Matrix Network Analyzer: a New Decomposition Algorithm for Phase-type Queueing Networks (Work in Progress Paper)

Zhuoyuan Li
Imperial College London
London, United Kingdom
zhuoyuan.li22@imperial.ac.uk

Giuliano Casale
Imperial College London
London, United Kingdom
g.casale@imperial.ac.uk

ABSTRACT

This paper proposes a new traffic decomposition method called MNA to solve multi-class queueing networks with first-come first-serve stations having phase-type (PH) service, which generalizes the classic QNA method by Whitt. MNA not only supports open queueing networks but also closed networks, which are useful to model concurrency limits in software systems. Using validation models, we show that under low SCV of service time and inter-arrival time, the new method is on average more accurate than QNA. Therefore, MNA can provide better software performance prediction for quality-of-service management tasks.

CCS CONCEPTS

• **Theory of computation** → *Theory and algorithms for application domains*; *Design and analysis of algorithms*; • **Mathematics of computing** → **Markov processes**; • **Computing methodologies** → **Modeling methodologies**; • **Software and its engineering**;

KEYWORDS

Queueing Theory; Matrix Analytic Method; Marked Markovian Arrival Process; Phase-type Distribution

ACM Reference Format:
Zhuoyuan Li and Giuliano Casale. 2024. Matrix Network Analyzer: a New Decomposition Algorithm for Phase-type Queueing Networks (Work in Progress Paper). In *Companion of the 15th ACM/SPEC International Conference on Performance Engineering (ICPE '24 Companion), May 7–11, 2024, London, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3629527.3651431

1 INTRODUCTION

In software engineering, the stochastic model is a useful tool for analyzing the uncertainty and variability of application workloads. These models offer a way to capture randomness in user behaviors, system loads, and operational environments. By integrating stochastic principles, software engineers can perform nuanced analyses, ranging from system performance and reliability assessments to optimal resource allocation and risk management.

In particular, queueing network models are well-suited to the above aims, helping to analyze the stochastic flow of tasks and data within distributed software systems. By simulating the behavior of resources and the interconnections between different services and components, queueing network models enable a granular analysis of system performance metrics such as response time, throughput, and resource utilization. This level of detail is crucial to accurately predict the behavior of the system under varying load conditions, which is often influenced by stochastic user demands. Furthermore, queueing networks provide insights into potential bottlenecks and resource contention issues, guiding engineers toward targeted optimizations and sizing.

Real workloads often deviate from exponential distributions, yet solution methods for *multi-class* non-exponential queueing networks are limited. In particular, existing methods may not effectively model phase-type (PH) inter-arrival and service times, especially under multiple job classes for which first-come first-serve (FCFS) stations typically do not admit a product-form solution. Using PH models in queueing networks can be beneficial, given that recent work has shown that they can closely fit distributions that are often difficult to represent in a Markovian setting, such as distributions with bounded support [10].

The Queueing Network Analyser (QNA) [13] is a well-known method for non-exponential multiclass queueing networks, focusing on open systems. At the same time, closed and mixed queueing networks also have wide applications such as in layered queueing network analysis. Yet, we notice that for closed queueing networks the matrix analytic method (MAM) [12] commonly used for queueing systems with non-exponential arrivals and service is rarely applied, and even for mixed models a complete theory leveraging MAM is missing. Existing works focus on open models and single-class workloads, e.g. [4, 8, 9]. Multiclass interpolation methods exist, such as the hybrid diffusion-G/G/k methods proposed in [3], however, such methods also do not leverage MAM. Given that MAM is one of the most sophisticated and complete approaches to analyzing non-exponential workloads, we believe that further investigation into the potential of these methods in the context of multiclass queueing networks is warranted. Therefore, in this paper, we introduce a new algorithm, called Matrix Network Analyzer (MNA), which replaces the Langenbach-Belz approximation used in QNA with a MAM solution technique for MMAP/PH/1 nodes [7], allowing us to solve multiple classes queueing networks with FCFS

ICPE '24 Companion, May 7–11, 2024, London, United Kingdom

ACM ISBN 979-8-4007-0445-1/24/05
https://doi.org/10.1145/3629527.3651431

queues with PH service time. In various test cases, MNA provides a more accurate result than QNA. Moreover, MNA is shown to extend to closed queueing networks.

