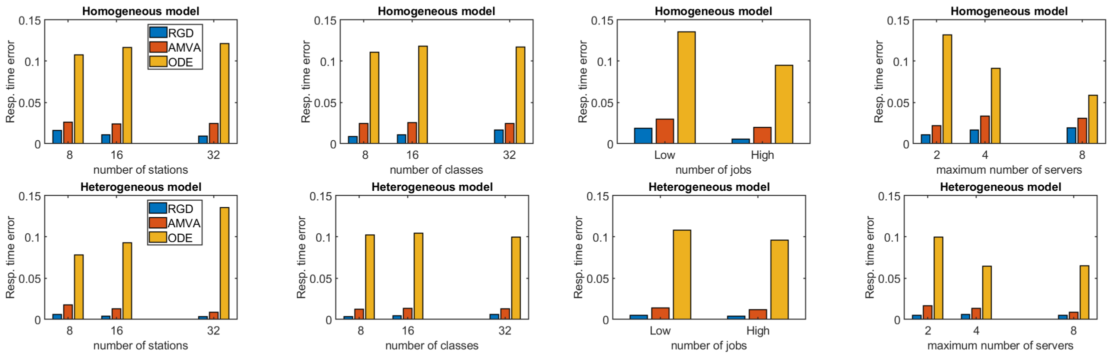


Figure 2: Percentage error of the proposed RD heuristic on large-scale models.

6.3. Validation results

6.3.1. Mean value analysis results

Figure 1 and Figure 2 compare the MAPE metric on the system response times for the mean value analysis approaches, namely RD, AMVA, and ODE. It is clear that RD delivers the most accurate results of the three methods in all instances, with an average error of 1.20% on the tractable models and 1.21% on large models. On the same instances, AMVA has respectively 3.56% and 2.49% errors, while ODE incurs 9.29% and 11.53% errors. These differences are significant, e.g., we show later in Section 7 how the increased precision of RD in this case results in far larger gains compared to AMVA and fluid ODEs in approximating non-product-form models.

Errors grow with $s^{\max}$, slightly more in the homogeneous case than in the heterogeneous case. The results also indicate that for large M and $s^{\max}$, the error of RD decreases compared to the one showed for smaller models. Accuracy is fairly insensitive to the number of classes R and moderately sensitive to the total population N, with higher loads being better for RD.

We now report timing results for the experiments described in Table 5. The timings are run using a AMD Ryzen 7 2700X 8-core desktop computer with 32GB of RAM. Each measurement captures the time to solve a model with the given number of stations, averaged over the different random seeds as well as different values of number of classes, total population, and LLD limit shown in Table 5. We compare AMVA, fluid ODEs, RD and also discrete-event simulation using the Java Modelling Tools suite (SIM) [38]. The latter tool leverages transient detection techniques to automatically terminate the simulation at steady-state. The fluid ODEs and AMVA implementations are those available in the open source LINE tool [49].

Table 6 reports the timing results and indicates that simulation and fluid ODEs incur the largest overhead. The larger times incurred by the fluid ODE result are explained by the cost of the setup and numerical solution of the ODEs, we point to Appendix A.12 for implementation details. Faster implementations may be possible in individual instances at the expense of accuracy. Conversely, AMVA and RD display rather similar running times, which show that RD is more

Table 6: Timing results for the experiments in Table 5

Num. of stations	Time [s]			
M	RD	AMVA	ODE	SIM
1	0.0024	0.0116	0.2525	1.4099
2	0.0067	0.0127	0.3344	1.6630
4	0.0157	0.0167	0.5320	2.1726
8	0.0376	0.0252	1.0406	3.1247
16	0.1116	0.0473	2.2294	5.1678
32	0.3051	0.1282	4.8575	9.5968

efficient on smaller models than AMVA, while with more stations the growth rate of AMVA and RD appears similar. We also remark that memory consumption was negligible with for methods.

Summarizing, the proposed RD method improves over the state of the art and it is applicable to both small and large scale models. Timing results indicate that RD tends to be faster than all other methods in networks with few stations and matches the timing growth of the faster of the baseline methods, namely AMVA.

6.3.2. **Further mean value analysis examples**

We here further illustrate the performance of the methods when the model approaches a "fluid" large-scale regime, i.e., with thousands of jobs, many stations and many servers. We refer to these examples as many-server models. We generate a set of 1102 random random models with $M \in \{1,2,4,8,16,32\}$ stations, $R \in \{1,2,4\}$, $s_{max} \in \{16,32,64,128\}$. As before, the population levels are $N_{lo} = R + M \cdot s^{\max}$ and $N_{hi} = R + 2M \cdot s^{\max}$, so that the large population reached in these experiments is $N = 8196$ jobs. The demands are generated randomly, in a similar fashion as for the random instances we have studied in the previous sections.

Stable numerical computation is achieved by approximating in several cases the correction factor $\gamma(\boldsymbol{N})$ using the single-class LLD asymptotics in (9), which takes time and space both independent of N. The asymptotic formula is used in approximating models with $N \geq 64$, while for small models we use the exact convolution algorithm to compute the correction factor [50]. For increased efficiency, the RD method is applied by approximating $\gamma(N)$ and $\gamma(N - 1_r)$ by computing only the last term of their defining summations, which is the largest one. The throughput for the fixed-rate model in the RD approximation is obtained with AQL [19].

The results are given in Table 7. As expected, ODEs are highly accurate in these models as they are close to the assumptions for which fluid approximations become exact. However, as the model scale grows, RD becomes increasingly competitive and more accurate for the large network sizes $M = 32$. Moreover, timing results indicate that RD is over 40 times faster than fluid solutions.

Table 7: Accuracy and timing results for many-server models

Num. of stations	MAPE [s]			Num. of stations	Time [s]		
M	RD	AMVA	ODE	M	RD	AMVA	ODE
1	0.0939	2278.4	0.0079	1	0.0038	0.0140	0.4238
2	0.0424	1953.7	0.0072	2	0.0161	0.0167	0.4814
4	0.0257	1041.5	0.0039	4	0.0308	0.0225	0.7203
8	0.0116	627.5	0.0033	8	0.0559	0.0332	1.3979
16	0.0051	374.6	0.0056	16	0.0863	0.0628	2.8494
32	0.0041	210.5	0.0139	32	0.2048	0.1710	6.4122

As remarked before, such differences vary with implementation details but hint at high efficiency for the RD method. On the contrary, in these experiments AMVA faces numerical instabilities in the soft-min approximation of the scaling functions. Similarly to what discussed in Section 2.6, we have attempted to resolve these issues by adjusting the soft-min parameter κ, but a single assignment of the parameter was not able to sanitize all the random models considered in the experiments. As such, AMVA displays large errors.

6.3.3. **Normalizing constant analysis results**

Additional validation results for the formulas we have developed for normalizing constants are given in Appendix A.13. We show therein that our integral forms provide the best performing methods for LLD normalizing constant analysis. In particular, the Norlund-Rice integral form instantiated with a logit transformation generally performs best, with an overall error in the experiments of just 6.46% for NRL, based on the logit transformation, as opposed to 11.72% for NRP, 0.49% for RHA, and 26.62% for RHD. We also observe that, as the number of classes R grows, both methods based on the Norlund-Rice integral (NRL and NRP) are less accurate than RHD, which may therefore be preferred for large R. Conversely, the NRP method, based on the probit transformation and the RHA method are dominated in accuracy by NRL and RHD.

In addition to the results shown on random models, we also report in Appendix A.13 further examples that demonstrate the speed of convergence and accuracy of the Norlund-Rice and simplex approximations to the integral form. Some of these examples have already been presented in a preliminary version of this work [22].

7. Applications

7.1. Approximations for non-product-form systems

In this section, our main aim is to show that the increased accuracy of RD with respect to AMVA in computing mean-value indexes can translate into large accuracy gains when the RD method is iteratively applied to approximate challenging non-product form systems.

We consider as a prototypical example a finite repairmen model, i.e., a cyclic multiclass closed queueing network with an infinite server node and a multi-server FCFS queue with non-exponential service time. This model has two core features that cannot be treated exactly by product-form analysis: multiclass FCFS scheduling and non-exponential service at a FCFS station. The FCFS queue, being multi-server, is here seen as a LLD station, but due to its non-product-form nature it is beyond the assumptions of the class of systems we have studied so far.

The performance of the RD approximation is illustrated on four non-product-form examples:

- **Example I (Exp, 2 classes)**. $R = 2$ classes. The FCFS queue has 2 servers and exponential service times with mean r in class r. The infinite server has mean think time $\theta_{0,r} = r$, $\forall r$.
- **Example II (Exp, 8 classes)**. $R = 8$ classes. The FCFS queue has 2 servers and exponential service times with mean r in class r. The infinite server has mean think time $\theta_{0,r} = r$, $\forall r$.
- **Example III (Increasing c^2)**. $R = 8$ classes. The FCFS queue has 2 servers and hyper-exponential service times with mean r and squared coefficient of variation $c^2 = r$ in class r. The infinite server has mean think time $\theta_{0,r} = r$, $\forall r$.
- **Example IV (Decreasing c^2)**. $R = 8$ classes. The FCFS queue has 2 servers and hyper-exponential service times with mean r and squared coefficient of variation $c^2 = R - r$ in class r. The infinite server has mean think time $\theta_{0,r} = r$, $\forall r$.

We study the total population range $N \in \{5, 10, 15, 20\}$, and set the class populations all equal to $N_r = \lceil \frac{N}{R} \rceil$, $\forall r$. We seek in particular to quantify mean and maximum absolute percentage errors on the mean response times

$$C_{k,r}(\boldsymbol{N}) = \frac{Q_{k,r}(\boldsymbol{N})}{X_r(\boldsymbol{N})} \tag{38}$$

at the FCFS queue, for all classes $r = 1, \dots, R$.

AMVA does not require a new approach to deal with this case as the standard FCFS-AMVA approximation [51] applies. The RD and fluid ODE methods we have studied before in LLD mean-value analysis are instead adapted to deal with multi-class FCFS in the following way. At start, the FCFS station is treated as a processor-sharing queue, so that the model admits a product-form solution and the approximations can be applied to obtain an initial solution. Afterwards, a fixed point iteration is used to sequentially correct the service demands $\theta_{k,r}$ at the FCFS station ($k = 1$), which is analyzed at each iteration again as a PS queue but with the newly formed demands. The process is repeated until convergence of all the $\theta_{k,r}$ values within a 10^{-4} tolerance, after which the response time errors are computed. The demand update at each iteration is based on a FCFS multiclass approximation formula in [49], which is a weighted polynomial interpolation of the approximate decay rates for the multiserver G/G/1 nodes [52, Section 7.2.2] and of the asymptotic

Table 8: FCFS multiclass models approximation using the RD method

	Exp - 2 classes						*Exp - 8 classes*					
	Average error			Maximum error			Average error			Maximum error		
N	RD	AMVA	ODE	RD	AMVA	ODE	RD	AMVA	ODE	RD	AMVA	ODE
5	0.0551	0.1046	0.0915	0.1542	0.2284	0.2095	0.0522	0.3289	0.1722	0.1611	0.8841	0.6522
10	0.0370	0.1439	0.1366	0.1186	0.2992	0.2838	0.1289	0.3635	0.2101	0.2857	0.8100	0.7378
15	0.0482	0.1477	0.1527	0.1188	0.2972	0.3123	0.1279	0.3627	0.2091	0.2838	0.8098	0.7375
20	0.0509	0.1452	0.1589	0.1158	0.2894	0.3261	0.1682	0.3526	0.2171	0.3513	0.7495	0.7537
	Increasing c^2 - 8 classes						*Decreasing c^2 - 8 classes*					
	Average error			Maximum error			Average error			Maximum error		
N	RD	AMVA	ODE	RD	AMVA	ODE	RD	AMVA	ODE	RD	AMVA	ODE
5	0.0600	0.0748	0.0936	0.1240	0.2395	0.2116	0.0641	0.0836	0.0859	0.1289	0.2030	0.2051
10	0.0292	0.1252	0.1365	0.1070	0.3020	0.2945	0.0346	0.1348	0.1327	0.1207	0.2841	0.2722
15	0.0485	0.1427	0.1522	0.1200	0.3084	0.3078	0.0437	0.1402	0.1522	0.1130	0.2892	0.3173
20	0.0477	0.1395	0.1591	0.1103	0.2953	0.3299	0.0518	0.1433	0.1552	0.1225	0.2936	0.3111

decay rate of Kobayashi's diffusion approximation [53]. While a formal proof of convergence is neither available for this method nor for the similar methods in the literature, we have not observed cases where the demands failed to converge after a sufficiently large number of iterations. Similar iterative schemes as the one described above are known in the literature [52, 49]. What is novel here is that, at each iteration, the approximate PS model with corrected demands is solved using RD, as opposed to methods such as AMVA or fluid ODEs used in earlier works.

Results are shown in Table 8. As we can see, the increased accuracy of the RD method produces a considerable increase in the precision of the approximation, which achieves significantly less error than AMVA and ODE. The average is in the worst case about 16.82% for RD, whereas the error grows to 35.2% for AMVA and 21.7% for ODE. The largest populations deliver the largest gaps, with RD featuring at worst 35% error, against about 75% for AMVA and ODE. It is particularly significant to observe the large gap between RD and the ODE method, which is based on the same iteration as RD.

