Novel Solutions for Closed Queueing Networks with Load-Dependent Stations*

Giuliano Casale, Peter G. Harrison, Ong Wai Hong
Department of Computing
Imperial College London, UK

1. INTRODUCTION

Load-dependent closed queueing networks are difficult to approximate since their analysis requires to consider state-dependent service demands. Commonly employed evaluation techniques, such as mean-value analysis, are not equally efficient in the load-dependent setting, where mean queue-lengths are insufficient alone to recursively determine the model equilibrium performance.

In this paper, we contribute to addressing this problem by obtaining novel solutions for the normalizing constant of state probabilities in the load-dependent setting. For single-class load-dependent models, we provide the first explicit exact formula for the normalizing constant that applies to models with arbitrary load-dependent rates, while retaining $\mathcal{O}(1)$ complexity with respect to the total population size. From this result, we derive two novel integral forms for the normalizing constant in multiclass load-dependent models, which involve integration in the real and complex domains. The paper also illustrates through experiments the computational gains and accuracy of the obtained expressions.

2. REFERENCE MODEL

We focus on closed multiclass queueing network models that admit a product-form solution [2], such as closed networks including $-/GI/1$ processor sharing queues, $-/M/k$ first-come first-served queues with identical service rates, and $-/GI/\infty$ delay nodes. Service time distributions are assumed to have a rational Laplace-Stieltjes transform.

The queueing network model under study is assumed to have M single-server queues and R job classes. Indexes k, i are used to denote queues ($k, i = 1, \ldots, M$), while indexes r, s are used to indicate classes ($r, s = 1, \ldots, R$). Each class is populated by N_r jobs. Thus, the total number of jobs in the network is $N = N_1 + \ldots + N_R$.

Recall that the state space of the Markov process underlying a network satifying the above assumptions may be written as $\mathcal{S}(\boldsymbol{N}) = \{\boldsymbol{n} = (n_{1,1}, \ldots, n_{M,R}) \mid n_{k,r} \geq 0, \sum_{k=1}^{M} n_{k,r} = N_r\}$, in which we focus only on the marginal states where $n_{k,r}$ denotes the total number of class-r jobs residing at queue k, either queueing or receiving service.

For queue k, we denote by $\theta_{k,r}$ the mean service demand of class r, which is the product of the mean number of visits with the mean service time of class r jobs at queue k. If queue k is load dependent then the service demand of the job in service is scaled by a load-dependent factor $\alpha_k(n_k)$ if the queue has $n_k = \sum_{r=1}^{R} n_{k,r}$ resident jobs.

With the above definitions, the equilibrium distribution of the network is then given by [2]

$$\pi(\boldsymbol{n}) = \frac{1}{H(\boldsymbol{N})} \prod_{k=1}^{M} \frac{n_k!}{\alpha_k(n_k)} \prod_{r=1}^{R} \frac{\theta_{k,r}^{n_{k,r}}}{n_{k,r}!} \qquad \boldsymbol{n} \in \mathcal{S}(\boldsymbol{N}) \quad (1)$$

where $\boldsymbol{N} = (N_1, \ldots, N_R)$. The normalizing constant $H(\boldsymbol{N})$ in (1) ensures that state probabilities sum to unity.

Throughout the paper we assume arbitrary load-dependent factors $\alpha_k(n_k)$ with $\alpha_k(0) = 1$. Some common assignments are the following: (i) Single-server load-independent station: $\alpha_k(n_k) = 1$; (ii) Station with s_k homogeneous servers: $\alpha_k(n_k) = \min(s_k, n_k)$; (iii) Delay node: $\alpha_k(n_k) = n_k$; (iv) Flow-equivalent server (FES): $\alpha_k(n_k) = X(n_k)$, where $X(n_k)$ is the mean throughput of the subsystem modelled by the FES when this has a population of n_k resident jobs. Note that the flow-equivalent server case only applies to single class models ($R = 1$).

The rest of this paper is organized as follows. Section 3 introduces our exact results for load-dependent models. Integral forms stemming from these developments are obtained in Section 4 and illustrated on representative examples.

3. EXACT SOLUTIONS

3.1 Single-class normalizing constant

For a single-class load-dependent closed queueing network with M nodes, let s_k, $1 \leq s_k \leq N$, be the smallest index for which there exist a constant c_k such that $\alpha_k(n_k) = c_k$, $\forall n_k \geq s_k$. Also, note that the population vector reduces to a scalar $N = N_1$. We are now ready to give an explicit solution for the normalizing constant in single-class closed queueing networks with load-dependent stations.