In all cases, we find that MNA compares favorably to QNA in the open queueing network model under low SCV of service time and interarrival time and also delivers high accuracy in closed networks, where results are instead compared to discrete-event simulation. We find in particular that the majority of the models enjoy under MNA a small relative approximation error within a 5 percent margin.

In this paper, Section 2 reviews related work and background, Section 3 details the new method, Section 4 presents the experimental results, Section 5 presents a real case example and Section 6 gives conclusions.

2 BACKGROUND

2.1 Queueing Network Models

A single-class open queueing network consists of a set of M service nodes (or queues), indexed by $i = 1, 2, \ldots, M$, where customers or jobs arrive from outside the network, receive service at one or more nodes, and then leave the network. A closed queueing network similarly consists of a set of M interconnected service nodes (or queues), indexed by, but has a fixed number of customers N, circulating within the network. Unlike open queueing networks, there is no arrival from or departure to the outside. The following components characterize the behavior of a network:

- *Arrival Process*: arrivals to node i follow a stochastic process, with rate λ_i.
- *Service Process at Each Node*: the service times at each node i are random variables. The service discipline (e.g., FCFS also affects performance.
- *Routing Matrix*: a routing matrix $P = [p_{ij}]$ of size $M \times M$ defines the probability p_{ij} that a customer, upon completing service at node i, will proceed to node j. For an open network, there is also a probability p_{i0} for a customer to leave the network after being served at node i.

A multi-class queueing network is similar to a single-class queueing network in terms of network elements, but they have r different customers or jobs classes, indexed by $n = 1, 2, \ldots, r$ and different classes of customers or jobs may have different arrival processes, service processes, and routing matrices.

2.2 Markovian Arrival Process (MAP)

In a PH distribution, events occur upon transitions into an absorbing state, whereas in a Markovian arrival process (MAP), the events may be generated by any specific transition between states. Such transitions are referred to as *apparent transitions*, while those that do not lead to an arrival event are termed *hidden transitions*.

MAPs may be specified by a matrix pair that is $D_0 = [C_{i,j}]$ and $D_1 = [D_{i,j}]$. D_0 contains all the hidden transitions, while D_1 is for the apparent transitions. [2]The generator matrix of the underlying continuous-time Markov chain (CTMC) of this MAP is: $Q = D_0 + D_1$.

The steady-state solution π for this CTMC can be calculated using the global balance equations. The steady-state event arrival rate $\bar{\lambda}$ is then: $\bar{\lambda} = \pi D_1 \mathbf{1}$, where $\mathbf{1}$ = (1,1,..,1,1)

Algorithm 1 MMAP superposition algorithm

```
D_0 = A_0 ⊕ B_0
for i = 1; i ≤ m; i++ do
    D_i = A_i ⊕ 0_{b×b}
end for
for i = 1; i ≤ n; i++ do
    D_{i+m} = B_i ⊕ 0_{a×a}
end for
```

2.3 Phase-type Renewal Process

A PH renewal process is an arrival process, where the interarrival time is an independent and identical PH distribution. The renewal process of a PH, of which the initial probability is α and the subgenerator is T, can be considered as a MAP, where an arrival event is generated when the embedded Markov chain of this PH transitions into the absorbing state and then immediately transitions to an initial state according to the initial probability α. Thus the D_0 and D_1 of the corresponding MAP is $D_0 = T$ and $D_1 = -T\mathbf{1}\alpha$, where $\mathbf{1}$ is a column vector with the same number of rows with T, of which all the element is 1.

