Operational Research (2022) 22:5205–5229
https://doi.org/10.1007/s12351-022-00712-2

ORIGINAL PAPER

On general multi-server queues with non-poisson arrivals and medium traffic: a new approximation and a COVID-19 ventilator case study

Carlos Chaves[1] · Abhijit Gosavi[2]

Received: 30 April 2021 / Revised: 10 April 2022 / Accepted: 14 April 2022 /
Published online: 10 May 2022

Abstract

We consider the multi-server, single-channel queue, i.e., a *G*/*G*/*k* queue with *k* identical servers in parallel, under the first-come-first-served discipline in which the inter-arrival process is non-Poisson, the service time has any given distribution, and traffic is of medium intensity. Such queues are common in factories, airports, and hospitals, where the inter-arrival times and service times are typically *not* exponentially distributed, but rather have double-tapering distributions whose probability density functions taper on both sides, e.g., gamma, triangular etc. For these conditions, a new closed-form approximation based on only the mean and variance of the two inputs, the inter-arrival and service times, is presented. Determining distributions of inputs typically requires additional human effort in terms of histogram-fitting and running a goodness-of-fit test, which is avoided here. The new approximation is tested on a variety of scenarios and its performance is benchmarked against simulation. Further, the new approximation is also implemented on a ventilator case study from the recent COVID-19 pandemic to demonstrate its utility in optimizing server capacity. The approximation provides errors typically in the range 1–15% and 31% in the worst case. In systems where data change rapidly and decisions must be made quickly, this approximation will be particularly useful.

Keywords *G*/*G*/*k* queue · Non-Poisson arrivals · Medium traffic · Multi-server queue

✉ Abhijit Gosavi
gosavia@mst.edu

[1] Boeing, Inc., 7755 E Marginal Way S, Seattle, WA 98108, USA

[2] Missouri University of Science and Technology, 210 Eman Bldg, Rolla, MO 65409, USA

1 Introduction

Multi-server queues appear in a variety of systems such as airports, factories, and hospitals. There is naturally significant research interest in analyzing such queues and quantifying their behavior. Two popular approaches for their performance evaluation are closed-form approximations and discrete-event simulation. An advantage of closed-form approximations is that one can plug in values of inputs into their formulas for performance evaluation. In contrast, for discrete-event simulations, one needs expensive software and exact distributions, and, as we shall discuss below, simulations can become very slow as the number of servers increases. Therefore, closed-form evaluation of multi-server queues remains an important problem.

The most general version of this problem is referred to as the *GI*/*G*/*k* or *G*/*G*/*k* queue (see "Appendix" for basic queueing notations), which is typically studied under the following assumptions:

A1. The inter-arrival time and the service time are allowed to have any given distribution (i.e., are generally distributed) and *k*, which exceeds 1, denotes the number of servers in parallel.
A2. There is only one waiting line, i.e., the queue has a single channel.
A3. Customers are served on a *first come first served* basis.
A4. The waiting line capacity is infinite.

We make two additional assumptions about the *G*/*G*/*k* queue for this paper:

A5. The traffic intensity, i.e., the proportion of busy time for the servers, lies between 0.5 and 0.8. This condition is often described as *medium traffic*.
A6. The service time has a *double-tapering* distribution, which is defined herein as one whose probability density function is an increasing function from the minimum value to the mode and a decreasing one from the mode to the maximum value. Such a double-tapering distribution is hence clearly unimodal. Examples of such distributions are triangular, gamma, Erlang, Weibull, and beta, among others.

Exact or approximate procedures (or formulas) have generally eluded the *G*/*G*/*k* queue. In general, non-Poisson arrivals make the analysis of queues far more complicated than the case of Poisson arrivals, but they have been considered more recently in the literature (Jain et al. 2020; Chydzinski 2020) due to the fact that non-Poisson arrivals are common in real-world settings. See Kimura (1994), Kimura (1995), Whitt (1993), Medhi (2003), Azadeh and Salehi (2018), and Yang et al. (2021) for analysis of such queues. For more recent examples of analysis and applications of multi-server queueing models, see Brandwajn and Begin (2016), Tadakamalla and Menascé (2017), and Khayyati and Tan (2021). Specific examples of *G*/*G*/*k* queues with non-Poisson arrivals in the real world include the following: flow shops in manufacturing systems (Altiok 2012), where there are multiple parallel machines (see Fig. 1), airport queues (Mao and Wu 2017), where Identification

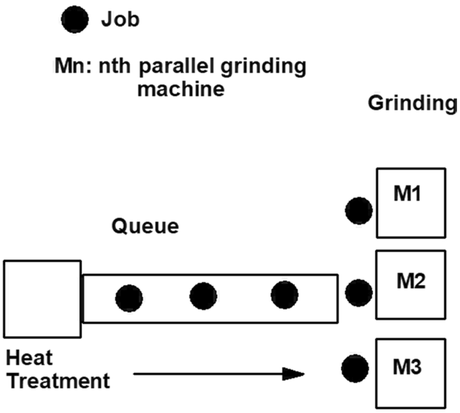


Fig. 1 The queue that forms in front of a parallel set of griding machines for jobs that arrive from heat treatment in a manufacturing plant

Documents (IDs) of travelers are checked by Transportation Security Administration (TSA) agents (see Fig. 2), and hospitals (Raffensperger et al. 2020) with multiple beds in Intensive Care Units (ICUs) that are equipped with ventilators (see Fig. 3). Other examples include packet switching in electronic communication (Hluchyj and Karol 1988; Zhang and Baillieul 2013; Roy et al. 2021). Under the so-called *heavy traffic* condition, i.e., when the traffic intensity exceeds 0.8, the approximation in Sakasegawa (1977) for *G*/*G*/*k* queues is known to be fairly accurate (Robinson and Chen 2011). However, its performance under medium traffic (Assumption **A5**) is known to be unsatisfactory (Hubing 1984).

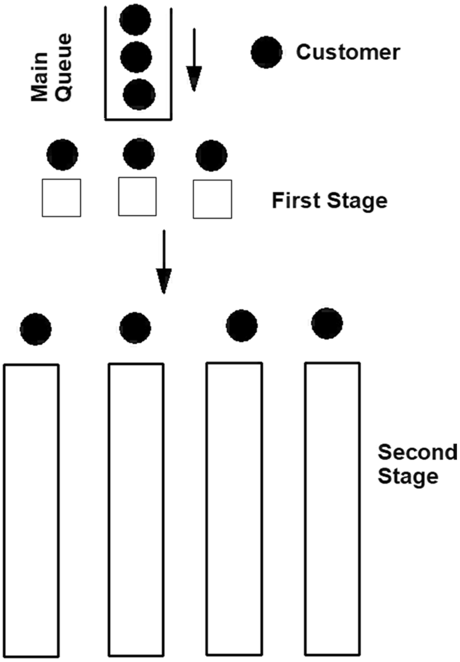


Fig. 2 An airport queueing system in which in the first stage where agents check identification documents of the customers is a multi-server queue

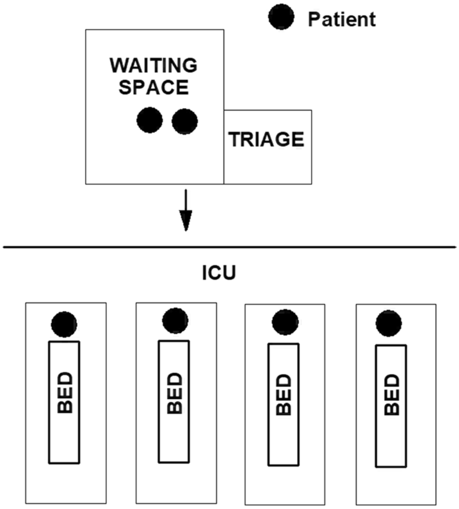


Fig. 3 The queue in a hospital that needs ICU beds is typically virtual within the computer system, but arriving patients assemble in the waiting areas while they are triaged by a nurse

For the so-called *M*/*G*/*k* queue, which is a well-studied multi-server queue with *k* servers, the arrival process is Poisson and the service time can have any given distribution. Existing approximations from the literature have been known to work well for *M*/*G*/*k* queues (Whitt 1993), regardless of the traffic intensity. However, the *M*/*G*/*k* model is *not* applicable when the exponential distribution does *not* hold for the inter-arrival time.