Overall, this study clearly highlights the advantage of resorting to the RD approximation for such models. It should be further noted that the case we have covered here is applicable to a wide range of software modeling applications. This is because the class of layered queueing network models [54], which is widely used for software system performance evaluation, are often simply as a collection of inter-related closed multiclass FCFS models similar to the one studied in this section.

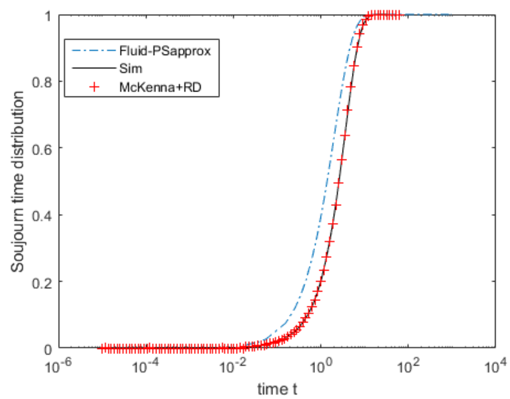


Figure 3: Sojourn time distribution analysis for a closed model of a FCFS multi-server station. McKenna's approximation can now be efficiently computed using the LLD correction factor we have derived in (16).

7.2. Sojourn time distribution analysis

As a further application, we consider the issue of estimating sojourn time distributions in multiclass closed queueing networks. Such distributions are relevant to system design, for example, where they provide a foundation for service-level agreement analysis [55].

Sojourn time distribution analyses are often performed using fluid ODE approximations [56] or analytical product-form methods [57, 58]. We here focus on the multiclass FCFS analysis technique given by McKenna in [58], which being exact provides a reliable evaluation technique. This method is not used in practice since it requires tens of independent normalizing constant evaluations, one for each point of the distribution, to produce an estimate of the distribution. Thus, it is generally very slow, and quickly becomes infeasible as the model complexity grows in terms of classes or stations. This situation illustrates a case where the RD approximation can enable the practical use of normalizing-constant methods like McKenna's.

A numerical example is given in Figure 3. The model has think times $\theta_{01} = 5$ and $\theta_{02} = 5$, and a first-come first-served station with mean demands $\theta_{11} = 5$ and $\theta_{12} = 3$, and $N_1 = N_2 = 2$ jobs within each class. The figure plots the sojourn time distribution for class-1 jobs at the queueing station obtained via simulation, fluid ODEs arising from the same method studied in the previous section, and an analytical evaluation produced by means of McKenna's method with the formula (16). It is clear that McKenna's method is nearly exact, and far more accurate than a fluid ODE approximations. The latter struggles to capture the non-preemptive characteristics of FCFS, which are instead fully accounted for in McKenna's normalizing-constant based formula. Overall, this provides additional evidence of the significance of the proposed LLD correction factor.

8. State-of-the-art

Exact analytical techniques for performance metrics, such as the convolution algorithm (CA) [50, 13], can be applied to mean performance metrics for both fixed-rate and LLD models. In general, the complexity of CA grows combinatorially with the number of classes R, requiring the use of approximations in large models. Moreover, working with normalizing constants implies implementation overheads related to numerical stability, since such constants can rapidly exceed floating-point range limits. RECAL is an alternative used also in LLD models [59, 47], but it shares similar difficulties with CA when the number of stations M is moderate-to-large. To overcome numerical issues, in the fixed-rate setting it is common to solve closed networks using mean-value analysis (MVA) [16]. However, MVA does not fully resolve instabilities in the case of LLD models [18], where an exact variant exists (MVA-LD) [14], besides being considerably more expensive than in the fixed-rate case as its complexity remains exponential in the number of classes [14, 15, 60]. Conditional MVA (CMVA) is exact and addresses this problem [33], but the method is effective only in single-class models as it becomes very expensive in the multi-class LLD setting.

Approximate MVA (AMVA) methods for LLD models were first introduced in [61, 62], but these methods are difficult to implement and overall brittle. This is a significant difference compared to AMVA in the fixed-rate setting where methods are robust and highly accurate, e.g., AQL [19], Linearizer [41]. The authors in [63] and [64] address this limitation by fixed point interpolations, for models that include multi-server nodes. However, these methods do not address the LLD case in full generality. Recently, the Stable MVA (SMVA) presents a new scheme valid for LLD models, restricted to a subclass of models where the scaling factors satisfy certain monotonicity assumptions that do not always hold [65].

More recently, [29] describes an iterative scheme, called QD-AMVA, that is applicable to general LLD models, but that is only guaranteed to converge to a unique solution under monotonicity assumptions on the shape of the scaling functions α_k. Similar conditions exist in also in the SMVA method mentioned above. In addition, QD-AMVA requires the scaling function to be replaced by a continuous interpolation, in particular the soft-min function for multi-server queues. Such functions introduce small numerical perturbations, which can affect the monotonicity of the scaling function required for convergence. As such, in general, there exist cases where QD-AMVA operates outside the assumptions and does not convergence. For example, numerical experiments with QD-AMVA suggest that under scaling functions such as $\alpha_k(n_k) = e^{n_k}$ it converges, while it does not terminate after 10^6 iterations when $\alpha_k(n_k) = e^{-n_k}$. The RD method proposed in this paper has been shown to have greater average accuracy than QD-AMVA. Moreover, it is guaranteed to converge if one uses an AMVA method with convergence guarantee (e.g., [40]) to obtain the X_r^{σ} terms.

Bounding and asymptotic approximations also exist in the single-class LLD setting [66, 37, 67, 36].

Extensions to closed non-product-form LLD models of techniques such as Marie's method are also actively studied [68]. However, similar results are not available for multiclass LLD systems.

9. Conclusion

In this paper, we have developed a range of novel computational tools for estimating in LLD closed queueing networks both mean performance metrics and normalizing constants used in probabilistic analysis. Noticeably, we have show that the solution of a LLD model can be exactly obtained, thanks to a correction factor, from the solution of a model with fixed-rate stations. This result has been leveraged to define a novel approximate mean-value analysis algorithm, called reduction (RD) heuristic. Additional results obtained in this paper have been shown to enable the efficient and accurate estimation of LLD normalizing constants by means of two integral forms. An implementation of the RD heuristic and of the proposed integral forms is available in the open source LINE solver[2] [49].

Possible directions for future research include the generalization of the present results to more general models, such as those with class-dependent scaling functions or with a mix of open and closed job classes.

References

[1] D. Menasce', V. A. F. Almeida, Capacity Planning for Web Services., Prentice Hall, 2002.

[2] L. Zhang, D. G. Down, Approximate mean value analysis for multi-core systems, in: Proc. of SPECTS, 2015, pp. 1:1–1:8.

[3] P. Castagno, V. Mancuso, M. Sereno, M. A. Marsan, Closed form expressions for the performance metrics of data services in cellular networks, in: Proc. of INFOCOM, IEEE, 2018, pp. 585–593.

[4] L. Xia, B. Shihada, A Jackson network model and threshold policy for joint optimization of energy and delay in multi-hop wireless networks, Eur. J. Oper. Res 242 (3) (2015) 778–787.

[5] P. Legato, R. M. Mazza, Queueing analysis for operations modeling in port logistics, Maritime Business Review (2019).

[6] L. Xia, B. Shihada, Max-min optimality of service rate control in closed queueing networks, IEEE Trans. Automat. Contr 58 (4) (2013) 1051–1056.

[2] `http://line-solver.sf.net/`

[7] I. F. Akyildiz, C. Strelen, Moment analysis for load-dependent mixed product form queueing networks, IEEE Trans. Communications 39 (6) (1991) 828–832.

[8] A. E. Krzesinski, Multiclass queueing networks with state-dependent routing, Perform. Eval. 7 (2) (1987) 125–143.

[9] I. F. Akyildiz, A. Sieber, Approximate analysis of load dependent general queueing networks, IEEE Transactions on Software Engineering 14 (11) (1988) 1537–1545.

[10] E. Sönmez, A. Scheller-Wolf, N. Secomandi, An analytical throughput approximation for closed fork/join networks, INFORMS Journal on Computing 29 (2) (2017) 251–267.

[11] K. M. Chandy, U. Herzog, L. Woo, Parametric analysis of queueing networks, IBM J. Res. and Devel. 19 (1975) 36–42.

[12] S. S. Lavenberg, A perspective on queueing models of computer performance, Perform. Eval. 10 (1) (1989) 53–76.

[13] M. Reiser, H. Kobayashi, Queueing networks with multiple closed chains: Theory and computational algorithms, IBM J. Res. Dev. 19 (3) (1975) 283–294.

[14] C. H. Sauer, Computational algorithms for state-dependent queueing networks, ACM Trans. Comp. Syst. 1 (1983) 67–92.

[15] S. C. Bruell, G. Balbo, P. V. Afshari, Mean value analysis of mixed, multiple class BCMP networks with load dependent service stations, Perform. Eval. 4 (1984) 241–260.

[16] M. Reiser, S. S. Lavenberg, Mean-value analysis of closed multichain queueing networks, JACM 27 (2) (1980) 312–322.

[17] P. Cremonesi, P. J. Schweitzer, G. Serazzi, A unifying framework for the approximate solution of closed multiclass queuing networks, IEEE Trans. Comp 51 (2002) 1423–1434.

[18] M. Reiser, Mean-value analysis and convolution method for queue-dependent servers in closed queueing networks, Perform. Eval. 1 (1981) 7–18.

[19] J. Zahorjan, D. L. Eager, H. M. Sweillam, Accuracy, speed, and convergence of approximate mean value analysis, Perform. Eval. 8 (4) (1988) 255–270.

[20] P. Flajolet, R. Sedgewick, Mellin transforms and asymptotics: Finite differences and Rice's integrals, Theoretical Computer Science 144 (1–2) (1995) 101–124.

[21] G. Casale, Accelerating performance inference over closed systems by asymptotic methods, POMACS 1 (1) (2017) 8:1–8:25.

[22] G. Casale, P. G. Harrison, O. W. Hong, Novel solutions for closed queueing networks with load-dependent stations, SIGMETRICS Perform. Evaluation Rev 47 (2) (2019) 30–32.

[23] G. Bolch, S. Greiner, H. de Meer, K. S. Trivedi, Queueing Networks and Markov Chains, Wiley, 2006.

[24] F. Baskett, K. M. Chandy, R. R. Muntz, F. G. Palacios, Open, closed, and mixed networks of queues with different classes of customers., JACM 22 (1975) 248–260.

[25] J. Zahorjan, The distribution of network states during residence times in product form queueing networks, Perform. Eval. 4 (2) (1984) 99–104.

[26] P. J. Denning, J. P. Buzen, The operational analysis of queueing network models, ACM Computing Surveys 10 (3) (1978) 225–261.

[27] H. R. Bahadori, A hierarchical approximation algorithm for large multichain product form queueing networks, Rept. No. UCB/CSD 87/382, University of California, Berkeley (Nov. 1987).

[28] D. Mitra, J. McKenna, Asymptotic expansions for closed markovian networks with state-dependent service rates, JACM 33 (1985) 568–592.

[29] G. Casale, J. F. Pérez, W. Wang, QD-AMVA: Evaluating systems with queue-dependent service requirements, Perform. Eval. 91 (2015) 80–98.

[30] A. Bertozzi, J. McKenna, Multidimensional residues, generating functions, and their application to queueing networks, SIAM Review 35 (2) (1993) 239–268.

[31] Y. Kogan, M. Shenfild, Asymptotic solution of generalized multiclass Engset model, Teletraffic Science and Engineering 1 (1994) 1239–1249.

[32] C. H. Sauer, K. M. Chandy, Computer systems performance modelling, Prentice-Hall, Inc (1982).

[33] G. Casale, On single-class load-dependent normalizing constant equations, in: Proc. of the 3rd Conf. on Quantitative Evaluation of Systems (QEST), IEEE Press, 2006, pp. 333–342.

[34] J. J. Gordon, The evaluation of normalizing constants in closed queueing networks, Operations Research 38 (5) (1990) 863–869.

[35] S. Lam, Dynamic scaling and growth behavior of queueing network normalization constants, JACM 29 (2) (1982) 492–513.

[36] D. K. George, C. H. Xia, M. S. Squillante, Exact-order asymptotic analysis for closed queueing networks, J. Applied Probability 49 (2) (2012).

[37] A. Birman, Y. Kogan, Asymptotic evaluation of closed queueing networks with many stations, Comm. in Statistics – Stochastic Models 8, 3 (1992) 543–563.

[38] M. Bertoli, G. Casale, G. Serazzi, Java Modelling Tools: an open source suite for queueing network modelling and workload analysis, in: Proc. of the 3rd Conf. on Quantitative Evaluation of Systems (QEST), IEEE, 2006, pp. 119–120.

[39] G. Casale, An efficient algorithm for the exact analysis of multiclass queueing networks with large population sizes, in: Proc. of joint ACM SIGMETRICS/IFIP Performance, ACM Press, 2006, pp. 169–180.

[40] K. R. Pattipati, M. M. Kostreva, J. L. Teele, Approximate mean value analysis algorithms for queuing networks: Existence, uniqueness, and convergence results, JACM 37 (1990) 643–673.

[41] K. M. Chandy, D. Neuse, Linearizer: A heuristic algorithm for queuing network models of computing systems, Comm. of the ACM 25 (2) (1982) 126–134.

[42] P. Blanchard, D. J. Higham, N. J. Higham, Accurately computing the log-sum-exp and softmax functions, IMA Journal of Numerical Analysis (08 2020).