THEOREM 3.1. *The normalizing constant of a single-class load-dependent closed queueing network with M stations may be written as*

$$H(N) = \sum_{\boldsymbol{0} \leq \boldsymbol{v} < \boldsymbol{s}} g(N - v) \prod_{k=1}^{M} \phi_k(v_k) \quad (2)$$

where

$$\phi_k(v_k) = \begin{cases} \dfrac{\theta_k^{v_k}}{\prod_{j=1}^{v_k} \alpha_k(j)} \left(1 - \dfrac{\alpha_k(v_k)}{\alpha_k(s_k)}\right) & \text{if } v_k > 0 \\ 1 & \text{otherwise} \end{cases}$$

*This work has been partially funded by the European Commission grant H2020-825040 (RADON).

and in which $\boldsymbol{s} = (s_1, \ldots, s_M)$ and $g(N-v)$ is the single-class normalizing constant of a load-independent model with demands $\sigma_k = \theta_k/\alpha_k(s_k)$ and a population of $N - v$ jobs.

Note that the above result is explicit since a closed-form expression for the load-independent normalizing constants $g(N-v)$ has been derived in [4, Eq.(3.12)] and can be computed in $\mathcal{O}(1)$ time and space under a growth of the population N. As such, to the best of our knowledge, equation (2) provides for the first time an explicit solution for load-dependent single-class closed queueing networks that is $\mathcal{O}(1)$ with respect to the population size.

3.2 Multiclass normalizing constant

We now derive a result showing that the normalizing constant $H(\boldsymbol{N})$ of a multiclass load-dependent model with class-independent scaling factors $\alpha_k(n_k)$ may be rewritten as a weighted sum of normalizing constant of single class models. In the theorem below, we use the following definition of n-th order finite difference [7]: $\Delta_n^N f(n) = \sum_{n=0}^{N} (-1)^{N-n} \binom{N}{n} f(n)$. This may be generalized to the multivariable case as $\Delta_{\boldsymbol{n}}^{\boldsymbol{N}}$, which denotes R finite differences of orders $\boldsymbol{N} = (N_1, \ldots, N_R)$ on the variables $\boldsymbol{n} = (n_1, \ldots, n_R)$.

THEOREM 3.2. *In a multiclass closed queueing network with M load-dependent queueing stations and R classes, the normalizing constant may be written as*

$$H(\boldsymbol{N}) = \frac{\Delta_{\boldsymbol{n}}^{\boldsymbol{N}} H_{\boldsymbol{n}}(N)}{N_1! \cdots N_R!} = \sum_{\boldsymbol{0} \leq \boldsymbol{n} \leq \boldsymbol{N}} \frac{(-1)^{N-n}}{N_1! \cdots N_R!} \prod_{r=1}^{R} \binom{N_r}{n_r} H_{\boldsymbol{n}}(N) \tag{3}$$

where $\boldsymbol{n} = (n_1, \ldots, n_R)$, $n = \sum_{r=1}^{R} n_r$, and in which $H_{\boldsymbol{n}}(N)$ is the normalizing constant of a single-class model with M load-dependent stations, population $N = \sum_{r=1}^{R} N_r$, demands $\theta_{k,\boldsymbol{n}} = \sum_{r=1}^{R} n_r \theta_{k,r}$ and scaling factors $\alpha_k(n_k)$ identical to the ones used in the multiclass model.

From (3) we readily see that an explicit form can be obtained for $H(\boldsymbol{N})$ once we replace $H_{\boldsymbol{n}}(N)$ by (2). The computational complexity of (3), instantiated with (2), is $\mathcal{O}(\prod_{k=1}^{M} s_k N^R)$ time and $\mathcal{O}(1)$ space. This form offers computational advantages over the load-dependent mean value analysis (MVA-LD), which is the standard method to assess queueing networks with multi-server stations, since MVA-LD complexity is $\mathcal{O}(N_{max} N^R)$ time and $\mathcal{O}(N_{max}\sqrt{N^R})$ space [5], where $N_{max} = \max_r N_r$. Note in particular that space complexity is reduced to $\mathcal{O}(1)$ with (3).

4. INTEGRAL FORMS

In this section, we develop a integral form for $H(\boldsymbol{N})$ over the complex and real domains. The first form is based on the Norlünd-Rice integral and computes the normalizing constant $H(\boldsymbol{N})$ using R contour integrals in the complex domain. Conversely, the second integral form is on the M-dimensional unit simplex. The two integral forms have complementary computational properties, with the first form being more efficient than the second form for large number of stations M, while the second form being more efficient for large number of classes R.