The following evidence points to real-world systems where the *M*/*G*/*k* model does not work. In manufacturing systems, the inter-arrival time for a job is often gamma distributed (Benjaafar et al. 2004), while the service time (production time) can have the gamma distribution in case the machine is failure-prone that leads to high variability (Das and Sarkar 1999). The material in the "Appendix" of the book of Baker and Trietsch (2013) clearly states that the production time on machines is *not* likely to have the exponential distribution. See also Burgin (1975), Muralidhar et al. (1992), and Savsar and Choueiki (2000), which provide extensive evidence of the inter-arrival times and service (production) times carrying the gamma distribution in manufacturing systems, rather than the exponential distribution. For airports, empirical evidence suggests that inter-arrival times commonly have the gamma distribution (Khadgi 2009; Suryani et al. 2010). For the ID checking queue at a TSA security line or at other service counters in an airport, one typically encounters a human server, whose service time is often modeled via the triangular distribution that approximates the beta distribution (Johnson 1997). Finally, Williford et al. (2020) make the case for using the gamma distribution rather than the exponential for the length of stay in a hospital during a serious illness. Thus, clearly, there is a need for studying multi-server queues where the inter-arrival time is not exponentially distributed and the service time is either gamma or triangular (i.e., Assumption **A6**).

Need for approximations When distributions of inter-arrival and service times are available, under restrictive assumptions on the nature of the system, tedious analytical procedures leading to closed-form approaches, involving Laplace transforms (Langaris 1986; Eckberg 1977), embedded Markov chains (Nadarajah 2008), or phase-type distributions (Altiok 2012), can also be used for performance evaluation. When distributions are available, an alternative route is to use discrete-event simulation software (Law 2014), although the latter requires expensive software and becomes sluggish as k increases beyond 10. After simulation became a popular approach in the 1980s, research interest in closed-form approaches waned.

When the analyst is able to derive the distributions of the inputs and has access to a simulation software, or is able to use exact analytical procedures, the approximation suggested in this paper will not be necessary. However, in the real world, means and variances of inter-arrival and service times (inputs) can be estimated with less effort; it is in those circumstances that an *approximation* based on the mean and variance of the inputs, such as the one proposed here, has great practical value. Furthermore, simulation software are expensive, while a closed-form formula within a spreadsheet software is cheaper and easier to use. Several specific scenarios in which such approximations are useful are described below.

First, to meet the needs of automated decision-making within the so-called Cyber-Physical System (CPS) in the era of Industry 4.0 (Tao et al. 2018), methods based on two moments are likely to be more attractive, as fitting distributions requires additional computational effort in terms of histogram fitting, analyzing different distributions, and finally employing a goodness-of-fit test, e.g., the Chi-square test and the Kolmogorov-Smirnov test, to select the best fit. In a CPS, decision-making and controls for hardware are exercised automatically through software written within so-called *digital twins*. In such systems, the requirement of using queueing models remains critical (Sinha and Roy 2019), and therefore models rooted in means and variances will remain attractive because they can be encoded into in-built functions within the hardware of digital twins for rapid computations and control.

Second, in traditional, computerized MRP (Materials Requirements Planning) systems, the proposed approximation based on means and variances of inputs will be useful in estimating lead times approximately. Since production data change frequently, determining distributions is typically ruled out and queueing estimates based on means and variances are popular (Askin and Goldberg 2002). Further, rough estimates of lead times are needed for determining the number of kanbans (Monden 1983), as well as for designing machine capacity (Heragu 2018); sub-optimally designed machine capacity leads to long lead times and increased operational costs (De Treville et al. 2004).

Third, large airports that witness major fluctuations in their demand-arrival patterns during the day use queueing models to determine server capacities—integrating queueing-performance formulas into numerical optimization techniques (Hafizogullari et al. 2003; Manataki and Zografos 2009) to minimize queue waiting times. Finally, hospitals serving critical patients in need of ventilators, e.g., during a pandemic where conditions can alter dramatically every hour, contain $G/G/k$ queues. The ongoing COVID-19 pandemic is making it critical to determine the optimal number of ICU beds equipped with ventilators to save lives; for optimization, these systems need to be modeled as multi-server queuing systems (Raffensperger et al. 2020).

In summary, simple closed-form approximations based on only the mean and variance of inputs (inter-arrival and service times), which can be executed in spreadsheet software or digital twins, continue to hold a special appeal in performance evaluation. Furthermore, even when exact techniques are available, simple approximations with a “back-of-the-envelope” nature (Whitt 1993) and the ability to perform “rough-cut optimization” (Papadopoulos and Heavey 1996) are attractive in practical, real-world settings. Such approximations are generally not very accurate; however, they can be used for rough-cut capacity optimization of machines. As such, even if the approximations are not very accurate, they rapidly provide usable estimates of lead times that help in quick decision-making.

Contributions of this Paper The new approximation in this paper deviates from the literature as follows. It is based on an aggregation procedure of a $G/G/1$ queue and not on the $M/M/k$ formula, unlike the trend in much of the literature (Lee and Longton 1957; Kimura 1986; Shore 1988; Page 1982; Sakasegawa 1977); see “Appendix” for definitions of various multi-server queues, including $M/M/k$. It is shown through extensive numerical testing that the new approximation exhibits more accurate behavior than that of existing $G/G/k$ approximations from the literature (Marchal 1985; Kraemer and Langenbach-Belz 1976). The new approximation is also benchmarked against simulation, as the latter has been commonly used in the literature for that purpose (Altiok 2012; Rabta 2013). The aggregation procedure within our proposed approximation first condenses a multi-server queue into a fictitious single-server queue, via a correction factor for the squared coefficient of variation of the service time, and then retrieves the original single-server queue via another correction factor. Furthermore, the new approximation is developed for Assumptions **A5** and **A6**, which are commonly true of conditions found in many real-world systems, but not studied as widely in the literature. Finally, the new approximation is based on only the mean and variance of two key queueing inputs, the inter-arrival time and the service time, which makes it suitable for automatic computations in manufacturing systems and in airports and hospitals where conditions can change rapidly.

The rest of this article is organized as follows. Theoretical background material and notations for the research in this paper are provided in Sect. 2. Section 3 details the methodology underlying the new approximation. Section 4 presents numerical results obtained from using the new methodology and the benchmarking exercises. Finally, Sect. 5 presents the conclusions drawn from this research, as well as directions for future research.

2 Theoretical background

The section begins with mathematical notation and then discusses two benchmarking models.

2.1 Notation

- k: Number of servers in the single-channel queue
- $\lambda = 1/\, \mathsf{E}$[inter-arrival time]: Mean rate of arrival
- $\mu = 1/\, \mathsf{E}$[service time]: Mean service rate
- $\rho = \lambda/(k\mu)$: Overall server utilization
- $L_q^{G/G/k}$: Mean number of customers in a *G*/*G*/*k* queue
- $W_q^{G/G/k}$: Mean waiting time in a *G*/*G*/*k* queue
- σ_a^2: Variance of the inter-arrival time
- σ_s^2: Variance of the service time of any server
- $C_a^2 = (\sigma_a^2)/(1/\lambda)^2$: Squared coefficient of variation in the inter-arrival time
- $C_s^2 = (\sigma_s^2)/(1/\mu)^2$: Squared coefficient of variation in the service time of any server

From Little's law:

$$L_q = \lambda W_q. \tag{1}$$

Two approximations, described in the following two subsections, have been selected for benchmarking of the new approximation. The reason for selecting them is: they also rely on only the mean and variance of the inter-arrival and service times, making them comparable. Further, both of these approximations are rooted in the so-called *M*/*M*/*k* model, which has been used widely in the literature to develop approximations for multi-server queues.

2.2 Marchal approximation

Marchal (1976) developed an approximation for *G*/*G*/1 queues that was combined with the exact *M*/*M*/*k* formula to develop an approximation for *G*/*G*/*k* queues (Marchal 1985). His *G*/*G*/1 approximation, which holds under Assumptions **A1**: **A4**, is shown below:

$$L_q^{G/G/1} = \frac{\rho^2(1 + C_s^2)(C_a^2 + \rho^2 C_s^2)}{2(1 - \rho)(1 + \rho^2 C_s^2)}. \tag{2}$$

The existing exact formula for an *M*/*M*/*k* queue (Ross 2014) is:

$$L_q^{M/M/k} = \frac{P_0\left(\frac{\lambda}{\mu}\right)^k \rho}{k!(1 - \rho)^2} \tag{3}$$

where

$$P_0 = \frac{1}{\frac{(k\rho)^k}{k!(1-\rho)} + \sum_{m=0}^{k-1} \frac{(k\rho)^m}{m!}}. \tag{4}$$

Note that P_0 above denotes the probability that there are zero customers in the system. Based on his *G*/*G*/1 approximation (given in Eq. (2)), Marchal (1985) developed a correction factor, denoted by *CF*, that when applied to the *M*/*M*/*k* formula works as an approximation for the *G*/*G*/*k* queue. The correction factor is given by:

$$CF = \frac{(1 + C_s^2)(C_a^2 + \rho^2 C_s^2)}{2(1 + \rho^2 C_s^2)}. \tag{5}$$

Combining Eqs. (3) and (5), one has the following approximation (Marchal 1985) for the *G*/*G*/*k* queue:

$$L_q^{G/G/k} = CF \cdot L_q^{M/M/k} = \frac{(1 + C_s^2)(C_a^2 + \rho^2 C_s^2)}{2(1 + \rho^2 C_s^2)} \cdot \frac{P_0(\lambda/\mu)^k \rho}{k!(1-\rho)^2}, \tag{6}$$

where P_0 is as defined in Eq. (4). The above approximation will be referred to as the MAR (short for Marchal) approximation.