[43] A. Grundmann, H. M. Moller, Invariant integration formulas for the n-simplex by combinatorial methods, SIAM Journal on Numerical Analysis 15 (2) (1978) 282–290.

[44] K. Ross, D. Tsang, J. Wang, Monte carlo summation and integration applied to multiclass queueing networks., JACM 41 (6) (1994) 1110–1135.

[45] R. E. Kass, L. Tierney, J. B. Kadane., The validity of posterior expansions based on laplace's method., Bayesian and likelihood methods in stat. and econ. 7 (1990) 473.

[46] J. Aitchison, The statistical analysis of compositional data, Journal of the Royal Statistical Society. Series B (Methodological) (1982) 139–177.

[47] J. McKenna, Calculating joint queue length moments with RECAL, Commun. in Statistics – Stochastic Models 7, 1 (1991) 47–66.

[48] T. G. Kurtz, Limit theorems for sequences of jump markov processes approximating ordinary differential processes, Journal of Applied Probability 8 (2) (1971) 344–356.

[49] G. Casale, Integrated performance evaluation of extended queueing network models with LINE (2020).

[50] J. P. Buzen, Computational algorithms for closed queueing networks with exponential servers, Comm. of the ACM 16 (9) (1973) 527–531.

[51] D. L. Eager, D. Sorin, M. K. Vernon, AMVA techniques for high service time variability, in: Proc. of ACM SIGMETRICS, ACM Press, 2000, pp. 217–228.

[52] N. Gautam, Analysis of Queues, CRC Press, 2012.

[53] H. Kobayashi, Application of the diffusion approximation to queueing networks I: equilibrium queue distributions, J. ACM 21 (2) (1974) 316–328.

[54] G. Franks, T. Omari, C. M. Woodside, O. Das, S. Derisavi, Enhanced modeling and solution of layered queueing networks, IEEE Trans. Software Eng 35 (2) (2009) 148–161.

[55] D. A. Menasce, V. A. F. Almeida, L. W. Dowdy, Performance by Design : Computer Capacity Planning By Example, Prentice Hall, 2004.

[56] J. F. Pérez, G. Casale, Assessing SLA compliance from palladio component models, in: Proc. of SYNASC, IEEE Computer Society, 2013, pp. 409–416.

[57] P. G. Harrison, W. J. Knottenbelt, Passage time distributions in large Markov chains, Vol. 30 of SIGMETRICS Performance Evaluation Review, ACM Press, New York, 2002, pp. 77–85.

[58] J. McKenna, Asymptotic expansions of the sojourn time distribution functions of jobs in closed product-form queuing networks, J. ACM 34 (4) (1987) 985–1003.

[59] A. E. Conway, N. D. Georganas, RECAL - A new efficient algorithm for the exact analysis of multiple-chain closed queueing networks, JACM 33 (4) (1986) 768–791.

[60] K. P. Hoyme, S. C. Bruell, P. V. Afshari, R. Y. Kain, A tree-structured mean value analysis algorithm, ACM Trans. on Computer Systems 4 (2) (1986) 175–185.

[61] J. Zahorjan, E. D. Lazowska, Incorporating load dependent servers in approximate mean value analysis, in: Proc. of ACM SIGMETRICS, ACM, 1984, pp. 52–62.

[62] A. E. Krzesinski, J. Greyling, Improved lineariser methods for queueing networks with queue dependent centres, in: Proc. of ACM SIGMETRICS, ACM, 1984, pp. 41–51.

[63] I. F. Akyildiz, H. Bolch, Mean value analysis approximation for multiple server queueing networks, Performance Evaluation 8 (1988) 77–91.

[64] A. Seidmann, P. J. Schweitzer, S. Shalev-Oren, Computerized closed queueing network models of flexible manufacturing systems: A comparative evaluation, Large Scale Systems 12 (2019) 91–107.

[65] L. Zhang, D. Down, Smva: A stable mean value analysis algorithm for closed systems with load-dependent queues, in: In: Puliafito A., Trivedi K. (eds) Systems Modeling: Methodologies and Tools. EAI/Springer Innovations in Communication and Computing. Springer, Cham, 2019.

[66] J. Shanthikumar, D. D. Yao, Throughput bounds for closed queueing networks with queue-dependent service rates, Perform. Eval. 9 (1988) 69 – 78.

[67] J. Anselmi, P. Cremonesi, Bounding the performance of BCMP networks with load-dependent stations, in: Proc. of MASCOTS, IEEE Computer Society, 2008, pp. 171–178.

[68] B. Y. Ekren, S. S. Heragu, Approximate analysis of load-dependent generally distributed queuing networks with low service time variability, Eur. J. Oper. Res 205 (2) (2010) 381–389.

[69] G. Casale, A note on stable flow-equivalent aggregation in closed networks, QUESTA 60 (3-4) (2008) 193–202.

[70] G. Casale, Exact analysis of performance models by the method of moments, Perform. Eval. 68 (6) (2011) 487–506.

[71] E. Koenigsberg, Cyclic queues, Operational Research Quarterly 9, 1 (1958) 22–35.

[72] V. Baldoni, N. Berline, J. A. D. Loera, M. Köppe, M. Vergne, How to integrate a polynomial over a simplex, Mathematics of Computation 80 (2011) 297–325.

[73] R. Kan, From moments of sum to moments of product, Journal of Multivariate Analysis 99 (3) (2008) 542 – 554.

[74] J. McKenna, Extensions and applications of RECAL in the solution of closed product-form queueing networks, Comm. in Statistics – Stochastic Models, Nr. 2 4 (1988) 235–276.

Appendix A.

A.1. Mean LLD performance metrics

In this section, we give additional results concerning a number of LLD performance metrics, completing the discussion in Section 2.4. In particular, we focus on mean queue-lengths and utilizations. Using Little's law, a formula for mean response times can be easily derived, as shown in (38).

A.1.1. Single class metrics

Consider a closed single-class LLD queueing network with population N. The work in [33] shows that it is possible to compute the expectation of an arbitrary function $f(n_i)$, n_i being the random variables for the number of jobs in a given station i, by studying a modified network that includes a suitably-defined LLD station called a *functional server*. This server can have negative LLD rates, but although the modified system does not admit a physical interpretation, a normalizing constant can still be obtained with these values. This normalizing constant can then be suitably scaled to find $E[f(n_i)]$ in the original model.

We now give the definition in [33] of the functional server when $f(n_i) = n_i + c$, for some arbitrary constant c. This can be used to compute the mean number of jobs $Q_i(N)$ in station i in the original network. If the original network has M queues, the functional server for station i in the modified system is a LLD station with index $M+1$, demand $\theta_{M+1} = \theta_i$, and with a scaling function α_{M+1} that is recursively computed from the scaling factors of station i as follows [33]

$$\alpha_{M+1}(1) = \begin{cases} \dfrac{\alpha_i(1)}{1+c} & n = 1 \\ \left(\dfrac{\prod_{v=1}^{n} \alpha_i(v)}{\prod_{h=1}^{n-1} \alpha_{M+1}(h)}\right) \left(1 - \sum_{k=1}^{n-1} \dfrac{\prod_{v=n-k+1}^{n} \alpha_i(v) - \prod_{v=n-k}^{n-1} \alpha_i(v)}{\prod_{h=1}^{k} \alpha_{M+1}(h)}\right)^{-1} & \text{otherwise} \end{cases}$$

where $c \neq -1$ is an arbitrary constant chosen so that $\alpha_{M+1}(n) \neq 0$, for all $1 \leq n \leq N$. As an example, the assignments $c = 0$ and $c = -1/2$ are often valid. In verifying the condition on c, we take the convention that if $\alpha_{M+1}(m) = +\infty$ for a population m, then the successive values are all set to $\alpha_{M+1}(n) = +\infty$, $\forall n > m$.

Denote by $H_{\boldsymbol{\theta}}^{-i}(N)$ the normalizing constant of the original network after removal of station i and let $H_{\boldsymbol{\theta}}^{(i)}(N)$ be the normalizing constant for the modified model that includes both station i and its functional server. The results in [33] then show that

$$Q_i(N) = \frac{H_{\boldsymbol{\theta}}^{(i)}(N)}{H_{\boldsymbol{\theta}}(N)} - 1 + c\left(\frac{H_{\boldsymbol{\theta}}^{-i}(N)}{H_{\boldsymbol{\theta}}(N)} - 1\right) = \frac{U_{M+1}^{(i)}(N)}{1 - U_{M+1}^{(i)}(N)} - cU_i(N)$$

where $U_i(N)$ is the utilization of i in the original model and $U^{(i)}_{M+1}(N)$ is the utilization of the functional server in the modified model. The derivation uses that the utilization of node i in the original model is

$$U_i(N) = 1 - \frac{H^{-i}_{\boldsymbol{\theta}}(N)}{H_{\boldsymbol{\theta}}(N)} \tag{A.1}$$

and for the functional server

$$U^{(i)}_{M+1}(N) = 1 - \frac{H_{\boldsymbol{\theta}}(N)}{H^{(i)}_{\boldsymbol{\theta}}(N)} \tag{A.2}$$

The results derived in the present paper for LLD normalizing constants can be readily applied to the above formulas to determine utilizations and mean queue-lengths in a LLD model.

A.1.2. Multiclass aggregate metrics

We now explain how the derivations in [33] generalize to multiclass networks. With very similar passages as in [33] one finds that

$$Q_i(\boldsymbol{N}) = \frac{H^{(i)}_{\boldsymbol{\theta}}(\boldsymbol{N})}{H_{\boldsymbol{\theta}}(\boldsymbol{N})} - 1 + c\left(\frac{H^{-i}_{\boldsymbol{\theta}}(\boldsymbol{N})}{H_{\boldsymbol{\theta}}(\boldsymbol{N})} - 1\right) \tag{A.3}$$

or in terms of mean values

$$Q_i(\boldsymbol{N}) = \frac{U^{(i)}_{M+1}(\boldsymbol{N})}{1 - U^{(i)}_{M+1}(\boldsymbol{N})} - cU_i(\boldsymbol{N}) \tag{A.4}$$

where $U^{(i)}_{M+1}(\boldsymbol{N})$ and $U_i(\boldsymbol{N})$ are now aggregate utilizations at stations $M+1$ and i summed across all classes, defined as

$$U_i(\boldsymbol{N}) = 1 - \frac{H^{-i}_{\boldsymbol{\theta}}(\boldsymbol{N})}{H_{\boldsymbol{\theta}}(\boldsymbol{N})} \tag{A.5}$$

and for the functional server

$$U^{(i)}_{M+1}(\boldsymbol{N}) = 1 - \frac{H_{\boldsymbol{\theta}}(\boldsymbol{N})}{H^{(i)}_{\boldsymbol{\theta}}(\boldsymbol{N})} \tag{A.6}$$

A.1.3. Multiclass per-class metrics

Given a LLD station i with utilization $U_i(\boldsymbol{N})$, we call *load-dependent demands* the quantities $\widehat{\theta}_{i,r}$ such that

$$U_i(\boldsymbol{N}) = \sum_{r=1}^{R} U_{i,r}(\boldsymbol{N}) = \sum_{r=1}^{R} \widehat{\theta}_{i,r} X_r(\boldsymbol{N}) \tag{A.7}$$

where $X_r(\boldsymbol{N})$ is the system throughout for class-r. In a fixed-rate system, the quantities $\widehat{\theta}_{i,r}$ match the mean service demands $\theta_{i,r}$. However, in a general load-dependent they also include the time-averaged effect of the varying service rate at which the jobs were processed, according to the specified scaling factors α_i.

Given a station i, let $\widehat{H}_{\boldsymbol{\theta}}(\boldsymbol{N} - 1_r)$ be the normalizing constant of a modified model with population $\boldsymbol{N} - 1_r$ and where the scaling factors for station i are modified as follows

$$\widehat{\alpha}_i(n) = \alpha_i(1 + n)$$

for $n = 1, ..., N-1$. The scaling factors for the other stations $k \neq i$ are instead left identical to the values $\alpha_k(n)$ in the original model. Let $\widehat{Q}_i(\boldsymbol{N})$ be the mean queue-length for station i in the modified model.

The following expressions are obtained in [69]

$$\widehat{\theta}_{i,r} = \frac{\theta_{i,r}}{\alpha_i(1)} \frac{\widehat{H}_{\boldsymbol{\theta}}(\boldsymbol{N}-1_r)}{H_{\boldsymbol{\theta}}(\boldsymbol{N}-1_r)} \tag{A.8}$$

$$Q_{i,r}(\boldsymbol{N}) = U_{i,r}(\boldsymbol{N})(1+\widehat{Q}_i(\boldsymbol{N}-1_r)) = \widehat{\theta}_{i,r} X_r(\boldsymbol{N})(1+\widehat{Q}_i(\boldsymbol{N}-1_r)) \tag{A.9}$$

Equation (A.8) allows one to compute the load-dependent demands, and therefore the per-class utilizations $U_{i,r}(\boldsymbol{N}) = \widehat{\theta}_{i,r} X_r(\boldsymbol{N})$. Equation (A.9) follows by an application of Little's law to the corresponding mean response time formula derived in [69]. The terms appearing in the above expressions can be efficiently computed by applying the methods developed in the present paper.