2.4 Marked Markovian Arrival Process (MMAP)

The marked Markovian Arrival Process is used to model the arrival process of multiple classes where the arrival process of each class is a MAP. An MMAP modeling the arrival process of n classes can be specified by $n + 1$ matrics that is $D_0, D_1, ..., D_{n+1}$. D_0 contains all the hidden transitions, and D_i contains the transitions generating a arrival event of class i. Using Kronecker sums of the MAP arrival streams that model inter-arrival times of individual classes, an MMAP is readily produced to describe the superposition of the arrival streams [5]; Given two MMAPs modeling the arrival process of m classes and n classes respectively, which can be specified by $A_0, A_1, ..., A_m$ and $B_0, B_1, ..., B_n$, the superposition of these two MMAPs is also an MMAP, which can be used to model the overall arrival process of these $m + n$ classes, specified by $D_0, D_1, ..., D_{m+n}$. Suppose $A_0, A_1, ..., A_m$ are squares matrices of size a, and $B_0, B_1, ..., B_n$ are squares matrices of size b. The superposition method is shown in Algorithm 1, where $\oplus$ stands for Kronecker sum and $\mathbf{0}_{b\times b}$ stands for a $b \times b$ matrix with all the elements being 0.

3 MNA: A NOVEL HYBRID APPROACH

3.1 MNA for Open Models

MNA is a novel approach for open queueing networks that integrates QNA, PH fitting, MMAP superposition, and the MAM. This method follows the same network decomposition principle as QNA but relies on the recent developments in MAM to approximate individual nodes. MNA begins by solving first and second-order traffic equations to determine the mean and the Squared Coefficient of Variation (SCV) of the interarrival times at each node. The notations used below are all listed in Table 1. The first-order traffic equations

are:

$$\lambda_{i,r} = \lambda_{0i,r} + \sum_{j=1}^{N} \lambda_{j,r} p_{ji,r}, \tag{1}$$

$$\lambda_i = \sum_{r=1}^{R} \lambda_{i,r}, \tag{2}$$

$$\lambda_{ij,r} = \lambda_{i,r} p_{ij,r}, \tag{3}$$

After solving these equations, one can obtain all the means of inter-arrival times for each class at all nodes. Thus, the utilization can be calculated as:

$$\rho_{i,r} = \frac{\lambda_{i,r}}{m_i \mu_{i,r}}, \tag{4}$$

$$\rho_i = \sum_{r=1}^{R} \rho_{i,r}, \tag{5}$$

$$\mu_i = \frac{\lambda_i}{\rho_i}, \tag{6}$$

Then, the second-order traffic equations are:

$$C_{S_i}^2 = -1 + \sum_{r=1}^{R} \frac{\lambda_{i,r}}{\lambda_i} (\frac{\mu_i}{m_i \mu_{i,r}})^2 (C_{S_{i,r}}^2 + 1) \tag{7}$$

$$C_{A_{i,j}}^2 = \frac{1}{\lambda_{i,j}} \sum_{j=0}^{N} C_{ji,r}^2 \lambda_{j,r} p_{ji,r} \tag{8}$$

$$C_{A_i}^2 = \frac{1}{\lambda_i} \sum_{r=1}^{R} C_{A_{i,j}}^2 \lambda_{i,j} \tag{9}$$

$$C_{D_i}^2 = 1 + \frac{\rho_i^2 (C_{S_i}^2 - 1)}{\sqrt{m_i}} + (1 - \rho_i^2)(C_{A_i}^2 - 1) \tag{10}$$

$$C_{ij,r}^2 = 1 + p_{ij,r} (C_{D_i}^2 - 1) \tag{11}$$

Upon solving the second-order traffic equations, the means and SCVs of inter-arrival times for each class at all nodes are obtained. The above equations are from QNA for the multi-class open model. [6]