2.3 Kraemer and Langenbach-Belz approximation

Kraemer and Langenbach-Belz (1976) developed the following approximation for the *G*/*G*/1 queue, which holds under Assumptions **A1**: **A4**:

$$L_q^{G/G/1} = \frac{\rho^2 (C_a^2 + C_s^2)}{2(1-\rho)} g \tag{7}$$

where

$$g = \exp\left(\frac{-2(1-\rho)(1-C_a^2)^2}{3\rho(C_a^2 + C_s^2)} \right) \text{ when } C_a^2 \leq 1; \tag{8}$$

$$g = \exp\left(\frac{(1-\rho)(1-C_a^2)}{C_a^2 + 4C_s^2} \right) \text{ when } C_a^2 > 1. \tag{9}$$

For benchmarking his own approximation, Marchal (1985) suggested an alternative correction factor from the single-server approximation in Kraemer and Langenbach-Belz (1976), which was:

$$CF = \frac{g(C_a^2 + C_s^2)}{2} \tag{10}$$

in which g is as defined in Eqs. (8)–(9). This leads to the following approximation for the *G*/*G*/*k* queue:

$$L_q^{G/G/k} = CF \cdot L_q^{M/M/k} = \frac{g(C_a^2 + C_s^2)}{2} \cdot \frac{P_0\left(\frac{\lambda}{\mu}\right)^k \rho}{k!(1-\rho)^2} \tag{11}$$

where P_0 is as defined in Eq. (4). The above *G*/*G*/*k* approximation will be referred to as the K-L-B (short for Kraemer and Langenbach-Belz) approximation in this paper.

3 Multi-server aggregation procedure (M-SAP)

The underlying principle in the new multi-server aggregation procedure, referred to as M-SAP for short, is to aggregate (or transform) a single-channel, multi-server queue into a hypothetical *single-server* queue with the same utilization, develop an estimate for the squared coefficient of variation in the hypothetical single-server queue, and then employ this estimate within an existing approximation for *G*/*G*/1 queues to evaluate the original multi-server queue's key performance metrics, i.e., the *expected* waiting time and the *expected* number in the queue. This last step is performed via a correction factor that retrieves the original multi-server queue. Steps in M-SAP are outlined as follows in order to first provide an overview of this procedure:

Step 1: **(Aggregation)** The *G*/*G*/*k* will be aggregated into a hypothetical *G*/*G*/1 queue, i.e., the mean and variance of the service time of this hypothetical *G*/*G*/1 queue will be computed via an aggregation procedure, whose details are provided below in Sect. 3.1.

Step 2: **(Single-Server Approximation)** The expected queue length of the hypothetical *G*/*G*/1 queue, denoted by $\hat{L}_q$, will be computed using either the MAR or the K-L-B approximation and the aggregation procedure of Step 1; the details are provided below in Sect. 3.2.

Step 3: **(Retrieval)** The expected queue length (L_q) of the original *G*/*G*/*k* queue will be obtained via details shown in Sect. 3.3.

Fig. 4 depicts the main idea underlying the M-SAP approximation procedure. In what follows, the three steps in the procedure are described in detail.

3.1 Step 1

The objective here is for the service process in the aggregated single-server queue to behave in a manner similar to that in the original multi-server queue. To this end, a new squared coefficient of variation for the service time of the aggregated queue, $\hat{C}_s^2$, is proposed. The intuition underlying the aggregation is that approximations in queueing networks are often handled via modifications of the squared coefficients of variation of either the service time or inter-arrival time (Buzacott and Shanthikumar 1993), which should ideally lead to *balanced behavior* in the final result. By "balanced behavior," one refers to behavior in the middle

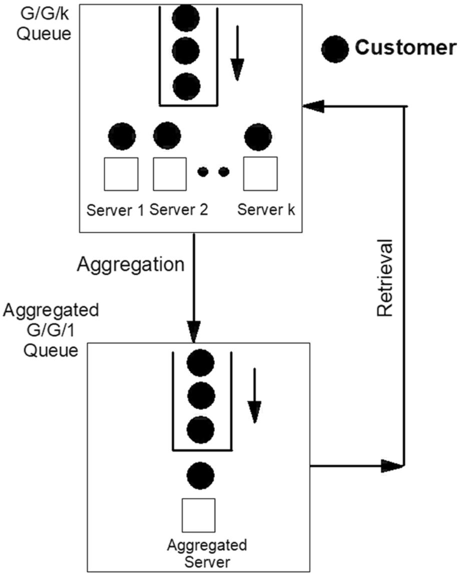


Fig. 4 Schematic showing the aggregation scheme in M-SAP that aggregates the k servers of a $G/G/k$ queue into one server to generate an aggregated single-server queue with modified mean and variance of the service time, as well as the retrieval to obtain the original $G/G/k$ queue

of the spectrum of values obtained of the mean queue length, rather than at the extremes. For instance with the two variables, the squared coefficients of variation of inter-arrival and service times, both variables at the same level (high or low) would represent the *middle* of the spectrum. On the other hand, when the two variables are at conflicting levels, one would obtain behavior at the two ends of the spectrum: one variable at a high level and the other at a low level would represent one extreme of the spectrum, while one at a low level and the other at a high level would represent the other extreme of the spectrum. Hence, the balance here is rooted in the notion that when the variability in one of the inputs (inter-arrival and service times) is high (low), that in the other inputs should also be high (low) to obtain reliable estimates. Although the intuition suggests this kind of behavior, the exact thresholds for what is considered "high" and "low" and the precise expression for modifying the squared coefficient of variation are determined empirically in this paper, i.e., via computational experiments to determine which threshold and which modification leads to the best results vis-á-vis the results from simulations. This entails trial-and-error based experimentation with different combinations of high and low values and benchmarking against simulation to determine which combination delivers the best performance.

Performing computational experiments to identify a suitable replacement for an existing term is common in queueing approximations, although this is a tedious process. For instance, see Sakasegawa (1977), where $W_q^{D/M/1}$ is replaced by $(W_q^{M/D/1} - \frac{\mu}{3})$ in which D denotes deterministic (constant); the reason for this replacement is justified there on grounds of empirically satisfactory results. Also, finding thresholds for determining fields of satisfactory behavior is also common in queueing literature. For instance, the classical heavy-traffic threshold, $\rho > 0.8$, above which heavy-traffic approximations rooted in the normal distribution are known to work in a satisfactory manner, has been determined via computational experiments (Whitt 1993).

From our extensive experimentation, the following thresholds and approximate formulas are proposed herein:

- When the variability in the inter-arrival time is low, i.e., $C_a^2 < 0.3$: the effect of the variability in the service time should be lower to maintain balance and hence the variance is reduced by the number of servers, k:

$$\hat{C}_s^2 \equiv \frac{1}{k}\frac{\sigma_s^2}{(1/\mu)^2}.$$

- When the variability in the inter-arrival time is high, i.e., when $C_a^2 >= 0.3$: the effect of the variability in the service time should be magnified, again to maintain balance, and hence the variance is multiplied by the number of servers, k:

$$\hat{C}_s^2 \equiv k\frac{\sigma_s^2}{(1/\mu)^2}.$$

The two formulas above will be combined as for convenience of representation:

$$\hat{C}_s^2 = \begin{cases} \frac{1}{k}\frac{\sigma_s^2}{(1/\mu)^2}, & \text{if } C_a^2 < 0.3. \\ \frac{k\sigma_s^2}{(1/\mu)^2}, & \text{if } C_a^2 >= 0.3. \end{cases} \tag{12}$$

Since we consider one server to replace a multi-server queueing system, it is necessary to divide the arrival process into k equal parts, and therefore the arrival rate to the aggregated queue will be λ/k; otherwise, one will have an unstable system. The service rate of the aggregated single server will be μ. Taken together, this implies that in the aggregated queue:

$$\rho = \frac{\lambda}{k\mu}. \tag{13}$$

The above is a necessary condition for consistency with the value of utilization in any multi-server queue (Whitt 1993) which should be less than one for stability.