4.1 Norlund-Rice integral

Note first that the single-class normalizing constant $g(N)$ appearing in (2) can be written as the divided difference [6]

$$g(N) = [\theta_1, \ldots, \theta_M] x^{N+M-1} \tag{4}$$

Because divided differences admit an integral form over the unit simplex through the Hermite-Genocchi formula [1, 6], the last expression may also be rewritten as

$$g(N) = \frac{(N+M-1)!}{N!} \int_{T_M} (\theta_1 u_1 + \ldots + \theta_M u_M)^N d\boldsymbol{u} \tag{5}$$

where $T_M = \{(u_1, \ldots, u_M) : u_1 + \ldots + u_M = 1, u_k \geq 0\}$ is the unit simplex in M dimension.

A well-known property of divided differences is that they may be equivalently computed using the Norlund-Rice integral [7]

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

where i denotes the imaginary unit and the contour of integration encircles the poles at $0, 1, \ldots, n$.

Note that (3) computes $H(\boldsymbol{N})$ using R independent finite difference operators. Since the integrand is a normalizing constant, it is a multivariate polynomial in the demands. Therefore $H_{\boldsymbol{z}}(N)$ is polynomially bounded and holomorphic, which allows us to apply the Norlund-Rice integral to (3) and write

$$H(\boldsymbol{N}) = \frac{1}{(2\pi i)^R} \oint \frac{H_{\boldsymbol{z}}(N)}{\prod_{r=1}^{R} z_r(z_r-1)(z_r-2)\cdots(z_r-N_r)} d\boldsymbol{z} \tag{6}$$

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

To illustrate the above result on a special case, let us consider a model with $R = 2$ classes, where we can denote $h(z_1, z_2) = H_{\boldsymbol{z}}(N)$, with $\boldsymbol{z} = (z_1, z_2)$. When the radius of integration grows asymptotically large, it is possible to show that the circle integral may be rewritten as the double integral

$$H(\boldsymbol{N}) = \frac{1}{(2\pi i)^2} \int_0^{2\pi} \int_0^{2\pi} h(\gamma(t_1), \gamma(t_2)) \times \frac{\gamma'(t_1)}{\gamma(t_1)^{N_1+1}} \frac{\gamma'(t_2)}{\gamma(t_2)^{N_2+1}} dt_1 dt_2$$

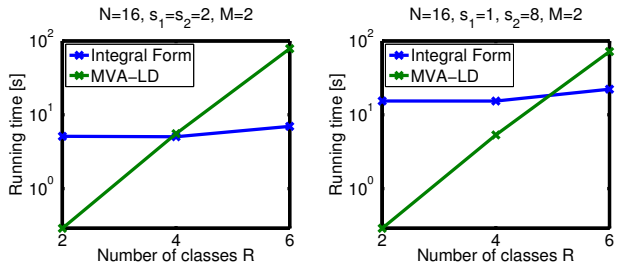
where $\gamma(t) = \cos(t) + i\sin(t)$, $\gamma'(t) = -\sin(t) + i\cos(t)$. The value of $h(z_1, z_2)$ can be calculated at each point using 2 with the load-independent normalizing constant formula in [4, Eq.(3.12)].

4.2 Simplex integral

Let s_k and $\phi_k(v_k)$ be defined as in Theorem 3.1, we can now give a second integral form for $H(\boldsymbol{N})$ based on the finite difference expression we have derived earlier.

THEOREM 4.1. *In a network with multiclass load-dependent queues the normalizing constant can be written as*

$$H(\boldsymbol{N}) = \frac{1}{\prod_{r=1}^{R} N_r!} \int_{T_K} \sum_{\boldsymbol{0} \leq \boldsymbol{v} < \boldsymbol{s}} a_{\boldsymbol{v}} \boldsymbol{\Delta}_{t_0}^{N-v} \boldsymbol{\Delta}_{\boldsymbol{t}}^{\boldsymbol{v}} f(t_0, \boldsymbol{t}, \boldsymbol{u}) \, d\boldsymbol{u} \tag{7}$$



(a) Balanced servers (b) Unbalanced servers

Figure 1: Simplex integral on two load-dependent models with varying number of servers.