Subsequently, each node is addressed individually. QNA solves the network node-by-node by using the Langenbach-Belz approximation:

$$\alpha_{m_i} = \begin{cases} \frac{\rho_i^{m_i} + \rho_i}{2}, & \text{if } \rho_i > 0.7, \\ \rho_i^{\frac{m_i+1}{2}}, & \text{if } \rho_i < 0.7, \end{cases} \tag{12}$$

$$W_{iq} \approx \frac{\alpha_{m_i}}{\bar{\mu_i}} \frac{1}{1 - \rho_i} \frac{C_{A_i}^2 + C_{S_i}^2}{2}, \tag{13}$$

$$L_{i,r} = \frac{\lambda_{i,r}}{\mu_{i,r}} + \lambda_{i,r} W_{iq}. \tag{14}$$

Instead, for each node, a three-moment matching algorithm [1] is first applied to fit the mean and SCV of interarrival times, and then use the corresponding PH renewal process as a MAP of this class to this node. Next, we superpose these MAPs to produce an MMAP as the overall arrival process to this node. Then, every single queue in the network is solved as a MMAP[R]/PH[R]/1/FCFS queue, allowing to account for class arrival cross-correlations. Indeed, while the superposition of Poisson processes is Poisson, the superposition even of renewal (i.i.d.) processes can produce non-renewal processes (non-i.i.d.) [11], thus the MMAP representation helps us to capture interarrival time covariances introduces in this fashion. He [7] proposes a matrix analytic method for calculating the margin distributions of the queue length of different classes in an MMAP[R]/PH[R]/1/FCFS queue. Finally, the mean queue length of each class can be calculated according to its margin distribution. Combining the utilization acquired in the previous traffic equation, other metrics of this node can be calculated.

Table 1: Notaitons and descriptions of MNA

Notation	Description
$\lambda_{ij,r}$	Mean arrival rate of class r customers from node i to node j
$\lambda_{i,r}$	Mean arrival (departure) rate of class r customers to (from) node i
λ_i	Mean aggregate arrival (departure) rate to (from) node j
$\mu_{i,r}$	Mean service rate of class r customers at node i
μ_i	Mean aggregate service rate at node i
$\rho_{i,j}$	Utilization of node i due to customers of class r
ρ_i	Utilization of node i
m_i	Numbers of servers of node i
$C_{ij,r}^2$	SCV of time between two consecutive class r customers going from node i to node j
$C_{Ai,r}^2$	SCV of interarrival time for class r to node i
C_{Ai}^2	Aggregate SCV of interarrival time to node i
C_{Di}^2	Aggregate SCV of node i inter-departure times
$C_{S_i}^2$	Aggregate SCV of service time of node i
$C_{S_{i,r}}^2$	SCV of service time for customer class r at node i
$p_{ij,r}$	Routing probability of Class r from node i to j
N	Numbers of queueing nodes
R	Numbers of classes
$Q_{i,r}$	Queue length of customer class r at node i
n_r	Numbers of jobs for class r.

3.2 MNA for Closed Models

The MNA method for closed queueing networks is shown in Algorithm 2.

Initialization (Lines 1-4): $\lambda_{i,r}$ is the arrival rate of class r to node i, while $\mu_{i,r}$ is the service rate of class r to node i. These lines set the upper bound of the arrival rate of class r to the reference node equal to the bottleneck service rate of this class.

Initial Guess (Lines 5-7): λ is a vector contains the guessed arrival rates of all the job classes to the reference node, namely $\lambda = (\lambda_{1,1}, \lambda_{1,2}, ...)$. In this first iteration, an initial guess is made that the arrival rate of class r to the reference node is equal to the bottleneck service rate of this class.