3.2 Step 2

Step 2 will employ a $G/G/1$ approximation for the aggregated single-server queue. Within the approximation, the squared coefficient of service will be used as defined above by Eq. (12) and ρ as defined above by Eq. (13); the value of C_a^2 will not be altered. A *regime*, defined herein as a well-defined area on the graph of which the x-axis is C_a^2 and the y-axis is C_s^2, is constructed for estimating the value of $\hat{L}_q$, i.e., the estimated mean length of the aggregated queue. The regime is described via four sub-areas or scenarios that have been identified on the graph. See Fig. 5 for the geometric structure

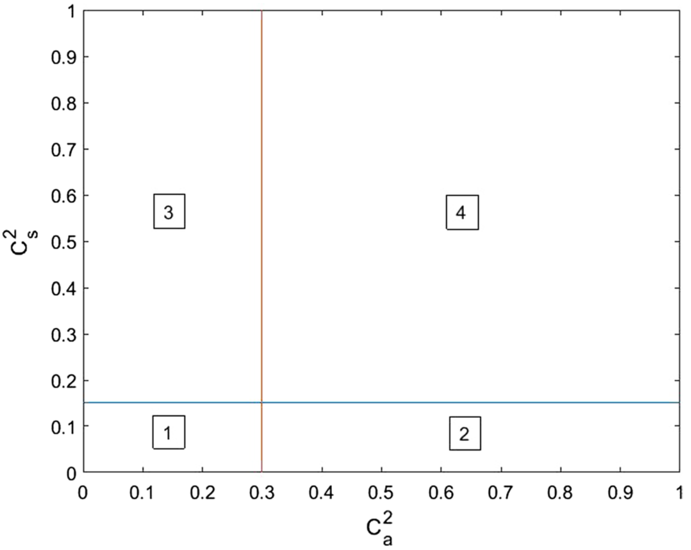


Fig. 5 The number in each box represents the Scenario number

of this regime. As stated above, extensive computational experimentation involving trial and error was conducted that led us to conclude that if the K-L-B rule is used within M-SAP for the hypothetical single-server queue, Eq. (9) works more accurately than Eq. (8) for calculating g.

The approximation formula needed in each scenario within the regime is presented below.

Scenario 1: Conditions: $C_a^2 < 0.30; C_s^2 \leq 0.15$: Use the MAR $G/G/1$ approximation given in Eq. (2) to compute $\hat{L}_q$.

Scenario 2: Conditions: $C_a^2 > 0.3; C_s^2 \leq 0.15$: Use MAR $G/G/1$ approximation provided in Eq. (2) to compute $\hat{L}_q$.

Scenario 3: Conditions: $C_a^2 < 0.3; 0.15 < C_s^2 \leq 1$: Use the K-L-B $G/G/1$ approximation found in Eq. (7) using the value of g computed via Eq. (9) to compute $\hat{L}_q$.

Scenario 4: Conditions: $C_a^2 > 0.3; 0.15 < C_s^2 \leq 1$: Use the MAR $G/G/1$ approximation given via Eq. (2) to compute $\hat{L}_q$.

3.3 Step 3

In this retrieval step, the value of the mean queue length of the original queue is obtained via the following equation that seeks to compress the elongated hypothetical queue by k:

$$L_q = \frac{\hat{L}_q}{k}. \tag{14}$$

The mean wait in the queue can now be computed via Little's law, i.e., Eq. (1). The intuition underlying the proposed approximation for the mean queue length, i.e., Eq. (14), is as follows: Since k servers are aggregated, the resulting variability in the aggregated (fictitious) server is artificially higher, which must be adjusted for in the final calculation. This adjustment is performed by dividing the queue length of the single-server, aggregated queue obtained from the previous two steps by k.

4 Numerical results

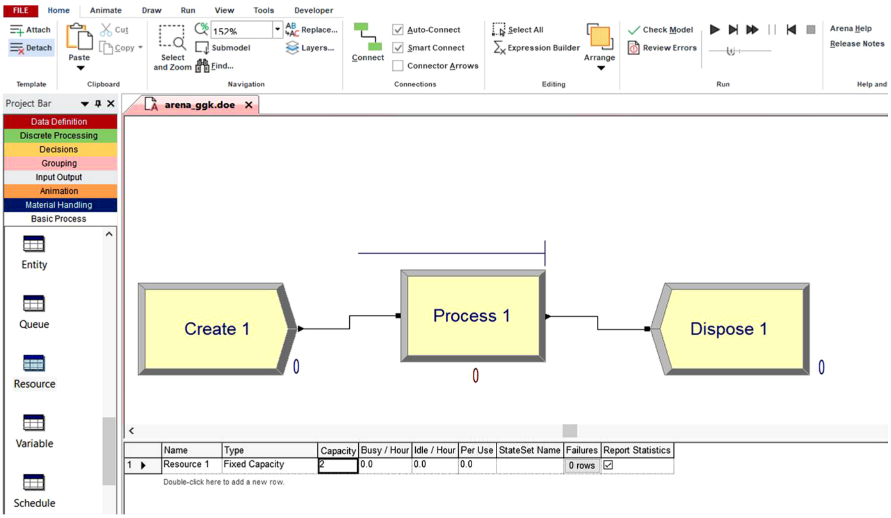
The numerical testing with M-SAP as well as that for the benchmarking techniques was performed under the condition: $C_a^2 < 1$. This condition is standard for most manufacturing, airport, and hospital systems; also when the variance is so high that C_a^2 exceeds 1, higher-order moments are often needed (Buzacott and Shanthikumar 1993; Shore 1988; Marchal 1985), which is beyond the scope of this work.

Computer programs for implementing M-SAP were run on a personal computer in a university setting that used an Intel Pentium Processor with a speed of 2.66 GHz on a 64-bit operating system. The simulation programs used the software ARENA. The M-SAP, MAR, and K-L-B approximations were implemented within the software MATLAB because it provided great flexibility in programming; however, this task can just as easily be carried out in spreadsheet software. The MATLAB program required no more than 5 seconds for any given scenario, while the simulations with ARENA used 10 replications each and needed about 55 seconds per scenario. In addition, every scenario required benchmarking via the two *G*/*G*/*k* models based on MAR and K-L-B; this also needed no more than 5 seconds per scenario. In all, for the four scenarios, a total of 91 cases were tested.

4.1 Numerical evaluation with M-SAP

The approximation was tested under the following conditions: (i) gamma distribution for inter-arrival times, (ii) $\lambda = 1/5$, and (iii) $\rho = \lambda/(k\mu)$ was approximately 0.67 (medium traffic); the value for μ was varied as follows. For $k = 2$, $\mu = 0.15$; for $k = 3$, $\mu = 0.1$; for $k = 4$, $\mu = 0.075$; for $k = 5$, $\mu = 0.06$; for $k = 6$, $\mu = 0.05$; for $k = 7$, $\mu = 0.043$; and for $k = 8$, $\mu = 0.0375$. The different parameters for the inter-arrival times are shown in Table 1. Other double-tapering distributions were not chosen for the inter-arrival time as no evidence was found for them in the literature as suitable choices for the inter-arrival time.

Figure 6 shows a screenshot of the computer program written in ARENA. Key details of this program are as follows: The main computer program is comprised of three modules, CREATE, PROCESS, and DISPOSE. Customers (entities) enter the system through the CREATE module, where the parameters of the inter-arrival time distribution are specified using the following ARENA format: *GAMM* (scale, shape), where *GAMM* denotes the gamma distribution. Within the PROCESS module, the parameters of the service time are specified from one of the following three choices for the model studied here: *GAMM* (scale, shape) for the gamma distribution



		Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	▸	Resource 1	Fixed Capacity	2	0.0	0.0	0.0		0 rows	☑

Double-click here to add a new row.