Since [69] does not report proofs for the multiclass case, we give here brief derivations of (A.8)-(A.9). The work in [69] shows the following load-dependent generalization of the convolution expression [13] for multiclass normalizing constants

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = H_{\boldsymbol{\theta}}^{-i}(\boldsymbol{N}) + \sum_{s=1}^{R} \frac{\theta_{i,s}}{\alpha_i(1)} \widehat{H}_{\boldsymbol{\theta}}(\boldsymbol{N}-1_s) \tag{A.10}$$

where $H_{\boldsymbol{\theta}}^{-i}(\boldsymbol{N})$ refers to a modification of the original model where station i is removed. Equation (A.8) follows immediately from (A.10), (A.5), (A.7), and the definition of per-class system throughput $X_s(\boldsymbol{N})$.

To obtain (A.9), we instead apply the following expression for load-dependent models in [70, Eq. (15)]

$$\frac{\partial H_{\boldsymbol{\theta}}(\boldsymbol{N})}{\partial \theta_{i,r}} = \frac{Q_{i,r}(\boldsymbol{N})}{\theta_{i,r}} H_{\boldsymbol{\theta}}(\boldsymbol{N}) \tag{A.11}$$

to each normalizing constant in (A.10). Equation (A.9) the follows with algebraic manipulations, after recognizing that $\frac{\partial}{\partial \theta_{i,r}} H_{\boldsymbol{\theta}}^{-i}(\boldsymbol{N}) = 0$ since the model no longer includes station i.

A.2. Proof of Theorem 1

Throughout the paper, we denote generating functions in calligraphic font and note that the generating function of the normalizing constant function $h_{\boldsymbol{\theta}}$ is

$$\mathcal{H}_{\boldsymbol{\theta}}(z) = \sum_{N=0}^{\infty} h_{\boldsymbol{\theta}}(N) z^N = \sum_{\boldsymbol{n} \geq \boldsymbol{0}} \prod_{k=1}^{M} \frac{(\theta_k z)^{n_k}}{\prod_{t=1}^{n_k} \alpha_k(t)} = \prod_{k=1}^{M} \sum_{n_k=0}^{\infty} \frac{(\theta_k z)^{n_k}}{\prod_{t=1}^{n_k} \alpha_k(t)} \tag{A.12}$$

This definition generalizes to a multiclass model with R classes as follows

$$\mathcal{H}_{\boldsymbol{\theta}}^{R}(\boldsymbol{z}) = \sum_{N_1=0}^{\infty} \dots \sum_{N_R=0}^{\infty} H_{\boldsymbol{\theta}}(\boldsymbol{N}) \prod_{r=1}^{R} z_r^{N_r} \tag{A.13}$$

The generating function of the normalizing constant $h_{\boldsymbol{\theta}}(n)$ is given by equation (A.12) as $\mathcal{H}_{\boldsymbol{\theta}}(N) = \prod_{j=1}^{M} F_j(z)$ where

$$\begin{aligned}
F_j(z) &= \sum_{n_j=0}^{\infty} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} \\
&= \sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} + \sum_{n_j=s_j}^{\infty} \frac{(\theta_j z)^{n_j}}{(\alpha_j(s_j))^{n_j-s_j} \prod_{t=1}^{s_j} \alpha_j(t)} \\
&= \sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} + \frac{(\theta_j z)^{s_j}}{\prod_{t=1}^{s_j} \alpha_j(t)} \sum_{n_j=s_j}^{\infty} \left(\frac{\theta_j z}{\alpha_j(s_j)}\right)^{n_j-s_j} \\
&= \sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} + \frac{(\theta_j z)^{s_j}}{\prod_{t=1}^{s_j} \alpha_j(t)} \frac{1}{1-\theta_j z/\alpha_j(s_j)} \\
&= \frac{1}{1-\theta_j z/\alpha_j(s_j)} \left(\sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} - \sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j+1}}{\alpha_j(s_j)\prod_{t=1}^{n_j} \alpha_j(t)} + \frac{(\theta_j z)^{s_j}}{\prod_{t=1}^{s_j} \alpha_j(t)} \right) \\
&= \frac{1}{1-\theta_j z/\alpha_j(s_j)} \left(\sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} - \sum_{n_j=0}^{s_j-2} \frac{(\theta_j z)^{n_j+1}}{\alpha_j(s_j)\prod_{t=1}^{n_j} \alpha_j(t)} \right) \\
&= \frac{1}{1-\theta_j z/\alpha_j(s_j)} \left(\sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} - \sum_{n_j=1}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\alpha_j(s_j)\prod_{t=1}^{n_j-1} \alpha_j(t)} \right)
\end{aligned}$$

Noting that $\alpha_j(0) = 0$, we can rewrite the last expression as

$$\begin{aligned}
F_j(z) &= \frac{1}{1-\theta_j z/\alpha_j(s_j)} \left(\sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} - \sum_{n_j=0}^{s_j-1} \frac{\alpha_j(n_j)(\theta_j z)^{n_j}}{\alpha_j(s_j)\prod_{t=1}^{n_j} \alpha_j(t)} \right) \\
&= \frac{1}{1-\theta_j z/\alpha_j(s_j)} \sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j} \alpha_j(t)} \left(1 - \frac{\alpha_j(n_j)}{\alpha_j(s_j)}\right) \\
&= \frac{1}{1-\theta_j z/\alpha_j(s_j)} \sum_{n_j=0}^{s_j-1} z^{n_j} \phi_j(n_j)
\end{aligned}$$

Therefore $\mathcal{H}_{\boldsymbol{\theta}}(N) = \mathcal{G}(z) \prod_{j=1}^{M} \mathcal{Q}_j(z)$, where

$$\mathcal{G}(z) = \prod_{j=1}^{M} \frac{1}{1-\theta_j z/\alpha_j(s_j)}$$

is the generating function of the normalizing constant $g_{\boldsymbol{\sigma}}(n)$ in a fixed-rate model with demands $\theta_j/\alpha_j(s_j)$ and

$$\mathcal{Q}_j(z) = \sum_{n_j=0}^{s_j-1} z^{n_j} \phi_j(n_j)$$

The derivation until now matches a known result in [37], following a derivation strategy similar to [34]. However, the next steps obtain the final result that proves the theorem.

Let us observe that $h_{\boldsymbol{\theta}}(N)$ is the coefficient of z^N in $\mathcal{G}(z) \prod_{j=1}^{M} \mathcal{Q}_j(z)$, which is obtained by differentiating N times with respect to z at the origin and dividing by $N!$. This yields, using

Leibnitz' rule for multiple differentiation of a product,

$$h_{\boldsymbol{\theta}}(N) = \frac{1}{N!}\sum_{v=0}^{N} \frac{N!}{(N-v)!}\mathcal{G}^{(N-v)}(0) \sum_{\substack{\mathbf{0}\leq \boldsymbol{v}<\boldsymbol{s}:\\ \sum v_j = v}} \prod_{k=1}^{M} \phi_k(v_k)$$

But $g_{\boldsymbol{\sigma}}(N-v) = \frac{\mathcal{G}^{(N-v)}(0)}{(N-v)!}$ since $\mathcal{G}(z)$ is the generating function of $g_{\boldsymbol{\sigma}}(n)$ and so

$$\begin{aligned} h_{\boldsymbol{\theta}}(N) &= \sum_{v=0}^{N} g_{\boldsymbol{\sigma}}(N-v) \sum_{\substack{\mathbf{0}\leq \boldsymbol{v}<\boldsymbol{s}:\\ \sum v_j = v}} \prod_{k=1}^{M} \phi_j(v_k) \\ &= \sum_{\mathbf{0}\leq \boldsymbol{v}<\boldsymbol{s}} g_{\boldsymbol{\sigma}}(N-v) \prod_{k=1}^{M} \phi_j(v_k), \end{aligned}$$

defining $g_{\boldsymbol{\sigma}}(n) = 0$ for $n < 0$.

A.3. Generalized formula for single-class LLD normalizing constants

Theorem 1 is derived in A.2 and as mentioned gives an algebraic equivalent of a result concerning generating functions obtained in [37, Eq. (8)]. We here further discuss the significance of the result, followed by an example. Note first that the outer summation in (8) may be equivalently rewritten as a sum of vectors $\boldsymbol{v}$ such that $0 \leq |\boldsymbol{v}| \leq V$, where $V = \min(\sum_{i=1}^{M}(s_i - 1), N)$. Significantly, we note that (8) is an explicit formula since a closed-form expression for the normalizing constant $g_{\boldsymbol{\sigma}}(N-v)$ is known from [30], which, incidentally, can be computed in $\mathcal{O}(1)$ time and space as the population N grows, for fixed M. Note that the above formulas are such that delay stations and fixed-rate nodes ($s_i = 1$) can be accounted for into the $g_{\boldsymbol{\sigma}}(N-v)$ terms, there is no requirement for them to be accounted for in $\Phi_{\boldsymbol{\theta}}(\boldsymbol{v})$ and the outer summation over $\boldsymbol{v}$.

Example. We now show that the proposed closed-form solution reduces to (6), i.e., the known result of Gordon [34], in the special case of multi-server stations with distinct scaled demands. Note first that, under the assumption that each $\sigma_i = \theta_i/s_i$ is distinct, it follows that [71]

$$g_{\boldsymbol{\sigma}}(N-v) = \sum_{i=1}^{M} \frac{(\theta_k/s_k)^{N+M-v-1}}{\prod_{k\neq i}(\theta_i/s_i - \theta_k/s_k)}.$$

In multi-server stations $\alpha_k(n_k) = \min(n_l, s_k)$ and because the $\phi_k(v_k)$ terms in Theorem 1 appear only for $v_k \leq s_k$, it is always $\alpha_k(v_k) = v_k$. As such, we obtain the expression $\phi_k(v_k) = \theta_k^{v_k}\left(1 - \frac{v_k}{s_k}\right)/v_k!$, for $v_k \leq s_k$, which gives precisely Gordon's result of (6) when plugged into (8).

A.4. Proof of Theorem 2

As shown in A.2, the product-form factor for a LLD station j admits the generating function

$$F_j(z) = \frac{1}{1-\theta_j z/\alpha_j(s_j)} \sum_{n_j=0}^{s_j-1} \frac{(\theta_j z)^{n_j}}{\prod_{t=1}^{n_j}\alpha_j(t)} \left(1 - \frac{\alpha_j(n_j)}{\alpha_j(s_j)}\right) \tag{A.14}$$

Let $\widetilde{F}_j(z) = F_j(z^{-1})$ be the corresponding z-transform, that is

$$\begin{aligned}
\widetilde{F}_j(z) &= \frac{z}{z-\theta_j/\alpha_j(s_j)} \sum_{n_j=0}^{s_j-1} \frac{(\theta_j/z)^{n_j}}{\prod_{t=1}^{n_j}\alpha_j(t)} \left(1-\frac{\alpha_j(n_j)}{\alpha_j(s_j)}\right) \\
&= \frac{1}{1-z\alpha_j(s_j)/\theta_j} \left(-z\alpha_j(s_j)/\theta_j + \sum_{n_j=1}^{s_j-1} \frac{(\theta_j/z)^{n_j-1}}{\prod_{t=1}^{n_j}\alpha_j(t)} (\alpha_j(n_j)-\alpha_j(s_j))\right) \\
&= \frac{1}{1-z/\rho_j} \left(-z/\rho_j + \sum_{n_j=1}^{s_j-1} \frac{(\theta_j/z)^{n_j-1}}{\prod_{t=1}^{n_j}\alpha_j(t)} (\alpha_j(n_j)-\alpha_j(s_j))\right)
\end{aligned}$$

Following a similar argument as in [36, Thm. 4], we can show that the leading term of the asymptotic expansion

$$h_{\theta}(N) \sim C_B N^{B-1}$$

is obtained by the following application of the final value theorem

$$C_B = \lim_{N\to\infty} \frac{h_{\theta}(N)}{N^B} = \frac{1}{(B-1)!} \lim_{z\to 1} \frac{(z-1)^B}{z^{B-1}} \prod_{j=1}^{M} \widetilde{F}_j(z)$$

Consider first the case where the service demands θ_j are scaled so that $\rho_{\max} = \max_j \rho_j = 1$, with $\rho_j = \theta_k/\alpha_j(s_j)$. Recall that indexes are ordered such that $\rho_1 = ... = \rho_B = \rho_{\max}$. Then

$$C_B = \frac{1}{(B-1)!} \prod_{i=1}^{B} \left(1 + \sum_{n_i=1}^{s_i-1} \frac{\theta_i^{n_i}}{\prod_{t=1}^{n_i}\alpha_i(t)} \left(1-\frac{\alpha_i(n_i)}{\alpha_i(s_i)}\right)\right) \cdot \prod_{j=B+1}^{M} \frac{1}{1-\rho_j} \left(1 + \sum_{n_j=1}^{s_j-1} \frac{\theta_j^{n_j}}{\prod_{t=1}^{n_j}\alpha_j(t)} \left(1-\frac{\alpha_i(n_i)}{\alpha_i(s_i)}\right)\right)$$

which may be rewritten as

$$C_B = \frac{1}{(B-1)!} \prod_{i=1}^{M} \sum_{n_i=0}^{s_i-1} \phi_i(n_i) \cdot \prod_{j=B+1}^{M} \frac{1}{1-\rho_j}$$

The final expression is obtained by pre-multiplying the expression by a scaling factor ρ_{max}^N to handle the general case where $\rho_{\max} \neq 1$.