Algorithm 2 MNA for closed queueing networks

1: **for** $r = 1; r \leq R; r++$ **do**
2: $\lambda_{1,r}^{\mathrm{UB}} = \min \mu_{i,r}$
3: $\lambda_{1,r}^{\mathrm{LB}} = 0$
4: **end for**
5: **for** $it = 1; it <$ it_max; $it++$ **do**
6: **if** $it == 1$ **then**
7: $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathrm{UB}}$
8: **else**
9: **if** $(\max_i(QN_r - n_r) <$ threshold$)$ **then**
10: **break**
11: **else**
12: $r^* = \arg\max_r \left(\frac{QN_r - n_r}{QN_r}\right)$
13: **if** $(QN_{r^*} > n_{r^*})$ **then**
14: $\lambda_{1,r^*}^{\mathrm{UB}} = \lambda_{1,r^*}$
15: **else**
16: $\lambda_{1,r^*}^{\mathrm{LB}} = \lambda_{1,r^*}$
17: **end if**
18: **end if**
19: $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^{\mathrm{UB}} + \boldsymbol{\lambda}^{\mathrm{LB}})/2$
20: **end if**
21: Solve the network under arrival rate of λ, using MNA algorithm for open queueing networks
22: $QN_r = \sum_{i=1}^{N} Q_{i,r}$
23: **end for**

Arrival Rate Adjustment (Lines 8-20): This step are performing a fixed point iteration, making the guessed arrival rate converge to the real arrival rate. QN_r is the sum of the queueing length of class r, and n_r is the population of job class r in this network. The iteration ends if the difference between QN_r and n_r is small enough for every job class r. Otherwise, MNA adjusts the arrival rate of the class, which has the largest population relative error. If QN_r is larger than n_r, then we set the upper bound of the arrival rate of this class to the guessed value, otherwise, we set the lower bound to the guessed value. MNA uses the average value of the upper bound and lower bound as the new guessed value in the next iteration.

Network Analysis (Line 21-22): Using the guessed arrival rate, the closed queueing network can be solved as an open queueing network. In each iteration, a method very similar to MNA for the open queueing network is called to evaluate the current state of the network given the current arrival rates $\boldsymbol{\lambda}$. The only difference is that, when solving closed models, after getting the margin distribution of the queue length of class r in a node, instead of directly using the mean of this distribution, the mean queue length of this class is calculated by $\sum_{i=1}^{n_r} \frac{P(x=i)i}{\sum_{k=1}^{n_r} P(x=k)}$, where $P(x=i)$ is the probability of this node having a queue length of i. After achieving the queue length of class r at each node, the total population of this class, namely QN_r, can be calculated by the sum of queue lengths at all the nodes. The outcome of this analysis feeds into the next iteration for further adjustment.

In summary, this algorithm initially guesses the arrival rate of the reference node, transforming the closed queueing network into an open queueing network, then applies a fixed point iteration, which adapts the rates based on the network performance in each iteration, aiming to achieve a balance between the actual and desired populations in each class.

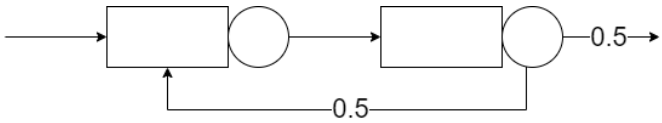

Figure 1: A feedback loop open queueing network

Table 2: Performance of MNA for single-class open queueing network

node	mean ARE	models within 5% ARE
queue 1	0.0201	951/1000
queue 2	0.0242	908/1000
overall	0.0221	871/1000

4 NUMERICAL EVALUATION OF MNA

In this chapter, the improvements of MNA will be shown through a selection of experimental cases, highlighting its higher accuracy compared to QNA. These experiments are conducted with assistance from LINE, a Matlab toolbox created by the QORE lab at Imperial College London. LINE is designed for resolving complex queueing network models using various algorithms or simulations. [3]

4.1 Model Design

To facilitate a comparative evaluation of the performance between QNA and MNA, we have designed a benchmark using an open feedback queuing network with a feedback loop. As depicted in Figure 1, this network comprises several key elements: a source with an arrival rate λ, and two FCFS queues with individual service rates μ_1 and μ_2, respectively.