Fig. 6 A screenshot of the simulation computer program written in ARENA: The main window shows the panel where the main code is written, and the window below it shows the panel in which the number of servers, i.e., the value of k, is assigned, which is 2 for Case 1 in Table 2

Table 1 Parameters in the inter-arrival time gamma distribution for the different values of C_a^2

C_a^2	Ga(scale, shape)
0.05	Ga (0.25, 20)
0.10	Ga (0.5, 10)
0.15	Ga (0.75, 6.67)
0.20	Ga (1, 5)
0.25	Ga (1.25, 4)
0.30	Ga (1.5, 3.333)
0.35	Ga (1.75, 2.8571)
0.40	Ga (2, 2.5)
0.45	Ga (2.25, 2.2)
0.50	Ga (2.5, 2)
0.55	Ga (2.75, 1.8182)
0.60	Ga (3, 1.67)
0.65	Ga (3.25, 1.5385)
0.70	Ga (3.5, 1.4286)
0.75	Ga (3.75, 1.33)
0.80	Ga (4, 1.25)
0.85	Ga (4.25, 1.1765)
0.90	Ga (4.5, 1.1111)
0.95	Ga (4.75, 1.0526)
1.0	Ga (5, 1)

Table 2 Results from Scenario 1 for $k < 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	2	0.10	T (1.6, 6, 12.4)	0.11	0.2122	0.06	95.68	29.70
2	2	0.15	T (1.6, 6, 12.4)	0.11	0.3203	6.55	73.61	16.41
3	2	0.20	T (1.6, 6, 12.4)	0.11	0.4362	11.71	59.79	6.50
4	2	0.25	T (1.6, 6, 12.4)	0.11	0.5632	16.36	48.78	0.17
5	2	0.10	Ga (1, 6.6667)	0.15	0.2901	20.287	65.22	21.93
6	2	0.15	Ga (1, 6.6667)	0.15	0.4057	21.65	53.54	11.64
7	2	0.10	Ga (1, 6.6667)	0.15	0.5259	23.05	45.81	3.51
8	2	0.10	Ga (0.8333, 8)	0.125	0.2579	14.63	71.42	29.92
9	2	0.15	Ga (0.8333, 8)	0.125	0.3601	14.91	62.28	15.14

Table 3 Results for Scenario 1 for $k \geq 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	5	0.05	T (4, 15, 31)	0.11	0.0380	5.42	341.93	1.41
2	5	0.10	T (4, 15, 31)	0.11	0.0777	4.97	227.59	17.69
3	5	0.15	T (4, 15, 31)	0.11	0.1263	14.77	169.86	29.90
4	6	0.20	Ga (1, 20)	0.05	0.0979	16.14	231.96	54.85
5	6	0.25	Ga (1, 20)	0.05	0.1497	5.40	165.97	52.72
6	6	0.30	Ga (1, 20)	0.05	0.2133	20.54	120.91	46.51
7	6	0.35	Ga (1, 20)	0.05	0.2638	25.18	106.35	51.81
8	7	0.15	Ga (2.3256, 10)	0.10	0.0636	16.35	293.01	81.65
9	8	0.15	Ga (4, 6.6667)	0.15	0.0586	7.85	289.76	123.00

and *TRIA*(minimum, mode, maximum) for the triangular distribution. The DISPOSE module allows entities to leave the system. The capacity of the server is specified in the bottom window and it equals k. The time for which the computer program is run and the number of replications is set within the execution panel (not shown in the figure).

Results from the computational work are provided in Tables 2, 3, 4, 5, 6, 7, 8, 9. In these tables, SERT is used to denote service times, and the following acronyms are used for three distributions: Ga (scale, shape) for the gamma distribution and T (minimum, mode, maximum) for the triangular distribution. The error against simulation for the approximation was defined as follows, following the literature (Rabta 2013):

$$Error(\%) = \left| \frac{W_q^{Approx} - W_q^{Sim}}{W_q^{Sim}} \right| \times 100$$

Table 4 Results from Scenario 2 for $k < 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	2	0.35	Ga (4, 6.6667)	0.15	0.7235	27.72	37.99	15.89
2	2	0.40	Ga (4, 6.6667)	0.15	0.8672	16.57	28.93	13.30
3	2	0.45	Ga (4, 6.6667)	0.15	0.9831	13.44	25.91	14.65
4	2	0.50	Ga (4, 6.6667)	0.15	1.1203	8.08	21.18	13.41
5	2	0.55	Ga (4, 6.6667)	0.15	1.2752	2.44	15.85	10.77
6	2	0.4	T (1.6, 6, 12.4)	0.11	1.0204	6.645	24.31	5.33
7	2	0.5	T (1.6, 6, 12.4)	0.11	1.2914	14.26	19.94	9.878
8	3	0.4	T (2.4, 9, 18.6)	0.11	0.9923	16.02	30.07	19.16
9	3	0.60	T (2.4, 9, 18.6)	0.11	1.2923	25.54	17.63	12.69

Table 5 Results for Scenario 2 for $k > 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	6	0.4	Ga (2, 10)	0.1	0.4132	13.16	61.21	35.45
2	6	0.4	Ga (1, 20)	0.05	0.3295	3.13	87.38	48.9
3	7	0.4	Ga (1.86, 12.5)	0.08	0.3125	21.73	77.60	45.84
4	7	0.4	Ga (1.63, 14.29)	0.07	0.2960	21.07	87.67	52.63
5	7	0.45	T (5.6, 21, 43.4)	0.11	0.4224	15.61	55.57	37.48
6	7	0.50	T (5.6, 21, 43.4)	0.11	0.4969	4.54	45.52	33.06
7	7	0.55	T (5.6, 21, 43.4)	0.11	0.5720	3.73	37.96	29.50
8	7	0.60	T (5.6, 21, 43.4)	0.11	0.6793	14.34	25.90	20.61
9	7	0.65	T (5.6, 21, 43.4)	0.11	0.7631	19.65	20.72	17.46

Table 6 Results from Scenario 3 for $k < 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	3	0.10	Ga (2, 5)	0.20	0.2878	10.22	60.76	5.82
2	3	0.15	Ga (2, 5)	0.20	0.3741	13.29	56.37	4.49
3	3	0.20	Ga (2, 5)	0.20	0.4726	18.15	49.71	10.35
4	3	0.25	Ga (2, 5)	0.20	0.5723	22.00	45.01	15.19
5	3	0.1	T (4, 6, 20)	0.13	0.1901	28.94	95.00	19.41
6	3	0.15	T (4, 6, 20)	0.13	0.2708	29.15	80.58	4.90
7	3	0.05	T (2, 18, 20)	0.13	0.1000	21.83	151.99	28.44
8	4	0.10	T (2, 18, 20)	0.09	0.0804	15.42	45.31	7.96
9	4	0.15	T (2, 18, 20)	0.09	0.143	5.61	163.49	16.42

Table 7 Results for Scenario 3 for $k \geq 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	6	0.05	Ga (4, 5)	0.20	0.0766	12.29	184.39	39.52
2	6	0.10	Ga (4, 5)	0.20	0.1169	11.51	153.65	48.65
3	6	0.15	Ga (4, 5)	0.20	0.1698	19.19	120.75	47.48
4	6	0.20	Ga (4, 5)	0.20	0.2212	23.50	104.96	51.06
5	6	0.25	Ga (4, 5)	0.20	0.2830	29.42	87.91	49.28
6	8	0.05	T (6, 12, 62)	0.22	0.0431	5.45	67.68	93.84
7	8	0.10	T (6, 12, 62)	0.22	0.0650	10.89	62.5	131.80
8	8	0.15	T (6, 12, 62)	0.22	0.0968	1.64	68.1	84.7
9	8	0.20	T (6, 12, 62)	0.22	0.1403	13.57	161.26	100.01
10	8	0.25	T (6, 12, 62)	0.22	0.1837	2.76	205.03	149.12

Table 8 Results from Scenario 4 for $k < 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	3	0.30	Ga (2, 5)	0.20	0.7097	29.03	34.20	12.94
2	3	0.35	Ga (2, 5)	0.20	0.8143	6.29	31.98	16.18
3	3	0.40	Ga (2, 5)	0.20	0.9456	1.05	26.61	15.44
4	3	0.45	Ga (2, 5)	0.20	1.0877	7.53	21.32	13.71
5	3	0.50	Ga (2, 5)	0.20	1.2331	12.74	16.96	12.00
6	3	0.55	Ga (2, 5)	0.20	1.3490	15.03	15.98	12.91
7	3	0.60	Ga (2, 5)	0.20	1.4591	16.63	15.63	13.98
8	3	0.65	Ga (2, 5)	0.20	1.6832	23.57	7.50	6.96
9	3	0.35	Ga (2.5, 4)	0.25	0.9029	10.37	27.68	16.78
10	3	0.40	Ga (2.5, 4)	0.25	1.0420	2.63	22.63	15.25
11	3	0.60	T (2.25, 4.5, 23.25)	0.22	1.4900	14.18	15.84	14.83
12	3	0.65	T (2.25, 4.5, 23.25)	0.22	1.5900	15.09	16.73	16.29
13	4	0.60	T (3, 6, 31)	0.22	1.0525	6.49	38.42	39.04
14	4	0.65	T (3, 6, 31)	0.22	1.2183	3.37	29.16	29.27

where *Approx* represents M-SAP, MAR, or K-L-B and *Sim* denotes simulation. M-SAP delivers good performance with the error in the range of 1–15% in most cases; occasionally the error exceeds 30%, but this is rare compared to MAR and K-L-B. In fact, MAR and K-L-B deliver large errors frequently with their errors, exceeding even 150% in many cases. What is important to note is that the performance of M-SAP is *consistently* reliable, whereas it is difficult to predict where MAR and/or K-L-B perform well. It should also be reiterated here that errors are unavoidable with these approximations, as they do not use distributions of the inter-arrival and service times (Whitt 1993; Sakasegawa 1977). However, this