A.5. Proof of Theorem 3

For $v_k \geq s_k$, we let the terms $\phi_k(v_k) = 0$, so that $\Phi_{\boldsymbol{\theta}}(\boldsymbol{v}) = 0$. Then we may rewrite (8) as

$$h_{\boldsymbol{\theta}}(N) = \sum_{v=0}^{V} g_{\boldsymbol{\sigma}}(N-v) \sum_{\substack{\boldsymbol{v}\geq 0:\\ |\boldsymbol{v}|=v}} \Phi_{\boldsymbol{\theta}}(\boldsymbol{v}) \tag{A.15}$$

under the convention that $g_{\boldsymbol{\sigma}}(n) = 0$ if $n \leq 0$. The last right-hand side term may now be rewritten as

$$e_{\boldsymbol{\theta n}}(v) = \sum_{\substack{\boldsymbol{v}\geq 0:\\ |\boldsymbol{v}|=v}} \Phi_{\boldsymbol{\theta n}}(\boldsymbol{v}) = \sum_{\substack{\boldsymbol{v}\geq 0:\\ |\boldsymbol{v}|=v}} \prod_{i=1}^{M} \frac{\left(\sum_{r=1}^{R}\theta_{i,r}n_r\right)^{v_i}}{\prod_{k=1}^{v_i}\alpha_i(k)} \left(1-\frac{\alpha_i(v_i)}{\alpha_i(s_i)}\right)$$

The structure of this expression is identical to that of a single-class LLD normalizing constant, as far as its dependence on $\boldsymbol{n}$ is concerned. Then, using a similar argument to that in the proof in A.7, we find that

$$E_{\boldsymbol{\theta}}(\boldsymbol{d}) = \frac{\Delta^{\boldsymbol{d}}_{\boldsymbol{k},\boldsymbol{0}} e_{\boldsymbol{\theta k}}(v)}{d_1! \cdots d_r!}$$

Using Lemma 1 and (A.15) we now get

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{N_1! \cdots N_R!} \Delta^{\boldsymbol{N}}_{\boldsymbol{n},\boldsymbol{0}} \sum_{v=0}^{V} g_{\boldsymbol{\sigma n}}(N-v) e_{\boldsymbol{\theta n}}(v)$$

The Leibnitz product rule for derivatives has an equivalent for finite differences, namely that for two polynomials $g(\boldsymbol{n})$ and $f(\boldsymbol{n})$ in R variables, of orders $\boldsymbol{v}$ and $\boldsymbol{N}-\boldsymbol{v}$ respectively ($\boldsymbol{0} \leq \boldsymbol{v} \leq \boldsymbol{N}$), we have

$$\Delta^{\boldsymbol{N}}_{\boldsymbol{n},\boldsymbol{0}} f(\boldsymbol{n}) g(\boldsymbol{n}) = \sum_{\boldsymbol{0} \leq \boldsymbol{d} \leq \boldsymbol{N}} \binom{\boldsymbol{N}}{\boldsymbol{d}} \Delta^{\boldsymbol{N}-\boldsymbol{d}}_{\boldsymbol{n},0} f(\boldsymbol{n}) \Delta^{\boldsymbol{d}}_{\boldsymbol{k},0} g(\boldsymbol{k}) \tag{A.16}$$

where $\binom{\boldsymbol{N}}{\boldsymbol{d}} = \prod_{r=1}^{R} N_r!/(d_r!(N_r-d_r)!)$. This follows since the difference operator is linear and can be applied term-by-term in the polynomial product. Then, using (A.19) and (A.19) the result can be seen.

Since finite difference is a linear operator, this yields

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{N_1! \cdots N_R!} \sum_{v=0}^{V} \sum_{\boldsymbol{0} \leq \boldsymbol{d} \leq \boldsymbol{N}} \binom{\boldsymbol{N}}{\boldsymbol{d}} \Delta^{\boldsymbol{N}-\boldsymbol{d}}_{\boldsymbol{n},0} g_{\boldsymbol{\sigma n}}(N-v) \Delta^{\boldsymbol{d}}_{\boldsymbol{k},0} e_{\boldsymbol{\theta k}}(v) \tag{A.17}$$

Because $e_{\boldsymbol{\theta k}}(v)$ is a polynomial of order v, if $|\boldsymbol{d}| < v$ then $\Delta^{\boldsymbol{d}}_{\boldsymbol{k},0} e_{\boldsymbol{\theta k}}(v) = 0$ by (A.19). Similarly, since $g_{\boldsymbol{\theta n}}(N-v)$ is a polynomial of order $N-v$, if $|\boldsymbol{d}| > v$ then $\Delta^{\boldsymbol{N}-\boldsymbol{d}}_{\boldsymbol{n},0} g_{\boldsymbol{\sigma n}}(N-v) = 0$. Hence, the inner sum can be restricted to $|\boldsymbol{d}| = v$, so that

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{N_1! \cdots N_R!} \sum_{v=0}^{V} \sum_{\substack{\boldsymbol{d} \geq \boldsymbol{0} \\ |\boldsymbol{d}|=v}} \binom{\boldsymbol{N}}{\boldsymbol{d}} \Delta^{\boldsymbol{N}-\boldsymbol{d}}_{\boldsymbol{n},0} g_{\boldsymbol{\sigma n}}(N-v) \Delta^{\boldsymbol{d}}_{\boldsymbol{k},0} e_{\boldsymbol{\theta k}}(v) \tag{A.18}$$

Substituting for $\binom{\boldsymbol{N}}{\boldsymbol{d}}$, the result follows from Theorem 1.

A.6. Finite shifts and differences

We use the following definitions of finite shifts and differences on the kth element $a(k)$ of a sequence $(a(0), a(1), ...)$ [20]:

- Analogously to the differentiation operator $\frac{\mathrm{d}}{\mathrm{d}x}$, the forward shift operator F_n is defined by $F_n a(n) = a(n+1)$ for $n \geq 0$. Higher-order shifts of size $k \geq 1$ are defined by $F_n^k a(n) = a(n+k)$. When applied to a particular element of the sequence rather than the generic one, e.g. the ith, we write $F^k_{n,i} a(n) \equiv F^k_n a(n)\big|_{n=i} = a(i+k)$. In our analysis we will usually take $i = 0$.

- The forward difference operator $\Delta_n = F_n - I$, where I is the identity function, so that $\Delta_n a(n) = a(n+1) - a(n)$ and $\Delta_{n,i} a(n) = a(i+1) - a*$. Since F_n is a linear operator, we have

$$\Delta_n^k a(n) = \left(\sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} F_n^m \right) a(n) = \sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} a(n+m)$$

  and, in particular, $\Delta_{n,i}^k a(n) = \sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} a(i+m)$ for $i \geq 0$.

In the special case that $a(n) = n^p$, $p \in \mathbb{N}$, it is straightforward to prove the following result.

Proposition 1. For non-negative integers k, p,

$$\Delta_{n,0}^k n^p = \sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} m^p = \begin{cases} 0 & \text{if } p < k, \\ k! & \text{if } p = k \end{cases} \tag{A.19}$$

Proof.

$$\Delta_n^k n^p = \left(\sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} F_n^m \right) n^p = \sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} (n+m)^p$$

giving the required expression at $n = 0$. In the case that $p = k = 0$, $\Delta_n^k n^p = 0$ and for $k = 1$, $\Delta_n^k n^0 = 0, \Delta_n^k n^1 = 1$. Now assume inductively that the result holds at $p, k \geq 0$. Then, if $k > p+1$,

$$\begin{aligned} \Delta_n^k n^{p+1} &= \sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} m^{p+1} = \sum_{m=0}^{k} (-1)^{k-m} \binom{k-1}{m-1} k m^p \\ &= k \sum_{m=0}^{k-1} (-1)^{k-1-m} \binom{k-1}{m} (m+1)^p \\ &= k \sum_{m=0}^{k-1} (-1)^{k-1-m} \binom{k-1}{m} \sum_{j=0}^{p} \binom{p}{j} m^j = 0 \end{aligned}$$

by the inductive hypothesis. If $k = p+1$, the last line becomes $k \sum_{m=0}^{k-1} (-1)^{k-1-m} \binom{k-1}{m} m^p = k!$ by the inductive hypothesis since $p = k-1$. □

In fact this result is just $k!$ multiplied by the Stirling number of the second kind, $S(p,k) = \frac{1}{k!} \sum_{m=0}^{k} (-1)^{k-m} \binom{k}{m} m^p$, for which it is known that $S(p,k) = 0$ for $p < k$ and 1 if $p = k$.

This implies that for univariate discrete polynomials

$$\Delta_{n,0}^k \left(c_0 + c_1 n + \ldots + c_n n^k \right) = c_n n! \tag{A.20}$$

so providing an alternative means to multiple differentiation for finding the coefficient of n^k in a polynomial of degree k in the integer variable n.

The previous results are easily generalized to multivariate polynomials $g(\boldsymbol{n})$, where $\boldsymbol{n} = (n_1, \ldots, n_p)$ is a set of integer variables. We introduce the notation $\Delta_{\boldsymbol{n}}^{\boldsymbol{k}} g(\boldsymbol{n}) = \Delta_{n_p}^{k_p} \Delta_{n_{p-1}}^{k_{p-1}} \cdots \Delta_{n_1}^{k_1} g(n_1, \ldots, n_p)$ for the composition of the differences of orders $k_1, \ldots, k_p$ in the discrete variables $n_1, \ldots, n_p$. The following result is now immediate.

Proposition 2. Let $g(\boldsymbol{n})$ be a polynomial of degree $k = k_1 + \cdots + k_m$ in the integer variables $\boldsymbol{n} = (n_1, \ldots, n_m)$. Denote by a_0 the coefficient of $n_1^{k_1} n_2^{k_2} \ldots n_m^{k_m}$ in $g(\boldsymbol{n})$. Then,

$$\Delta^{\boldsymbol{k}}_{\boldsymbol{n},\boldsymbol{0}} g(\boldsymbol{n}) = a_0 \prod_{i=1}^{m} k_i! \tag{A.21}$$

The proof follows easily from (A.20) by induction on m.

A consequence of the last proposition is that for arbitrary coefficients $u_i \in \mathbb{R}$, $1 \leq i \leq M$,

$$\begin{aligned} \frac{1}{n!} \Delta^{\boldsymbol{n}}_{\boldsymbol{k},\boldsymbol{0}} \left(\sum_{i=1}^{m} k_i u_i \right)^{n} &= \frac{1}{n!} \Delta^{\boldsymbol{n}}_{\boldsymbol{k},\boldsymbol{0}} \sum_{\substack{\boldsymbol{v} \geq \boldsymbol{0}: \\ v = n}} \frac{n!}{v_1! \cdots v_m!} \prod_{i=1}^{m} k_i^{v_i} u_i^{v_i} \\ &= u_1^{n_1} u_2^{n_2} \cdots u_m^{n_m} \end{aligned} \tag{A.22}$$

where $\boldsymbol{v} = (v_1, \ldots, v_m)$, $v = \sum_{i=1}^{m} v_i$, and we have used the multinomial theorem and Proposition 2.

A.7. Proof of Lemma 1

Before giving a proof, we need to give a few preliminaries. The following result first obtained in [37] shows that the normalizing constant, $H_{\boldsymbol{\theta}}(\boldsymbol{N})$, of a multiclass LLD model with class-independent scaling functions $\alpha_k(n_k)$ may be written as a weighted sum of normalizing constants of single-class models.

Proposition 3. The generating function of the normalizing constants in an R-class model with M nodes, $\mathcal{H}^R_{\boldsymbol{\theta}}(\boldsymbol{z})$ is the same as the generating function in a single-class model $\mathcal{H}_{\boldsymbol{\theta}}(\boldsymbol{w})$ with demand at node k set to $\sum_{r=1}^{R} z_r \theta_{k,r}/w$ and the same scaling functions α_k.

Proof.

$$\mathcal{H}^R_{\boldsymbol{\theta}}(\boldsymbol{z}) = \sum_{N_1=0}^{\infty} \ldots \sum_{N_R=0}^{\infty} H_{\boldsymbol{\theta}}(\boldsymbol{N}) \prod_{r=1}^{R} z_r^{N_r} = \sum_{\boldsymbol{n} \geq \boldsymbol{0}} \prod_{k=1}^{M} \frac{n_k!}{\prod_{t=1}^{n_k} \alpha_k(t)} \prod_{r=1}^{R} \frac{(z_r \theta_{k,r})^{n_{k,r}}}{n_{k,r}!}$$

by equation (2), where $\boldsymbol{n}$ is a matrix of non-negative integers $(n_{k,r} \mid 1 \leq k \leq M, 1 \leq r \leq R)$ and $n_k = \sum_{r=1}^{R} n_{k,r}$. Hence

$$\begin{aligned} \mathcal{H}^R_{\boldsymbol{\theta}}(\boldsymbol{z}) &= \prod_{k=1}^{M} \sum_{n_k=0}^{\infty} \frac{1}{\prod_{t=1}^{n_k} \alpha_k(t)} \sum_{n_{k,r}=0}^{n_k} \frac{n_k!}{\prod_{r=1}^{R} n_{k,r}!} \prod_{r=1}^{R} (z_r \theta_{k,r})^{n_{k,r}} \\ &= \prod_{k=1}^{M} \sum_{n_k=0}^{\infty} \frac{1}{\prod_{t=1}^{n_k} \alpha_k(t)} \left(\sum_{r=1}^{R} z_r \theta_{k,r} \right)^{n_k} \end{aligned} \tag{A.23}$$

by the multinomial theorem, and the result follows. □

The normalizing constant $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ is, by construction, the coefficient of $z_1^{N_1} \ldots z_M^{N_M}$ in $\mathcal{H}^R(\boldsymbol{z})$ and we can proceed as in Theorem 1 to pursue a closed form expression. We can also obtain $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ by finding the coefficient of $z_1^{N_1} \ldots z_M^{N_M}$ in just the term

$$h_{\boldsymbol{\theta}\boldsymbol{z}}(N) = \sum_{\substack{\boldsymbol{n} \geq \boldsymbol{0}: \\ |\boldsymbol{n}|=N}} \prod_{k=1}^{M} \frac{\left(\sum_{r=1}^{R} z_r \theta_{k,r} \right)^{n_k}}{\prod_{t=1}^{n_k} \alpha_k(t)}, \tag{A.24}$$

where $\boldsymbol{z} = (z_1, \dots, z_R)^T$ and $h_{\boldsymbol{\theta z}}(N)$ is a single-class normalizing constant with demands $\boldsymbol{\theta z}$, which are determined by a product of the multiclass demands matrix $\boldsymbol{\theta}$ with column vector $\boldsymbol{z}$. This follows because $h_{\boldsymbol{\theta z}}(N)$ contains all the terms of total degree N that occur in $\mathcal{H}^R(\boldsymbol{z})$. Perhaps the best known way to find this coefficient is by multiple differentiation at the origin. We show that an alternative approach is to use forward differences defined on a real sequence.