Jobs generated by the source initially enter Queue 1. After being serviced in Queue 1, they are transferred to Queue 2. Upon completion of service in Queue 2, each job has a probability of 0.5 of proceeding directly to a sink. Conversely, there is a probability 0.5 that the job will be re-routed back to Queue 1.

4.2 Single-class Open Queueing Networks

Consider the following example and its numerical result in Table 2 & Figure 2: the interarrival time, and the service time of node 1 and node 2 are all PH distributed. the mean interarrival time is generated by a random variable uniformly distributed between 40 and 45, the mean service time of node 1 is generated by a uniform distribution between 5 and 8, and the mean service time of node 2 is generated by a uniform distribution between 3 and 6. The SCVs for interarrival time and service time of node 1 and node 2 are all generated by a uniform distribution a uniform distribution between 0.01 and 0.8. Through the random selection of 1000 instances, these conditions are examined empirically.

In the experimental analysis, it is observed in Table 2 that 90 percent of the results obtained from MNA exhibited an Absolute

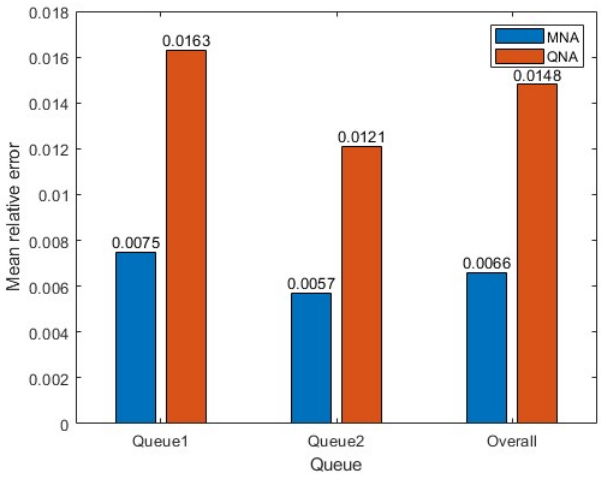

Figure 2: MAE for single-class open queueing networks

Table 3: Interarrival time parameters for multi-class open queueing network test

class	mean	SCV
1	$U(40, 50)$	1/4
2	$U(52, 57)$	1/5

Table 4: Service time parameter for multi-class open queueing network test

node	class	mean	SCV
1	1	$U(2, 5)$	1/6
1	2	$U(2, 5)$	1/7
2	1	$U(3, 6)$	1/9
2	2	$U(3, 5)$	1/2

Relative Error (ARE) within the 5 percent threshold. Additionally, MNA provides a better result for both node 1 and node 2 than QNA. This is evidenced by the lower Mean Absolute Errors (MAE) observed for MNA in Figure 2.

4.3 Multiclass Open Queueing Networks

Considering the following example which uses the same network routing as the last example but has multiple job classes. The interarrival time, and the service time are all phase-type distributed. The means and SCVs are generated as shown in the table 3 and 4 where all the means follow uniform distributions and SCVs are fixed values. For this example, without loss of generality, we set SCV as a fixed value. This is because randomly generated values may include some irrational numbers, which may lead to a MAP with extremely large state space which increases the execution time.

As is shown in Table 5 and Figure 3, MNA demonstrates high accuracy for multi-class scenarios. Approximately 95% of the samples exhibit an ARE within a 5% margin. When compared to QNA,

Table 5: Performance of MNA for multi-class queueing network

node and class	mean ARE	models within 5% ARE
queue 1, class 1	0.0115	993/1000
queue 2, class 1	0.0220	932/1000
queue 1, class 2	0.0201	985/1000
queue 2, class 2	0.0256	891/1000
overall	0.0198	947/1000