Table 9 Results for Scenario 4 for $k \geq 5$: Entries under M-SAP, MAR, and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	M-SAP	MAR	K-L-B
1	5	0.45	T (3.75, 7.5, 38.75)	0.22	0.7411	19.75	24.43	27.38
2	5	0.50	T (3.75, 7.5, 38.75)	0.22	0.7610	22.80	42.85	37.99
3	5	0.55	T (3.75, 7.5, 38.75)	0.22	0.8914	10.18	32.34	29.54
4	5	0.60	T (3.75, 7.5, 38.75)	0.22	0.9660	6.49	31.34	30.18
5	5	0.65	T (3.75, 7.5, 38.75)	0.22	1.1763	8.34	15.58	15.48
6	6	0.45	Ga (4, 5)	0.20	0.7274	7.75	44.02	34.97
7	6	0.50	Ga (4, 5)	0.20	0.7663	7.49	30.10	24.58
8	6	0.55	Ga (4, 5)	0.20	0.7821	10.40	28.17	24.77
9	6	0.60	Ga (4, 5)	0.20	0.8913	1.36	21.27	19.56
10	6	0.65	Ga (4, 5)	0.20	1.0130	6.89	14.45	13.86
11	7	0.60	T (5.25, 10.5, 54.25)	0.22	0.6934	31.84	36.87	35.63
12	7	0.65	T (5.25, 10.5, 54.25)	0.22	0.7515	26.38	35.33	35.20

approximation delivers reasonable results in settings where distribution fitting is ruled out, as discussed in Sect. 1.

There are cases where MAR or K-L-B perform well, but no pattern can be found for that except for the following condition: $0.7 < C_a^2 \leq 1$. Under this specific condition, K-L-B and MAR perform extremely well because as C_a^2 approaches 1, the inter-arrival time distribution starts approximating the exponential distribution; approximations rooted in the *M*/*M*/*k* formula used by MAR and K-L-B are then naturally appropriate, leading to good performance. While this specific condition is *not* common in the systems studied in this paper, computational results are provided within the "Appendix" to demonstrate the good performance of approximations from the literature under this condition.

4.2 Optimization results with M-SAP

Finally, optimization was performed to illustrate how the M-SAP model is useful for optimizing server capacity. The goal here is to determine the *minimum* server capacity at which the mean waiting time is lower than a pre-set upper threshold. Mathematically, this implies:

$$\text{Minimize } k \text{ such that } W_q < T \text{ where } T \text{ is a pre-set threshold.}$$

The optimal value of k obtained from the optimization exercise is denoted by k^*, while the minimum server capacity needed to obtain a stable system is denoted by $\hat{k}$. Note that $\hat{k}$ can be obtained for any queue by finding the minimum integer at which $\frac{\lambda}{k\mu} < 1$. It should also be mentioned that at a server capacity of $\hat{k}$, the mean wait times are expected to be very long, although finite.

Two cases from the COVID-19 pandemic were used for optimization using available data. The first case is representative of an urban area where the arrival rate is likely to be higher, while the second one is representative of a rural area where the arrival rate is likely to be lower.

Urban Area Hospital from NHS Data Data from the National Health Service (NHS), UK, from the peak of the pandemic in 2020 were gathered from the website (Data 2020), where NHS has made data available. The raw data are provided in the "Appendix" for the reader's convenience. This data led to the following estimates for the length of stay: 13.1972 days with a variance of 4.5456 days-squared. This implies $\mu = 1/(13.1972)$ and $C_s^2 = 0.1186$. The variance in the inter-arrival time is not provided at the NHS website, but will clearly vary from place to place and hence was estimated from other sources. It must be noted, however, that the model used (M-SAP) is general and should be applicable for any given dataset, provided one has access to the mean and variance of the inter-arrival and service times. The inter-arrival time in an urban area was assumed to be 1 per week, i.e., $\lambda = 1/7$ per day with a gamma distribution whose $C_a^2 = 0.15$, from existing data (Raffensperger et al. 2020). The optimization was performed via performance evaluation at each value of k using $T = 0.025$ day or 36 min. Since the M-SAP approach carries out performance evaluation in a very short time period on a computer (requiring no more than 5 seconds), no optimization algorithm was used, but rather the performance was evaluated at all feasible values of k. For this case, $\hat{k} = 93$ and $k^* = 118$. Figure 7 plots the mean waiting time versus the number of ventilators (k) for these data. When $k = \hat{k} = 93$, i.e., k satisfies the stability condition, the mean wait is 1.296 days, which exceeds the threshold, T.

Rural Area Hospital from United States The inter-arrival time in an urban area from the United States was assumed to be 1 per week, i.e., $\lambda = 1/7$ per day with a gamma distribution whose $C_a^2 = 0.15$. The service time was assumed to have a gamma distribution with a mean of 9.1 days, i.e., $\mu = 1/9.1$ per day, and $C_s^2 = 1/2$; both the inter-arrival time and service times in this case were based on data from

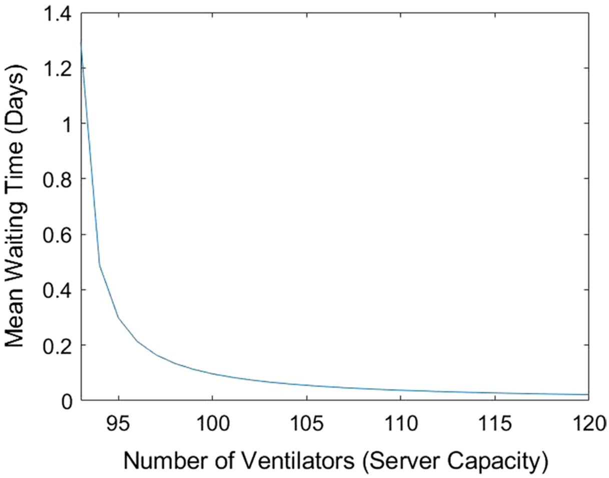


Fig. 7 Plot of mean waiting time (unit of time is day) versus number of ventilators for an urban hospital

Raffensperger et al. (2020). T was set to 0.025 day as in the urban area case. The stability value of ventilator capacity, $\hat{k}$, here is 2 while the optimal value, k^* equals 5. The resulting plot of the mean wait against ventilator capacity is shown in Fig. 8. At the stability condition, i.e., $k = \hat{k} = 2$, the mean waiting time is 1.1165 days, which exceeds the reasonable threshold of 36 min, as in the urban case.

5 Conclusions

A motivating factor for this research was the need to develop closed-form multi-server queueing approximations under the following conditions: (a) traffic intensity is medium, (b) the inter-arrival time is *not* exponentially distributed but carries a double-tapering distribution, and (c) the service time also has a double-tapering distribution. In particular, in many real-world settings, e.g., airports, hospitals, and manufacturing systems, all three conditions apply, which rule out the usage of the fairly accurate, existing $M/G/k$ models or the heavy traffic approximation for $G/G/k$ queues. The non-Poisson arrivals and non-Poisson service rates make these systems difficult to approximate in closed form (Gupta et al. 2010). In the context of a hospital, it must be noted, inaccuracies often lead to under-designed systems with lengthened waits, and waiting beyond acceptable thresholds can cause the patient's death. In airports and factories also, poorly designed systems can cause long, harmful delays. While discrete-event simulation does provide a reliable mechanism to solve problems of this nature, it requires (a) expensive software and (b) distribution fitting. Further, simulations of $G/G/k$ systems can become unacceptably sluggish for large values of k. Therefore, in the real-world, closed-form approximations based on only the mean and variance that are usable within spreadsheet software continue to remain of practical importance.