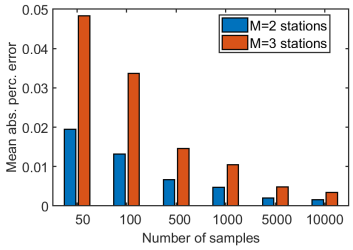


Figure 2: Results - logistic sampling

where $\sigma_{kr} = \theta_{kr}/\alpha_k(s_k)$, $\boldsymbol{v} = (v_1, \ldots, v_K)$, $v = \sum_{k=1}^{K} v_k$, $\boldsymbol{t} = (t_1, \ldots, t_K)$,

$$a_{\boldsymbol{v}} = \frac{(N+K-1-v)!}{(N-v)!} \prod_{k=1}^{K} \phi_k(v_k)$$

and we define $f(t_0, \boldsymbol{t}, \boldsymbol{u}) = \prod_{r=1}^{R} \left(\sum_{k=1}^{K} \sigma_{k,r}(t_k + t_0 u_k) \right)^{N_r}$.

4.2.1 Example

In this example, we consider three models with $M = 2$ multi-server stations and $R \in \{2, 4, 6\}$ classes. The demands are set to $\theta_{k,r} = k \cdot r$. We consider two scenarios differing for the number of servers at the two stations: balanced ($s_1 = s_2 = 2$) and unbalanced ($s_1 = 1, s_2 = 8$).

We first use Grundmann-Möller (GM) cubature rules to exactly integrate over the unit simplex [6]. The total population of jobs is equal to $N = 16$ and we set $N_r = \lceil N/R \rceil$. Integral form (7) is evaluated with a GM rule of degree N. The results shown in Figure 1 illustrate the much greater scalability of the integral form (7) compared to the standard MVA-LD algorithm [5] as the number of classes increases. For larger populations, MVA-LD quickly experience memory bottlenecks since space complexity is quadratic in the population size, whereas it becomes $\mathcal{O}(1)$ in the simplex integral.

We also illustrate the computation of the contour integral (6). The integral is evaluated using a step size $\delta = \pi/k$ in which we increase the number of integration points k. Tables 1 and 2 illustrate the accuracy in approximating the normalizing constant as the number of integration points increases up to $k = 512$. The exact values of $H(\boldsymbol{N})$ are obtained using the convolution algorithm. Both tables refer to the case $R = 2$ and show that with as little as 32 points the

Table 1: Norlund–Rice: Balanced servers, $R = 2$

Integration Step size	Integral Value	Error	Time [s]
$2\pi/16$	21255742.539	7.5	0.09
$2\pi/32$	20448824.691	3.4	0.34
$2\pi/64$	20092657.586	1.6	1.26
$2\pi/128$	19926396.988	0.8	5.43
$2\pi/256$	19846222.427	0.4	18.22
$2\pi/512$	19806874.082	0.2	81.72
$H(\boldsymbol{N})$:	19768018.359	Exact	

Table 2: Norlund–Rice: Unbalanced servers, $R = 2$

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
$H(\boldsymbol{N})$:	24346063.132	Exact	

Norlund-Rice method obtains an approximation for $H(\boldsymbol{N})$ having less than 5% error.

4.2.2 Logistic sampling

The unit simplex integral in (7) may also be estimated numerically by adapting the logistic sampling in [6] to the integrand in (7). An open source implementation has been made available in the JMVA tool, part of the Java Modelling Tools suite [3]. The mode of the integrand is obtained, after an additive logistic transformation, by running a conjugate gradient search, implemented using multiprecision arithmetic to avoid numerical issues associated with finite differences and normalizing constants.

The effectiveness of logistic sampling has been tested against random instances with the following characteristics: number of classes $R \in [1, 2]$; number of stations $M \in [2, 3]$; class populations $N_r \in [1, 5]$; number of servers in station 2 $s_i \in [1, 2]$; random service demands $\theta_{kr} \in [0, 1]$. The average errors are shown in Figure 2, indicating a rapid decrease in the magnitude of the errors as the number of samples grows.

5. REFERENCES

[1] K. E. Atkinson *An Introduction to Numerical Analysis*. John Wiley & Sons, 2nd ed., 1989. 139-177, 1982.

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

[3] M. Bertoli, G. Casale, G. Serazzi. JMT: performance engineering tools for system modeling. *ACM PER*, 36(4), 10-15, March 2009.

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

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

[6] G. Casale. Accelerating Performance Inference over Closed Systems by Asymptotic Methods. *Proc. of ACM SIGMETRICS, POMACS*, 1(1):1–25, 6 2017.

[7] P. Flajolet, R. Sedgewick. Mellin transforms and asymptotics: Finite differences and rice's integrals. *TCS: Theoretical Computer Science*, 144, 1995.

[8] J. J. Gordon. The evaluation of normalizing constants in closed queueing networks. *Operations Research* **38, 5** 863–869.