As already noted, $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ is the coefficient of $z_1^{N_1} \dots z_M^{N_M}$ in $\mathcal{H}^R(\boldsymbol{z})$ and also in the finite sum $h_{\boldsymbol{\theta z}}(N)$ (equation A.24). Therefore, by Proposition 2, when $\boldsymbol{z}$ is the integer vector $\boldsymbol{m}$, $H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{\Delta^{\boldsymbol{N}}_{\boldsymbol{m},\boldsymbol{0}} h_{\boldsymbol{\theta m}}(N)}{N_1! \cdots N_R!}$, proving the first part of the theorem.

For the second part, applying the multinomial theorem to the terms $\sum_{r=1}^R \theta_{i,r} m_r$ yields

$$h_{\boldsymbol{\theta m}}(N) = \sum_{\substack{\boldsymbol{k} \geq \boldsymbol{0}: \\ |\boldsymbol{k}| = N}} \sum_{\substack{\boldsymbol{v} \geq \boldsymbol{0}: \\ \sum_{r=1}^R v_{i,r} = k_i}} \prod_{i=1}^M k_i! \prod_{r=1}^R m_r^{V_r} \left(\frac{\theta_{i,r}^{v_{i,r}}}{v_{i,r}! \alpha_i(k_i)} \right)$$

where $\boldsymbol{v} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_M)$, $\boldsymbol{v}_i = (v_{i,1}, \dots, v_{i,R})$, and $V_r = \sum_{i=1}^M v_{i,r}$. Therefore combining the last result with Proposition 2

$$\frac{\Delta^{\boldsymbol{N}}_{\boldsymbol{m},\boldsymbol{0}} h_{\boldsymbol{\theta m}}(N)}{\prod_{r=1}^R N_r} = \sum_{\substack{\boldsymbol{v} \geq \boldsymbol{0}: \\ V_r = N_r}} \prod_{i=1}^M v_i! \prod_{r=1}^R \left(\frac{\theta_{i,r}^{v_{i,r}}}{v_{i,r}! \alpha_i(k_i)} \right)$$

where $v_i = \sum_{r=1}^R v_{i,r}$ and we have noted that the constraints $\sum_{i=1}^M k_i = N$ are implied by $\sum_{r=1}^R v_{i,r} = k_i$ and $V_r = N_r$, $\forall r$. The right hand side of this last formula is simply $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ as given by the product-form expression in equation (2).

Note that a merit of (27) is that one can evaluate the formula exactly (up to numerical precision) without recourse to symbolic manipulation, which is necessary when using derivatives. It is also generally more memory efficient than convolution, requiring constant storage requirements as opposed to $\mathcal{O}(N^R)$. However, in this paper we are mainly interested in the fact that (27) leads to new expressions and integral representations for $H_{\boldsymbol{\theta}}(\boldsymbol{N})$.

A.8. Proof of Theorem 6

According to (27), $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ can be computed using R independent finite difference operators. Since the integrand is a normalizing constant, it is a multivariate polynomial in the demands. Therefore $h_{\boldsymbol{\theta z}}(N)$ is polynomially bounded and holomorphic, which allows us to apply the Norlund-Rice integral

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi i)^R} \oint \frac{h_{\boldsymbol{\theta z}}(N)}{\prod_{r=1}^R z_r(z_r-1)(z_r-2)\cdots(z_r-N_r)} d\boldsymbol{z} \tag{A.25}$$

where $\boldsymbol{z} = (z_1, \dots, z_R)$. This expression can be evaluated numerically by ensuring that integration contour for each variable z_r encircles the poles $0, \dots, N_r$.

In particular, let us consider R circular contours centered at the origin with common radius $X > \max N_r$, which encircles all the poles of the Norlund-Rice integrand. On the contour,

$z_r = Xe^{it_r}$ and $dz = iXe^{it_r}$ for $0 \leq \theta \leq 2\pi$ and $1 \leq r \leq R$. With these changes of variable, as $X \to +\infty$, the integrand becomes

$$h_{e(\boldsymbol{t})}(N) \prod_{r=1}^{R} \frac{ie^{it_r}}{e^{i(N_r+1)t_r}} = h_{e(\boldsymbol{t})}(N) \prod_{r=1}^{R} \frac{i}{e^{iN_r t_r}}$$

since, referring to equation (A.24), $h_{\boldsymbol{\theta z}}(N)$ is a multinomial in $\{z_r | 1 \leq r \leq R\}$ of uniform degree N. Using the transformed integrand, we then have

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi)^R} \int_0^{2\pi} \cdots \int_0^{2\pi} h_{\boldsymbol{\Theta}(\boldsymbol{t})}(N) e^{-iN_1 t_1} \dots e^{-iN_R t_R} d\boldsymbol{t} \tag{A.26}$$

where $\boldsymbol{t} = (t_1, \dots, t_R)$.

If we define $\boldsymbol{\beta} = N^{-1}\boldsymbol{N}$ to be the class-mix vector and exploit the scaling property of the normalizing constant [35], i.e., $a^N h_z(N) = h_{a \cdot z}(N)$, for arbitrary scalar a, the Norlund-Rice integral can be rewritten using (A.26) as

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{(2\pi)^R} \int_0^{2\pi} \cdots \int_0^{2\pi} h_{\boldsymbol{\Theta}(\boldsymbol{t} - \boldsymbol{\beta}^T \boldsymbol{t})}(N) d\boldsymbol{t} \tag{A.27}$$

where $\boldsymbol{t} = (t_1, \dots, t_R)$.

A.9. Proof of Theorem 7

We begin by first rewriting (8) as

$$h_{\boldsymbol{\theta}}(N) = \sum_{\boldsymbol{0} \leq \boldsymbol{v} < \boldsymbol{s}} g_{\boldsymbol{\sigma}}(N - v) \prod_{i=1}^{M} \frac{\sigma_i^{v_i} (\alpha_i(v_i))^{v_i}}{\prod_{t=1}^{v_i} \alpha_i(t)} \left(1 - \frac{\alpha_i(v_i)}{\alpha_i(s_i)}\right) \tag{A.28}$$

or, more compactly, as

$$h_{\boldsymbol{\theta}}(N) = \sum_{\boldsymbol{0} \leq \boldsymbol{v} < \boldsymbol{s}} \alpha_{\boldsymbol{v}} g_{\boldsymbol{\sigma}}(N - v) \prod_{i=1}^{M} \sigma_i^{v_i} \tag{A.29}$$

Using in a sequence (32) and the following result [72, 73]

$$t_1^{n_1} t_2^{n_2} \cdots t_m^{n_m} = \frac{1}{n!} \sum_{\boldsymbol{0} \leq \boldsymbol{k} \leq \boldsymbol{n}} (-1)^{n-k} \prod_{i=1}^{m} \binom{n_i}{k_i} \left(\sum_{i=1}^{m} k_i t_i\right)^n \tag{A.30}$$

we may express the single-class LLD normalizing constant as

$$\begin{aligned} h_{\boldsymbol{\sigma}}(N) &= \sum_{\boldsymbol{0} \leq \boldsymbol{v} \leq \boldsymbol{s}} \alpha_{\boldsymbol{v}} \int_{T_M} \left(\sum_{k=1}^{M} \sigma_k u_k\right)^{N-v} \prod_{i=1}^{M} \sigma_i^{v_i} d\boldsymbol{u} \\ &= \sum_{\boldsymbol{0} \leq \boldsymbol{v} \leq \boldsymbol{s}} \frac{\alpha_{\boldsymbol{v}}}{N!} \int_{T_M} \Delta_{t_0}^{N-v} \Delta_{\boldsymbol{t}}^{\boldsymbol{v}} \left(t_0 \sum_{k=1}^{M} \sigma_k u_k + \sum_{i=1}^{M} t_i \sigma_i\right)^N d\boldsymbol{u} \end{aligned}$$

Combining the last result with (27), we get

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{\Delta_{\boldsymbol{n}}^{\boldsymbol{N}} h_{\boldsymbol{\theta n}}(N)}{N_1! \cdots N_R!} = \frac{1}{\prod_{i=1}^{R} N_r!} \int_{T_K} \sum_{\boldsymbol{0} \leq v < s} \frac{a_{\boldsymbol{v}}}{N!} \cdot \boldsymbol{\Delta}_{t_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{t}}^{\boldsymbol{v}} \boldsymbol{\Delta}_{\boldsymbol{n}}^{\boldsymbol{N}} \left(\sum_{k=1}^{M} \sum_{r=1}^{R} \sigma_{k,r} n_r (t_0 u_k + t_k)\right)^N d\boldsymbol{u}$$

Using (A.22) we finally obtain

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{\prod_{i=1}^{R} N_r!} \int_{T_K} \sum_{\boldsymbol{0} \leq v < s} a_{\boldsymbol{v}} \cdot \boldsymbol{\Delta}_{t_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{t}}^{\boldsymbol{v}} \prod_{r=1}^{R} \left(\sum_{k=1}^{M} \sigma_{k,r} n_r (t_0 u_k + t_k)\right)^{N_r} d\boldsymbol{u}$$

A.10. Gradient and Hessian for the Norlund-Rice integral form

A.10.1. **Gradient formula.**

Because the integrand in (35) is a complex-valued function of the real vector $\boldsymbol{x}$, we can differentiate it using the derivatives of its real and imaginary parts with respect to the variables in $\boldsymbol{x}$. We wish to study the stationary points of the integrand, thus by the product-rule we have the complex-valued gradient

$$\frac{\partial h_{\boldsymbol{x}}(N)J(\boldsymbol{x})}{\partial x_r} = h_{\boldsymbol{x}}(N)\frac{\partial J(\boldsymbol{x})}{\partial x_r} + J(\boldsymbol{x})\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_r} \tag{A.31}$$

for $r = 1, \dots, R$.

Before, we need to give an auxiliary result. $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ can be shown to be a Fourier coefficient by virtue of (A.26), given in the proof of Theorem 6. This implies the following inversion formula.

Proposition 4.

$$h_{\boldsymbol{\Theta}(\boldsymbol{t})}(N) = \sum_{\substack{\boldsymbol{n}\geq 0:\\ |\boldsymbol{n}|=N}} H_{\boldsymbol{\theta}}(\boldsymbol{n})e^{i\sum_{r=1}^{R} n_r t_r} \tag{A.32}$$

Proof. $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ is a Fourier coefficient and we know that it is the coefficient of $e^{iN_1t_1} \dots e^{iN_Rt_R}$ in the R-dimensional Fourier series

$$\mathcal{H}^R_{\boldsymbol{\theta}}(\boldsymbol{t}) = \sum_{N_1=0}^{\infty} \dots \sum_{N_R=0}^{\infty} H_{\boldsymbol{\theta}}(\boldsymbol{N}) \prod_{r=1}^{R} e^{iN_rt_r}$$

from equation (A.23). Restricting the series to terms with $N = \sum_{r=1}^{R} N_r$ yields

$$h_{\boldsymbol{\Theta}(\boldsymbol{t})}(N) = \sum_{\substack{\boldsymbol{n}\geq 0:\\ |n|=N}} H_{\boldsymbol{\theta}}(\boldsymbol{n}) \prod_{r=1}^{R} e^{in_rt_r}$$

as required. $\square$

In particular, the last expression implies, by Euler's formula for the complex exponential, that the integrand in (31) is

$$\Re h_{\boldsymbol{\Theta}(\boldsymbol{t}-\boldsymbol{\beta}^T\boldsymbol{t})}(N) = \sum_{\substack{\boldsymbol{n}\geq 0:\\ |\boldsymbol{n}|=N}} H_{\boldsymbol{\theta}}(\boldsymbol{n}) \cos\left(\sum_{r=1}^{R} n_r \left(t_r - \sum_{s=1}^{R} \beta_s t_r\right)\right) \tag{A.33}$$

where $\beta_s = N_s/N$.