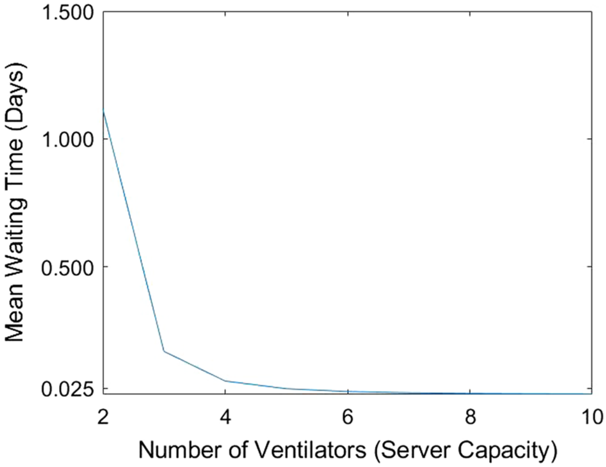


Fig. 8 Plot of mean waiting time (unit of time is day) versus number of ventilators for a rural hospital

The novelty of this work lies in developing a new scheme to aggregate a single-channel, multi-server queue into a fictitious single-channel, single-server queue with the same utilization. The scheme allows one to exploit existing, accurate $G/G/1$ approximations to develop formulas for mean waiting times, rather than the $M/M/k$ formula used extensively in the literature. A conclusion from this study is that for medium-traffic, multi-server queues in manufacturing and service systems, where the inter-arrival time density function and the service time density function are double tapering, M-SAP performs well *consistently* in comparison to the existing MAR and K-L-B approaches from the literature. The research also leads to new insights about the performance of MAR and K-L-B in systems where the performance gradually improves as the inter-arrival time's distribution starts approaching the exponential distribution. Future research in this topic should be directed toward using higher-order moments to address the condition $k \geq 10$ and developing approximations for variance of the waiting times in $G/G/k$ queues.

Appendix

Queueing Notation Glossary: We provide a glossary of terms commonly used in queueing theory for the convenience of the reader:

- Generally Distributed Random Variable: This is a random variable that can have *any given* distribution.
- C^2: Squared coefficient of variation of a random variable: This the variance of a random variable divided by the square of its mean.
- Double-Tapering Distribution: This is a continuous random variable who probability density function tapers on both sides to zero.
- $M/M/k$ Queue: This is a queue with infinite waiting capacity in which there are k servers in parallel, and both inter-arrival and service times have the exponential distribution. M denotes Markovian, which means exponential distribution here, and the first and second letters in this notation denote the distributions of the inter-arrival and service times, respectively. The mean length of this queue can be computed using the well-known, exact formula given via Eqs. (3) and (4).
- $M/G/k$ Queue: This a queue with infinite waiting capacity in which there are k servers, the inter-arrival time has the exponential distribution, and the service time has *any given* distribution. The latter is also often described as the service time being generally distributed.
- $G/G/k$ Queue: This a queue with infinite waiting capacity in which there are k servers, and both the inter-arrival and service times have any given distribution.
- Poisson Arrival and Service Processes: A Poisson arrival process to a system implies that the inter-arrival time is exponentially distributed. Similarly, a Poisson service process implies that the service time is exponentially distributed.
- ρ: This is the traffic intensity of the queue (which equals the proportion of time the servers are busy). This is a positive number and it has to be less than 1 for a queue to be *stable*. In general, only stable queues can be analyzed for steady-state (long-

Table 10 Scenario 2 when $C_a^2 > 0.7$: Entries under MAR and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	MAR	K-L-B
1	5	0.90	T (4, 15, 31)	0.11	1.6689	2.03	1.8
2	5	0.95	T (4, 15, 31)	0.11	1.8533	7.12	6.95
3	5	1.00	T (4, 15, 31)	0.11	1.9310	6.39	6.35
4	6	0.75	T (4.8, 18, 37.2)	0.11	1.2314	2.58	3.33
5	6	0.80	T (4.8, 18, 37.2)	0.11	1.3244	3.74	3.95
6	6	0.85	T (4.8, 18, 37.2)	0.11	1.4304	5.62	5.52
7	6	0.90	T (4.8, 18, 37.2)	0.11	1.5065	5.39	5.17
8	6	0.95	T (4.8, 18, 37.2)	0.11	1.5953	5.94	5.77
9	6	1.00	T (4.8, 18, 37.2)	0.11	1.7946	12.20	12.20
10	9	0.75	Ga (3.0303, 10)	0.10	0.8444	2.9	3.74
11	9	0.80	Ga (3.0303, 10)	0.10	0.8558	1.83	1.53
12	9	0.85	Ga (3.0303, 10)	0.10	0.9666	4.5	4.45
13	9	0.90	Ga (3.0303, 10)	0.10	1.0481	7.00	6.82
14	9	0.95	Ga (3.0303, 10)	0.10	1.1746	12.63	12.47
15	9	1.00	Ga (3.0303, 10)	0.10	1.2403	13.09	13.09
16	9	0.75	Ga (4.5455, 6.667)	0.15	0.8398	2.74	2.61
17	9	0.80	Ga (4.5455, 6.667)	0.15	0.9137	0.21	0.46
18	9	0.85	Ga (4.5455, 6.667)	0.15	1.0522	7.97	7.57
19	9	0.90	Ga (4.5455, 6.667)	0.15	1.0935	6.61	6.21
20	9	0.95	Ga (4.5455, 6.667)	0.15	1.2343	12.98	12.73
21	9	1.00	Ga (4.5455, 6.667)	0.15	1.2689	11.19	11.19

term) behavior. The condition $0 < \rho < 0.5$ is defined as low traffic, $0.5 < \rho \leq 0.8$ as medium traffic, and $0.8 < \rho < 1$ as high traffic.

- Queuing Discipline: This is the order in which customers are served. *First come first served* implies that within a pool of customers waiting in line, the customer who enters first is served first. Other queuing disciplines include *last in first out* and *shortest time first* etc.
- Correction Factor: This is a scalar quantity often used in queueing approximations to multiply the performance metric of a class of queues (e.g., *M*/*M*/*k*) to obtain the corresponding metric for another class of queues (*M*/*G*/*k*).

Compact Representation of Approximation The approximation proposed in this paper can be presented in the following user-friendly format for programming as follows: If $C_a^2 >= 0.7$, use MAR or K-L-B. Otherwise:

- First compute $\hat{C}_s^2$ via Eq. (12) and ρ via Eq. (13).
- Then, compute the mean length in the *G*/*G*/*k* queue as (Tables 10, 11, 12):

Table 11 Inputs for Scenario 4 when $C_a^2 > 0.7$: Entries under MAR and K-L-B columns denote errors in %

Case	k	C_a^2	SERT	C_s^2	W_q^{Sim}	MAR	K-L-B
1	5	0.80	T (3.75, 7.5, 38.75)	0.22	1.6383	0.98	0.13
2	5	0.85	T (3.75, 7.5, 38.75)	0.22	1.7878	4.19	3.38
3	5	0.90	T (3.75, 7.5, 38.75)	0.22	1.9530	7.66	7.03
4	5	0.95	T (3.75, 7.5, 38.75)	0.22	2.1008	9.85	9.49
5	5	1.00	T (3.75, 7.5, 38.75)	0.22	2.1481	7.61	7.61
6	7	0.80	T (5.25, 10.5, 54.25)	0.22	1.2467	0.39	0.46
7	7	0.85	T (5.25, 10.5, 54.25)	0.22	1.3959	6.07	5.28
8	7	0.90	T (5.25, 10.5, 54.25)	0.22	1.4735	6.32	5.28
9	7	0.95	T (5.25, 10.5, 54.25)	0.22	1.5865	8.62	8.25
10	7	1.00	T (5.25, 10.5, 54.25)	0.22	1.6418	7.47	7.47
11	5	0.8	Ga (2.5, 6.67)	0.15	1.5601	2.17	1.92
12	5	0.9	Ga (2.5, 6.67)	0.15	1.4700	9.81	9.81
13	5	0.9	Ga (2.5, 6.67)	0.15	1.89	9.93	9.54
14	7	0.95	Ga (4.6512, 5)	0.20	1.5110	7.33	6.99
15	7	1	Ga (5.814, 4)	0.25	1.67	8.51	8.51

Table 12 Data for length of stay (service time) from March to September, 2020 (Data 2020)

Length of Stay (days)	Frequency
6	10,594
9	3,964
11	55,896
17	23,588
19	1,961
21	9,783
23	3,905

$$L_q = \begin{cases} \frac{\rho^2(C_a^2+\hat{C}_s^2)}{2k(1-\rho)} \exp\left(\frac{(1-\rho)(1-C_a^2)}{C_a^2+4\hat{C}_s^2}\right) & \text{if } C_a^2 < 0.3 \text{ and } 0.15 < C_s^2 \leq 1 \\ \frac{\rho^2(1+\hat{C}_s^2)(C_a^2+\rho^2\hat{C}_s^2)}{2k(1-\rho)(1+\rho^2\hat{C}_s^2)} & \text{otherwise} \end{cases}$$

Acknowledgements The authors want to thank the three anonymous reviewers for their careful reading of the manuscript and for providing numerous suggestions that significantly improved the paper. In particular, the authors thank the first reviewer for the added discussion on using the discrete-event simulation program, the second reviewer for suggesting an appropriate acronym, M-SAP, for the new model, and the third reviewer for numerous comments and suggestions, including one that led to providing a justification for the model.