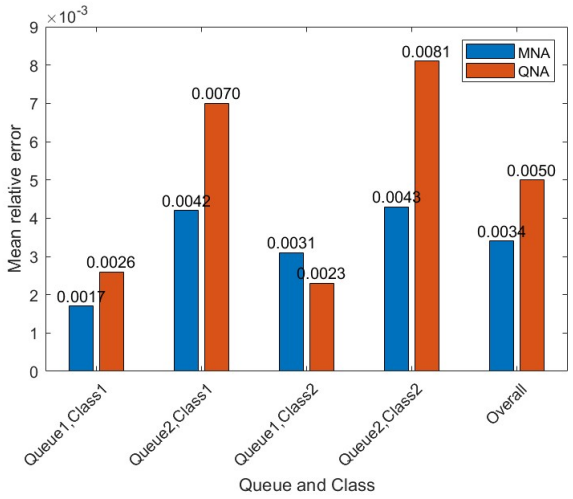

Figure 3: MAE for multi-class open queueing network

Table 6: Performance of MNA for single-class closed models

node	Mean ARE	models within 5% ARE
queue 1	0.0247	942/1000
queue 2	0.0161	967/1000
overall	0.0204	951/1000

MNA offers more accurate results. While there are instances where MNA does not perform as well as QNA for specific classes at certain nodes, it consistently outperforms QNA for the whole system.

4.4 Closed Queueing Networks

Consider this example: a closed queueing network with 3 nodes and 1 class. The first node is a delay node, and the other two are FCFS queueing nodes. The interarrival time, and the service time are all phase-type distributed. The service time of the delay node has a mean of μ, where μ follows a uniform distribution $U(1, 3)$, and the SCV is 4. The service time of the first FCFS node has a mean of μ, where μ follows a uniform distribution $U(2, 4)$, and the SCV is 4. The service time of the second FCFS node has a mean of μ, where μ follows a uniform distribution $U(1, 4)$, and the SCV is 4. The number of jobs is 4, and the jobs follow a circular routing, namely from the delay node to the first FCFS node, then to the Second FCFS node, and finally back to the delay node.

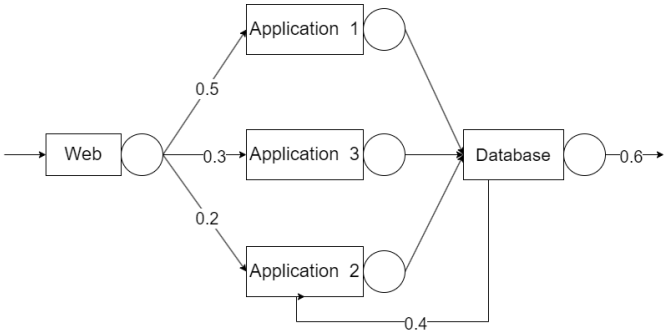

Figure 4: Routing for class 1

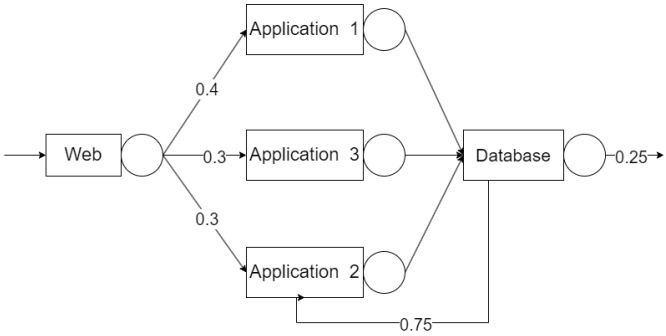

Figure 5: Routing for class 2

As is shown in Table 6, there is only a 2 percent mean relative error between the result of MNA and simulation. 95% of the samples exhibit an ARE within a 5% margin.

5 APPLICATION AND EXAMPLE

We model as a queueing network a three-tier e-commerce system consisting of the following components: *Web Server*: the first point of contact is a web server that handles initial customer requests. *Application Servers*: once the web server processes the initial request, it forwards the customer to one of the three available application servers, depending on their specific needs. *Database Server*: each application server is connected to a central database server responsible for data access.