We can now give an analytical expression for the derivatives of $h_{\boldsymbol{x}}(N)$ after introducing some definitions. Let $h^{(k)}_{\boldsymbol{x}}(N-1)$ be the normalizing constant of a queueing network derived from the model of $h_{\boldsymbol{x}}(N)$ by moving one job into a dedicated class C. In this modified model, the new class c consists of a single job, called a *self-looping job*, whose routing probabilities are modified so to make it perpetually loop at station k placing there a demand $\theta_{k,c} = 1$. By this definition, it is therefore $\theta_{j,c} = 0$, $\forall j \neq k$. The modified model is therefore no-longer single-class: it is a multiclass LLD model with modified demands and that retains the same scaling factors $\alpha_k(\cdot)$ as the original model.

Theorem 8.

$$\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_r} = i\frac{\partial t_r(\boldsymbol{x})}{\partial x_r}\left(\sum_{k=1}^{M}\theta_{k,r}e^{if_r(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) - \beta_r N h_{\boldsymbol{x}}(N)\right) \tag{A.34}$$

Proof. From Proposition 4, using a similar argument as the one to derive (A.33), we find after applying the change of variable $\boldsymbol{t}(\boldsymbol{x})$ to the integrand that

$$\Re h_{\boldsymbol{x}}(N) = \sum_{\substack{\boldsymbol{n}\geq 0:\\ |n|=N}} H_{\boldsymbol{\theta}}(\boldsymbol{n})\cos\left(\boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}\right) \tag{A.35}$$

$$\Im h_{\boldsymbol{x}}(N) = \sum_{\substack{\boldsymbol{n}\geq 0:\\ |n|=N}} H_{\boldsymbol{\theta}}(\boldsymbol{n})\sin\left(\boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}\right) \tag{A.36}$$

where $\Re$ and $\Im$ respectively denote real and imaginary part and $f(\boldsymbol{x})$ is defined in (34). In this derivation, it is useful to recall that $h_{\boldsymbol{x}}(N) = h_{\boldsymbol{\Theta}(\boldsymbol{f}(\boldsymbol{x}))}(N)$.

Let us now define $\delta_{rs} = 1$ if and only if $r = s$, and $\delta_{rs} = 0$ otherwise. We first show that

$$\frac{\partial}{\partial x_r}\Re h_{\boldsymbol{x}}(N) = -\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\left(\cos(f_s(\boldsymbol{x}))\sum_{k=1}^{M}\theta_{k,s}\Im h_{\boldsymbol{x}}^{(k)}(N-1)+\sin(f_s(\boldsymbol{x}))\sum_{k=1}^{M}\theta_{k,s}\Re h_{\boldsymbol{x}}^{(k)}(N-1)\right) \tag{A.37}$$

$$\frac{\partial}{\partial x_r}\Im h_{\boldsymbol{x}}(N) = -\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\left(\sin(f_s(\boldsymbol{x}))\sum_{k=1}^{M}\theta_{j,k}\Im h_{\boldsymbol{x}}^{(k)}(N-1)-\cos(f_s(\boldsymbol{x}))\sum_{k=1}^{M}\theta_{k,s}\Re h_{\boldsymbol{x}}^{(k)}(N-1)\right) \tag{A.38}$$

We only give the proof for $\frac{\partial}{\partial x_r}\Re h_{\boldsymbol{x}}(N)$, since the proof for $\frac{\partial}{\partial x_r}\Im h_{\boldsymbol{x}}(N)$ follows similar passages. We begin by noting that

$$\frac{\partial}{\partial x_r}\Re h_{\boldsymbol{x}}(N) = -\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\sum_{\substack{\boldsymbol{n}=(n_1,\ldots,n_R):\\ n_1+\ldots+n_R=N}} n_s H_{\boldsymbol{\theta}}(\boldsymbol{n})\sin\left(\boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}\right)$$

We now plug-in the RECAL recurrence relation for load-dependent models [59, 74]

$$n_s H_{\boldsymbol{\theta}}(\boldsymbol{n}) = \sum_{k=1}^{M}\theta_{k,s}H_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{n}-1_s) \tag{A.39}$$

where $H_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{n}-1_s)$ is the multiclass normalizing constant of model with $\boldsymbol{n}-1_s$ and a self-looping job with unit demand perpetually looping at station k. In the case $n_s = 1$, this term needs to be evaluated using the termination conditions obtained in [74, Eq. (2.47)].

We get

$$\begin{aligned}\frac{\partial\Re h_{\boldsymbol{x}}(N)}{\partial x_r} &= -\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,r}\sum_{\boldsymbol{n}:n_s>0}H_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{n}-1_s)\sin\left(\boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}\right)\\ &= -\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,r}\sum_{\boldsymbol{n}:\sum_r n_r=N-1}H_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{n})\sin\left(\sum_{r=1}^{R}f_r(\boldsymbol{x})(n_r+\delta_{rs})\right)\end{aligned}$$

where $\delta_{rs} = 1$ if and only if $r = s$, and $\delta_{rs} = 0$ otherwise. The last passage is standard and used in the derivation of known formulas such as the convolution algorithm [13].

Using standard relations for trigonometric functions, we may write

$$\sin\left(\sum_{r=1}^{R} f_r(\boldsymbol{x})(n_r + \delta_{rs})\right) = \sin\left(\boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}\right)\cos\left(f_s(\boldsymbol{x})\right) + \cos\left(\boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}\right)\sin\left(f_s(\boldsymbol{x})\right)$$

which concludes the proof of (A.37). As mentioned, the proof for (A.38) is similar and omitted.

Using expressions (A.37) and (A.37), it is now possible to obtain the partial derivative of the Norlund-Rice integrand. Note that by Euler's formula for the complex exponential

$$\begin{aligned}
\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_r} =& \frac{\partial \Re h_{\boldsymbol{x}}(N)}{\partial x_r} + i\frac{\partial \Im h_{\boldsymbol{x}}(N)}{\partial x_r} \\
=& -\sum_{s=1}^{R} \frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\left(e^{if_s(\boldsymbol{x})}\sum_{k=1}^{M}\theta_{k,s}\Im h_{\boldsymbol{x}}^{(k)}(N-1) - ie^{if_s(\boldsymbol{x})}\sum_{k=1}^{M}\theta_{k,s}\Re h_{\boldsymbol{x}}^{(k)}(N-1)\right) \\
=& i\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,s}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1)
\end{aligned}$$

Plugging in

$$\frac{\partial f_s(\boldsymbol{x})}{\partial x_r} = \frac{\partial t_s(\boldsymbol{x})}{\partial x_r} - \sum_{c=1}^{R}\beta_c\frac{\partial t_c(\boldsymbol{x})}{\partial x_r}$$

we get

$$\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_r} = i\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,s}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) \tag{A.40}$$

Since we assumed that the change of variables satifies $t_r(\boldsymbol{x}) = t_r(x_r)$, we now have

$$\begin{aligned}
\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_r} &= i\sum_{s=1}^{R}\frac{\partial f_s(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,s}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) \\
&= i\sum_{s=1}^{R}\frac{\partial t_s(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,s}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) - i\sum_{s=1}^{R}\sum_{c=1}^{R}\beta_c\frac{\partial t_c(\boldsymbol{x})}{\partial x_r}\sum_{k=1}^{M}\theta_{k,s}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) \\
&= i\frac{\partial t_r(x_r)}{\partial x_r}\sum_{k=1}^{M}\theta_{k,r}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) - i\beta_r\frac{\partial t_r(x_r)}{\partial x_r}\sum_{s=1}^{R}\sum_{k=1}^{M}\theta_{k,s}e^{if_s(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1)
\end{aligned}$$

The result now follows with a little algebra applying to the last term the RECAL recurrence relation (A.39) for the normalizing constant $h_{\boldsymbol{x}}(N)$, namely

$$Nh_{\boldsymbol{x}}(N) = \sum_{k=1}^{M}\left(\sum_{s=1}^{R}\theta_{k,s}e^{if_s(\boldsymbol{x})}\right)h_{\boldsymbol{x}}^{(k)}(N-1) \tag{A.41}$$

where the terms within brackets are the elements of the demand vector $\Theta(\boldsymbol{f}(\boldsymbol{x}))$. $\square$

The final expression for the first-order partial derivative of the Norlund-Rice integrand is then computed by (A.31), where the expression of the Jacobian depend on the selected change of variables, e.g., the Jacobians of the probit or logit transformations. We also remark that the above derivation also obtains a more general expression (A.40) for the case where the components of $\boldsymbol{t}(\boldsymbol{x})$ are multivariable functions $t_r(\boldsymbol{x})$.

A.10.2. **Hessian formula.**

It is now possible to derive the expressions for the entries of the Hessian matrix

$$\frac{\partial^2 h_{\boldsymbol{x}}(N)J(\boldsymbol{x})}{\partial x_r \partial x_s} = \frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_s}\frac{\partial J(\boldsymbol{x})}{\partial x_r} + \frac{\partial J(\boldsymbol{x})}{\partial x_s}\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_r} + h_{\boldsymbol{x}}(N)\frac{\partial^2 J(\boldsymbol{x})}{\partial x_r \partial x_s} + J(\boldsymbol{x})\frac{\partial^2 h_{\boldsymbol{x}}(N)}{\partial x_r \partial x_s} \tag{A.42}$$

where

$$\frac{\partial^2 h_{\boldsymbol{x}}(N)}{\partial x_r \partial x_s} = i\frac{\partial^2 t_r(\boldsymbol{x})}{\partial x_r \partial x_s}\left(\sum_{k=1}^{M}\theta_{k,r}e^{if_r(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) - N\beta_r h_{\boldsymbol{x}}(N)\right) + i\frac{\partial t_r(\boldsymbol{x})}{\partial x_r}\Bigg($$
$$i\delta_{rs}\frac{\partial f_r(\boldsymbol{x})}{\partial x_s}\sum_{k=1}^{M}\theta_{k,r}e^{if_r(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k)}(N-1) + \sum_{k=1}^{M}\theta_{k,r}e^{if_r(\boldsymbol{x})}\frac{\partial h_{\boldsymbol{x}}^{(k)}(N-1)}{\partial x_s} - N\beta_r\frac{\partial h_{\boldsymbol{x}}(N)}{\partial x_s}\Bigg) \tag{A.43}$$

Even though $h_{\boldsymbol{x}}^{(k)}(N-1)$ consists of two classes, one being the self-looping customer class, it is possible to show that (A.34) remains valid for this normalizing constant since the additional class is independent of $\boldsymbol{x}$. That is

$$\frac{\partial h_{\boldsymbol{x}}^{(k)}(N-1)}{\partial x_r} = i\frac{\partial t_r(\boldsymbol{x})}{\partial x_r}\left(\sum_{j=1}^{M}\theta_{j,r}e^{if_r(\boldsymbol{x})}h_{\boldsymbol{x}}^{(k,j)}(N-2) - (N-1)\beta_r h_{\boldsymbol{x}}^{(k)}(N-1)\right) \tag{A.44}$$

where $h_{\boldsymbol{x}}^{(k,j)}(N-2)$ adds a second self-looping class at node j.

Complexity analysis. The Hessian matrix can thus be computed from the definitions, (A.34) and (A.44) by computing the single-class normalizing constants $h_{\boldsymbol{x}}(N)$, $h_{\boldsymbol{x}}^{k}(N-1)$, and $h_{\boldsymbol{x}}^{(k,j)}(N-2)$ which are in total $\mathcal{O}(R^2M^2)$. On models where the RECAL algorithm is applicable, they can all be obtained in a single invocation of the method to obtain $h_{\boldsymbol{x}}(N)$.