References

Altiok T (2012) Performance analysis of manufacturing systems. Springer Science & Business Media, New York

Askin R, Goldberg J (2002) Design and analysis of lean production systems. Wiley, New York

Azadeh A, Salehi V et al (2018) Optimum alternatives of tandem g/g/k queues with disaster customers and retrial phenomenon: interactive voice response systems. Telecommun Syst 68(3):535–562

Baker KR, Trietsch D (2013) Principles of sequencing and scheduling. Wiley, Hoboken

Benjaafar S, Kim JS, Vishwanadham N (2004) On the effect of product variety in production-inventory systems. Ann Oper Res 126(1–4):71–101

Brandwajn Alexandre, Begin Thomas (2016) Breaking the dimensionality curse in multi-server queues. Comput Oper Res 73:141–149

Burgin TA (1975) The gamma distribution and inventory control. J Oper Res Soc 26(3):507–525

Buzacott JA, Shanthikumar JG (1993) Stochastic models of manufacturing systems. Prentice Hall, New Jersey

Chydzinski A (2020) Queues with the dropping function and non-Poisson arrivals. IEEE Access 8:39819–39829

Das TK, Sarkar S (1999) Optimal preventive maintenance in a production inventory system. IIE Trans 31:537–551

De Treville S, Shapiro RD, Hameri A (2004) From supply chain to demand chain: The role of lead time reduction in improving demand chain performance. J Oper Manag 21(6):613–627

Eckberg AE Jr (1977) Sharp bounds on Laplace-Stieltjes transforms, with applications to various queueing problems. Math Oper Res 2(2):135–142

Gupta V, Harchol-Balter M, Dai JG, Zwart B (2010) On the inapproximability of $M/G/K$: Why two moments of job size distribution are not enough. Queueing Syst 64(1):5–48

Hafizogullari S, Bender G, Tunasar C (2003) Simulation's role in baggage screening at the airports: a case study. In: Proceedings of the winter simulation conference, pp 1833–1837

Heragu Sunderesh S (2018) Facilities design, 4th edn. CRC Press, Boca Raton

Hluchyj MG, Karol MJ (1988) Queueing in high-performance packet switching. IEEE J Sel Areas Commun 6(9):1587–1597

Hubing N (1984) An approximation for the average waiting time in a $G/G/c$ queue. PhD thesis, North Carolina State University, Raleigh, NC, USA, Department of Electrical and Computer Engineering

Jain M, Kaur S, Singh P (2020) Supplementary variable technique (SVT) for non-Markovian single server queue with service interruption (QSI). Oper Res Int J (in press)

Johnson D (1997) The triangular distribution as a proxy for the beta distribution in risk analysis. J R Stat Soc Ser D 46(3):387–398

Khadgi P (2009) Simulation analysis of passenger check-in and baggage screening area at Chicago Rockford International Airport. North Illinois Univ Eng Rev 1(1):29–34

Khayyati S, Tan B (2021) Supervised-learning-based approximation method for multi-server queueing networks under different service disciplines with correlated interarrival and service times. Int J Prod Res, pp 1–25

Kimura T (1986) A two-moment approximation for the mean waiting time in the $GI/G/s$ queue. Manage Sci 32(6):751–763

Kimura T (1994) Approximations for multi-server queues: system interpolations. Queu Syst 17(3):347–382

Kimura T (1995) Approximations for the delay probability in the $M/G/s$ queue. Math Comput Model 22(10–12):157–165

Kraemer W, Langenbach-Belz M (1976) Approximate formulae for the delay in the queueing system $GI/G/1$. In: Proceedings of the 8th international telegraphic congress, vol 2(3), pp 235/1–235/8

Langaris C (1986) The waiting-time process of a queueing system with gamma-type input and blocking. J Appl Probab 23(1):166–174

Law Averill M (2014) Simulation modeling and analysis, vol 5. McGraw-Hill, New York

Lee AM, Longton PA (1957) Queueing process associated with airlines passenger check-in. Oper Res Quart 10:56–71

Manataki IE, Zografos KG (2009) A generic system dynamic based tool for airport terminal performance analysis. Trans Res C Emerg Technol 17(4):428–443

Mao X, Wu Z (2017) The optimizing of the passenger throughput at an airport security checkpoint. Open J Appl Sci 7(09):485

Marchal WG (1976) An approximation formula for waiting times in single-server queues. AIIE Trans 8:473

Marchal WG (1985) Numerical performance of approximate queuing formulae with application to flexible manufacturing systems. Ann Oper Res 3:141–152

Medhi J (2003) Stochastic models in queueing theory, 2nd edn. Academic Press, Amsterdam

Monden Yasuhiro (1983) Toyota production system. An Integrated Apprpach to Just-In-Time

Muralidhar K, Swensethj SR, Wilson RL (1992) Describing processing time when simulating JIT environments. Int J Prod Res 30(1):1–11

Nadarajah S (2008) Probabilities for queueing systems with embedded Markov chains. Stoch Anal Appl 26(3):526–536

NHS Data (2020) Data for length of stay in UK made available by NHS, https://www.england.nhs.uk/statistics/statistical-work-areas/covid-19-hospital-activity/

Page E (1982) Tables of waiting times for $M/M/n$, $M/D/n$ and $D/M/n$ and their use to give approximate waiting times in more general queues. J Oper Res Soc 33(5):453–473

Papadopoulos HT, Heavey C (1996) Queueing theory in manufacturing systems analysis and design: a classification of models for production and transfer lines. Eur J Oper Res 92(1):1–27

Rabta B (2013) A hybrid method for performance analysis of G/G/m queueing networks. Math Comput Simul 89:38–49

Raffensperger JF, Brauner MK, Briggs RJ (2020) Planning hospital needs for ventilators and respiratory therapists in the COVID-19 Crisis. RAND Corporation: https://www.rand.org/pubs/perspectives/PEA228-1.html

Robinson LW, Chen RR (2011) Estimating the implied value of the customer's waiting time. Manufact Serv Oper Manage 13(1):53–57

Ross SM (2014) Introduction to probability models, 11th edn. Academic Press, San Diego

Roy A, Pachuau JL, Saha AK (2021) An overview of queuing delay and various delay based algorithms in networks. Computing, pp 1–39

Sakasegawa H (1977) An approximation formula $l_q = \alpha\rho^{\beta}/(1-\rho)$. Ann Inst Stat Math 29(1):67–75

Savsar M, Choueiki MH (2000) A neural network procedure for kanban allocation in JIT production control systems. Int J Prod Res 38(14):3247–3265

Shore H (1988) Simple approximations for the $GI/G/c$ queue-I: the steady-state probabilities. J Oper Res Soc 39(3):279–284

Sinha D, Roy R (2019) Scheduling status update for optimizing age of information in the context of industrial cyber-physical system. IEEE Access 7:95677–95695

Suryani E, Chou S-Y, Chen C-H (2010) Air passenger demand forecasting and passenger terminal capacity expansion: a system dynamics framework. Expert Syst Appl 37:2324–2339

Tadakamalla V, Menascé DA (2017) Analysis and autonomic elasticity control for multi-server queues under traffic surges. In: 2017 International conference on cloud and autonomic computing (ICCAC), pp 92–103. IEEE

Tao F, Qi Q, Liu A, Kusiak A (2018) Data-driven smart manufacturing. J Manuf Syst 48:157–169

Whitt W (1993) Approximations for the $GI/G/m$ queue. Prod Oper Manag 2:114–161

Williford E, Haley V, McNutt L, Lazariu V (2020) Dealing with highly skewed hospital length of stay distributions: the use of Gamma mixture models to study delivery hospitalizations. PLoS ONE 15:1–18

Yang D-Y, Chang P-K, Cho Y-C (2021) Optimal control of arrivals in a G/G/c/K queue with general startup times via simulation. Int J Manage Sci Eng Manage 16(1):27–33

Zhang B, Baillieul J (2013) A novel packet switching framework with binary information in demand side management. In: 52nd IEEE conference on decision and control, pp 4957–4963. IEEE