The system serves two different classes of customers. Upon visiting the website, customers first arrive at the web server. The web server evaluates their needs and routes them to the most appropriate application server for further processing. Once the application server completes its tasks, it interacts with the database server for data storage or retrieval. After being served by the database server, customers have two options: they may continue using the website, in which case they are sent back to the application server for additional services. Alternatively, they may choose to leave the system.

The routing probability of class 1 and class 2 are shown in Figure 4 and Figure 5 respectively. The interarrival time, and the service time are all phase-type distributed. The means of interarrival time of class 1 and class 2 are 4 and 5 respectively, and the SCVs of interarrival time are all 1/10. The means and SCVs of the service time are generated as shown in Table 7. For this example, QNA provides a result with a relative error of 8.6 percent while MNA provides a more accurate result with a relative error of 1.5 percent.

Table 7: Service time parameters for real case example

node	classes	mean	SCV
Application 1	1,2	0.5	1/10
Application 2	1	1	1/10
Application 2	2	0.3	1/10
Application 3	1, 2	0.5	1/10
Database	1	0.3	1/10
Database	2	0.5	1/10

6 CONCLUSION

In this paper, we have proposed MNA, a new algorithm for solving multiclass PH queueing networks that leverages the matrix-analytic method. The method has been shown to improve the accuracy of the class QNA algorithm.

In future work, we seek to integrate additional features into MNA. In particular, the method should be extended to incorporate functionalities such as self-loops, class-switching, and mixed workloads. A comparison with gradient-based methods to seek the fixed point would also be beneficial.

Additionally, throughput calculation for closed multi-class queueing networks may face challenges due to its use of fixed iterations, which may not converge. A more efficient and accurate method, perhaps based on gradient search, may be needed.

REFERENCES

[1] Andrea Bobbio, Andras Horvath, and M. Telek. 2005. Matching Three Moments with Minimal Acyclic Phase Type Distributions. *Stochastic Models* 21 (01 2005), 303–326. https://doi.org/10.1081/STM-200056210
[2] Peter Buchholz, Jan Kriege, and Iryna Felko. 2014. *Input modeling with phase-type distributions and Markov models: theory and applications.* Springer.
[3] Giuliano Casale. 2021. Integrated performance evaluation of extended queueing network models with line. In *Proceedings of the Winter Simulation Conference* (Orlando, Florida) *(WSC '20)*. IEEE Press, 2377–2388.
[4] Giuliano Casale and Peter Harrison. 2012. A class of tractable models for runtime performance evaluation. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*. 63–74.
[5] Giuliano Casale, Andrea Sansottera, and Paolo Cremonesi. 2016. Compact Markov-modulated models for multiclass trace fitting. *European Journal of Operational Research* 255, 3 (2016), 822–833. https://doi.org/10.1016/j.ejor.2016.06.005
[6] Natarajan Gautam. 2012. Analysis of queues: methods and applications. CRC press.
[7] Qiming He. 2012. Analysis of a continuous time SM [K]/PH [K]/1/FCFS queue: Age process, sojourn times, and queue lengths. *Journal of Systems Science and Complexity* 25 (2012), 133–155.
[8] Armin Heindl. 2001. Decomposition of general tandem queueing networks with MMPP input. *Performance Evaluation* 44, 1-4 (2001), 5–23.
[9] András Horváth, Gábor Horváth, and Miklós Telek. 2010. A joint moments based analysis of networks of MAP/MAP/1 queues. *Performance Evaluation* 67, 9 (2010), 759–778.
[10] A Horváth and E Vicario. 2023. Construction of phase type distributions by Bernstein exponentials. In *European Workshop on Performance Engineering*. Springer, 201–215.
[11] MF Neuts. 1989. Structured Stochastic Matrices of M/G/1 Type and Their Applications. *Marcel Dekker* (1989).
[12] Marcel F Neuts. 1994. *Matrix-geometric solutions in stochastic models: an algorithmic approach.* Courier Corporation.
[13] Ward Whitt. 1983. *The queueing network analyzer. The bell system technical journal* 62, 9 (1983), 2779–2815.