A.11. Gradient and Hessian for the simplex integral form

To evaluate the integral form (33), an additive logistic transform may be applied to induce a change of variables

$$t_i(\boldsymbol{x}) = \begin{cases} e^{x_i}\left(1+\sum_{k=1}^{i}e^{x_k}\right)^{-1} & i = 1 \dots M-1 \\ \left(1+\sum_{k=1}^{i}e^{x_k}\right)^{-1} & i = M \end{cases}$$

This yields the integral in the following form

$$H_{\boldsymbol{\theta}}(\boldsymbol{N}) = \frac{1}{\prod_{r=1}^{R}N_r!}\int_{\mathbb{R}^M} e^{-k(\boldsymbol{x})}d\boldsymbol{x} \tag{A.45}$$

where we have integrated the contribution of the Jacobian within $k(\boldsymbol{x})$. Then, the logarithm of the integrand is given by

$$-k(\boldsymbol{x}) = \log\left(\sum_{\boldsymbol{0}\le\boldsymbol{v}<\boldsymbol{s}}\boldsymbol{\alpha}_{\boldsymbol{v}}\boldsymbol{\Delta}_{l_0}^{N-v}\boldsymbol{\Delta}_{\boldsymbol{l}}^{\boldsymbol{v}}e^{-d_{\boldsymbol{\theta}}(\boldsymbol{x},l_0,\boldsymbol{l})}\right) \tag{A.46}$$

and

$$-d_{\boldsymbol{\theta}}(\boldsymbol{x},l_0,\boldsymbol{l}) = \sum_{r=1}^{R}N_r\log\left(\sum_{k=1}^{M-1}\sigma_{k,r}(l_k+l_0e^{x_k})\right) + \sum_{k=1}^{M-1}x_k - (N+M)\log\left(1+\sum_{k=1}^{M-1}e^{x_k}\right)$$

A.11.1. **Gradient formula.**

It is now straightforward to derive an expression for the gradient of the log of the integrand

$$\frac{\partial(k(\boldsymbol{x}))}{\partial x_i} = -e^{k(\boldsymbol{x})} \sum_{\mathbf{0} \le \boldsymbol{v} < \boldsymbol{s}} \left(\boldsymbol{\alpha_v} \boldsymbol{\Delta}_{l_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{l}}^{\boldsymbol{v}} \left[-\frac{\partial}{\partial x_i} \Big(d_{\boldsymbol{\theta}}(\boldsymbol{x}, l_0, \boldsymbol{l}) \Big) e^{-d_{\boldsymbol{\theta}}(\boldsymbol{x}, l_0, \boldsymbol{l})} \right] \right) \tag{A.47}$$

Where the derivative of $d_{\boldsymbol{\theta}}(\boldsymbol{x}, l_0, \boldsymbol{l})$ is given by

$$-\frac{\partial}{\partial x_i} \Big(d_{\boldsymbol{\theta}}(\boldsymbol{x}, l_0, \boldsymbol{l}) \Big) = u_i(\boldsymbol{x}) \left(l_0 \sum_{r=1}^{R} \left[Y_r(\boldsymbol{u}(\boldsymbol{x}), l_0, \boldsymbol{l}) \Big(\sigma_{ir} - \sum_{k=1}^{M} \sigma_{kr} u_k(\boldsymbol{x}) \Big) \right] - M \right) + 1$$

and

$$Y_r(\boldsymbol{u}(\boldsymbol{x}), l_0, \boldsymbol{l}) = N_r \left(\sum_{k=1}^{M} \sigma_{kr} (l_k + l_0 u_k(\boldsymbol{x})) \right)^{-1}$$

A.11.2. **Hessian formula.**

The Hessian at the stationary point is then given by the following equation

$$\begin{aligned} -\frac{\partial^2(k(\hat{\boldsymbol{x}}))}{\partial x_i \partial x_j} = e^{k(\hat{\boldsymbol{x}})} \sum_{\mathbf{0} \le \boldsymbol{v} < \boldsymbol{s}} \boldsymbol{\alpha_v} \boldsymbol{\Delta}_{l_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{l}}^{\boldsymbol{v}} \Bigg[-\frac{\partial^2}{\partial x_i \partial x_j} \Big(d_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}, l_0, \boldsymbol{l}) \Big) \\ + \Big(-\frac{\partial}{\partial x_i} d_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}, l_0, \boldsymbol{l}) \Big) \Big(-\frac{\partial}{\partial x_j} d_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}, l_0, \boldsymbol{l}) \Big) \Bigg] e^{-d_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}, l_0, \boldsymbol{l})} \end{aligned} \tag{A.48}$$

and the Hessian of $d_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}, l_0, \boldsymbol{l})$ can be written as

$$\begin{aligned} -\frac{\partial^2}{\partial x_i \partial x_j} \Big(d_{\boldsymbol{\theta}}(\boldsymbol{x}, l_0, \boldsymbol{l}) \Big) = \Big(-l_0^2 \sum_{r=1}^{R} \sigma_{ir} \sigma_{jr} \frac{Y_r(\boldsymbol{u}(\boldsymbol{x}), l_0, \boldsymbol{l})^2}{N_r} - \frac{\delta_{ij}}{(u_j(\boldsymbol{x}))^2} \Big) \frac{\partial u_p(\boldsymbol{x})}{\partial x_j} \frac{\partial u_q(\boldsymbol{x})}{\partial x_i} \\ - \frac{\partial}{\partial u_p(\boldsymbol{x})} \Big(d_{\boldsymbol{\theta}}(\boldsymbol{x}, l_0, \boldsymbol{l}) \Big) \frac{\partial^2 u_p(\boldsymbol{x})}{\partial x_i \partial x_j} \end{aligned} \tag{A.49}$$

where the Hessian of $u_p(\boldsymbol{x})$ is as follows

$$\frac{\partial u_p}{\partial x_i} = \delta_{pj} - u_p u_j$$

$$\frac{\partial^2 u_p}{\partial x_i \partial x_j} = \delta_{pj} \delta_{pi} u_p - \delta_{pj} u_p u_i - \delta_{ij} u_p u_j - \delta_{pi} u_p u_j - 2(u_p u_i u_j)$$

where δ_{ij} is the standard Kronecker delta, and we use the shorthand notation $u_p = u_p(\boldsymbol{x})$.

A.12. Implementation remarks

Detailed implementation remarks for the methods listed in tables 3 and 4 are as follows.

- SIM is a discrete event simulation of the queueing network model, based on an explicit simulation of the queueing nodes. Simulation is stopped after collecting 10^6 samples, which returns very accurate estimates on the considered models.

- EX provides an exact analysis solution by the MVA-LD method [14]. It should be noted that, using the definition of LLD throughput it is $H_{\boldsymbol{\sigma}}(\boldsymbol{N}-1_r) = H_{\boldsymbol{\sigma}}(\boldsymbol{N})X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$. With this relation, MVA-LD can also be easily adapted to determine the exact value of the normalizing constant in a numerically stable fashion, from the values of the computed throughputs.

- RD is implemented using AQL [19] to solve the fixed-rate model and convolution for the correction factor.

- AMVA is based on the algorithm in [29]. In this method, the scaling function $\alpha_k(n_k) = \min(n_k, s_k)$ of multi-server stations needs to be approximated with a continuous interpolation. As suggested in the original paper, we set this to be the soft-min function $\alpha_k(n_k) \approx a(n_k, s_k) = (n_k e^{-\kappa \cdot n_k} + s_k e^{-\kappa \cdot s_k}) \cdot (e^{-\kappa \cdot n_k} + e^{-\kappa \cdot s_k})^{-1}$, for large $\kappa > 0$. We here set $\kappa = 20$.

- ODE implements Kurtz's mean-field approximation [48, 56] and becomes increasingly accurate in the limit of a model with large N and M. The underpinning ordinary differential equations (ODEs) are solved using MATLAB's `ode45` solver and available in the open source LINE solver [49].

- NRL is the closed-form approximation (36) of the Norlund-Rice integral form, for the integrand obtained after applying a logit transformation.

- NRP is defined similarly to NRL, but we use instead a probit transformation.

- RDH is the RD heuristic in the form shown in (22), which directly applies to the normalizing constant. The normalizing constant $G_{\boldsymbol{\sigma}}$ is estimated using Monte Carlo sampling for fixed-rate models using 10^7 samples [44], which is efficient and takes 1-2 seconds at most.

- RDA is a baseline approximation of the recurrence relation $G_{\boldsymbol{\sigma}}(\boldsymbol{N}-1_r) = G_{\boldsymbol{\sigma}}(\boldsymbol{N})X_r^{\boldsymbol{\sigma}}(\boldsymbol{N})$ based on the fact that for a large enough population we expect $X_r(\boldsymbol{N}-1_r) \approx X_r(\boldsymbol{N})$ so that we may approximately write $H_{\boldsymbol{\theta}}(\boldsymbol{N}) \approx \prod_{r=1}^{R} X_r(\boldsymbol{N})^{N_r}$. This baseline is used to illustrate a basic method that one may use to approximate $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ without the techniques we have developed.

All the above methods are implemented in MATLAB and tested on commodity hardware.

A.13. Additional validation: normalizing constant approximations

Large-scale models can only be evaluated in simulation, since exact methods are not applicable to those instances. Thus we cannot obtain, in the group of large models, estimates for the normalizing constant $H_{\boldsymbol{\theta}}(\boldsymbol{N})$. Since there is no way to estimate the ground truth on those instances, we focus in this section on the $96,000$ smaller, tractable, models with up to 4 classes and stations.

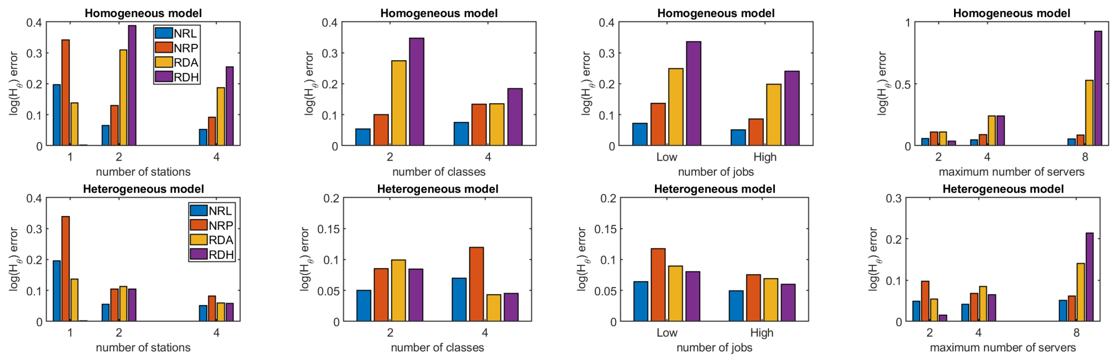


Figure A.4: Percentage error of normalizing constant analyses on tractable, exactly solvable, models.

Since integral forms are estimated more accurately when demands are scaled in $[0, 1]$, as this generally renders the integrand more symmetric, we apply without loss of generality the methods on the related system with demands

$$\theta'_{i,r} = \frac{\theta_{i,r}}{\max\limits_{k=0,\dots,M} \theta_{k,r}}$$

for $i = 0, \dots, M$. After running the method, we re-scale exactly the result to be a normalizing constant for the original model using the exact technique proposed in [35]. For the sake of the validation, we represent the think times using a LLD multi-server station with N homogeneous servers.

We can see in Figure A.4 that the NRL method, based on the logistic transformation, consistently outperforms the other methods. Overall, the average error is 6.46%, as opposed to 11.72% for NRP, 0.49% for RHA, and 26.62% for RHD. As an exception, in the heterogeneous case, as the number of classes R grows the Norlund-Rice integral is less accurate than RHD, which may therefore be preferred for large R. Conversely, the NRP method, based on the probit transformation and the RHA are dominated in accuracy by NRL and RHD.

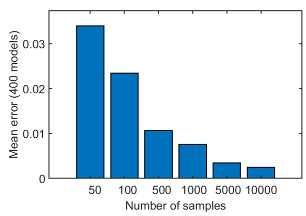
Lower performance for the RHD method is evident in models with a large $s^{\max}$, which is consistent with the fact that the RD heuristic in this case is less accurate, moving from $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ to $e_{\boldsymbol{\theta}}(v)$ generally incurs more error in this scenario, where the state space of $E_{\boldsymbol{\theta}}(\boldsymbol{d})$ is larger. In such models, one may either use NRL. For a large number of classes one can use directly Monte Carlo sampling for the simplex form (33), which as shown earlier in Figure A.5 rapidly converges to an accurate solution after relatively few samples.

A.13.1. **Further normalizing constant analysis examples**

The Monte Carlo integration method of the simplex integral form (33) is not included in the study reported in the previous section. This is because this method is a computationally-demanding iterative approximation. Thus, we illustrate it here separately.

Table A.9: Norlund–Rice evaluation example

Integration Step size	Integral Value	Error	Time $[s]$
$2\pi/16$	26178327.051	7.5%	0.08
$2\pi/32$	25184536.348	3.4%	0.30
$2\pi/64$	24745885.054	1.6%	1.16
$2\pi/128$	24541120.421	0.8%	4.93
$2\pi/256$	24442378.359	0.4%	19.13
$2\pi/512$	24393917.391	0.2%	75.40
$H_{\boldsymbol{\theta}}(\boldsymbol{N})$:	24346063.132	Exact	



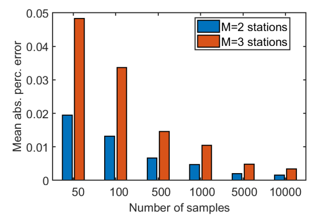


Figure A.5: Monte Carlo sampling of the simplex integral form (33): average errors in estimating $H_{\boldsymbol{\theta}}(\boldsymbol{N})$

We generate random instances with the following characteristics: number of classes $R \in [1, 2]$; number of stations $M \in [2, 3]$; class populations $N_r \in [1, 5]$; numbers of servers $s_i \in [1, 2]$; random service demands $\theta_{kr} \in [0, 1]$. For the Monte Carlo integration method described in Section 5.2, the maximum of the integrand is first obtained by running a conjugate gradient search, implemented using multi-precision arithmetic to avoid numerical issues associated with finite differences and normalizing constants. The average errors in estimating $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ are shown in Figure A.5, aggregated and distinguished by number of stations, indicating a rapid decrease in the magnitude of the errors as the number of samples grows.

Lastly, we showcase a further validation example of the convergence properties of the Norlund-Rice integral form. In this example, we consider two models with $M = 2$ multi-server stations and $R = 2$ classes. The demands are set to $\theta_{k,r} = k \cdot r$. We consider an unbalanced scenario ($s_1 = 1, s_2 = 8$). The integral (31) is evaluated numerically using a step size $\delta = \pi/k$ in which we increase the number of integration points k. Figure A.9 illustrates the accuracy in approximating the normalizing constant as the number of integration points increases up to $k = 512$. The exact values of $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ are obtained using the LD convolution algorithm of [13]. The table refers to the case $R = 2$ and show that with as little as 32 points the Norlund-Rice integral obtains an approximation for $H_{\boldsymbol{\theta}}(\boldsymbol{N})$ having less than 5% error, with increasing accuracy for large samples.