Technical Report
Numerical Validation of Fill Rate Estimation Methods for Two- and Three-Demand Class Rationing Policies with One-for-One Replenishment and General Lead Time Distributions

Oğuzhan Vıcıl & Peter Jackson

January 8, 2015

Abstract

In this report, we conduct numerical simulations of two- and three-demand class inventory threshold rationing systems under one-for-one replenishment policies. The performance metrics of interest are the fill rates of the high priority demand classes (the *gold* fill rate in the two-demand class system and the *platinum* and *gold* fill rates in the three-demand class system). Our main interest is in the sensitivity of these fill rates to the form of the replenishment lead time probability distribution and the resulting quality of approximation methods used to estimate these fill rates. We consider three approximation methods: what we call the *single cycle approach* attributed to Dekker et al and Deshpande et al, the *embedded Markov chain approach* of Fadigloglu and Bulut, and the *continuous time Markov chain* approach of Vicil and Jackson. We confirm the superiority of the embedded Markov chain approach for the case of constant lead times but we find that the fill rates are relatively insensitive to the form of the lead time distribution and both latter approaches, the embedded Markov chain approach and the continuous time Markov chain approach, perform well over wide ranges of lead time variability. For the three-demand class system, we demonstrate that it is possible to achieve highly differentiated fill rates by demand class and show that these fill rates can be estimated with high accuracy using the continuous time Markov chain approach, provided the fill rate of the lowest priority demand class (the *silver* fill rate) is not too low.

1 Introduction and Literature Review

Inventory rationing among different customer classes arises in several contexts. Our primary motivation is the situation of managing service parts inventory in a parts distribution center serving multiple customers, each of whom has contracted for a specific level of customer service, typically measured as fill rate. The different contractual fill rates result in a classification of the customers ranked by priority (eg. platinum, gold, silver, and bronze levels of service where platinum service has the highest contractual fill rate and bronze the lowest).

The forces driving the importance of this problem are the increasing need to provide customer differentiated service as market niches are discovered and the need to limit the growth in inventory in support of this differentiation. Unless demand is pooled and served from a common inventory stock, there can be severe diseconomies of scope. That is, an organization which is forced to maintain separate inventories for each customer class (as some contracts require) will likely carry significantly more safety stock than would be required if demand and inventory were pooled. As total demand is split into finer customer classes, the diseconomy of this approach grows. The inventory management solution is to allow inventory to be pooled but to enforce a rationing discipline which ensures each customer class experiences the service level for which it has contracted.

This area has been an active subject of research for several decades. It remains a challenging problem because of the difficulty of computing exact or accurate performance measures. In this paper, we explore a specific critical-level (threshold) rationing policy in concert with a continuous review $(S-1, S)$ replenishment policy. Such policies are used in practice and we provide approximations for the special case of backorders, Poisson demand process, generally distributed lead times, and two customer classes.

Dekker et al. (1998) use a continuous review $(c, S-1, S)$ policy for two demand-classes. The demand process is a Poisson process and order lead times are constant. Their model is based on the assumption that excess demand is backordered. They use a hitting time approach under the approximating assumption that there was no order outstanding a lead time ago. The accuracy of the approach can be increased by assuming instead that there was no order outstanding two lead times ago. Dekker et al. (2002) consider a $(c, S-1, S)$ replenishment policy for $n-$demand

classes (c is an $n-$dimensional vector in this case). Their model includes lost sales, Poisson demand processes, and a general lead time distribution. The lost sales character of the problem simplifies the state space. They derive the exact steady state distribution of on-hand inventory and from there develop techniques to find optimal policy parameters.

Deshpande et al. (2003) use a continuous review (c, s, Q) policy for two-demand-classes (the parameter c is the critical level for on-hand inventory below which low priority customers are not served). In their model, backorders are allowed, demand is a Poisson process, and the order lead time is constant. They allow multiple replenishment orders to be present in the pipeline at the same time. Nevertheless, they too use a hitting time approach with a creative approximation to the distribution of backorders among customer classes at the time a replenishment order arrives. Empirical results demonstrate that the approximation is quite good for the parameters considered. Deshpande and Cohen (2005) extended their threshold clearing mechanism from 2 to N-demand-classes.

The problem we consider is most closely related to the models in Dekker et al. (1998). For zero setup costs, it is also identical to the model of Deshpande et al. (2003). We focus on $(S-1, S)$ replenishment policies because these are appropriate in the high-cost, low-demand-rate service parts distribution contexts of our applied work. We also assume a fixed threshold policy and seek to determine the provided service levels for each customer class.

In a more recent work, Fadiloglu and Bulut (2010) consider a model which is identical to the one developed in this paper but restricted to a constant lead time. They suggest that an embedded Markov chain approach can be used to estimate the stationary probability distribution by sampling the system at multiples of the lead time. The transition probabilities are approximated under the assumption that delivery times are independent of the number of low-priority backorders. They provide a recursive procedure for computing the transition probabilities of the Markov chain. The stationary probabilities are computed as the limit of a convergent sequence of bounds using a sophisticated technique from computational algebra. They demonstrate through simulation that the approximation is quite good. Our approach can be seen as an application of the same assumption to general lead time distributions. Instead of a Markov chain approach, we are led to the analysis of a continuous time Markov process. We refer to our approach as the continuous time Markov

chain (CTMC) approach to distinguish it from the embedded Markov chain approach of Fadiloglu and Bulut.

2 The Two Demand-Class Model

We first consider a model with two priority demand classes: *gold* and *silver*. The *gold* customers have contracted for a higher level of service, expressed as fill rate, than the *silver* customers. Rather than dedicate completely separate inventories to these two types of customers, the service provider opts to use an inventory pool common to both customer types. The service provider provides differential levels of service between the two customer classes by means of preferential stock allocation policies. In particular, a reserve level of inventory, denoted by S_g, is held for use by *gold* customers only. That is, as long as an arriving demand would not reduce on-hand inventory below the level S_g, it is satisfied from the common pool without respect to its demand class. On the other hand, any *silver* demands that would otherwise reduce on-hand inventory below S_g are backordered. Furthermore, the delivery of a replenishment order is used first to satisfy any *gold* backorders, if any, and then to replenish the *gold* reserve inventory. Only when on-hand inventory would otherwise exceed the level S_g is a delivery used to satisfy *silver* backorders. If on-hand inventory is at or above level S_g and there are no further *silver* backorders, then deliveries are added to the common pool and the on-hand inventory level is allowed to exceed S_g.

Observe that there are two allocation policies at play: one for when a demand occurs and one for when a delivery order is received. The first is known as the threshold rationing policy and the second is known as the priority clearing mechanism. Both policies are specified using the single threshold level, S_g. Unlike traditional single-priority-class inventory models, it is possible under this policy to experience both backorders (for *silver* customers) and on-hand inventory (reserved for *gold* customers).

We assume the demand streams for *gold* and *silver* customers are independent Poisson processes with demand rates λ_g and λ_s, respectively. We further assume that replenishment orders are placed according to an $(S-1, S)$ policy based on inventory position. Hence, the arrival of any demand, either by a *gold* or a *silver* customer, triggers an immediate replenishment order of size 1. The

parameters (S, S_g) completely specify the replenishment and allocation policies. The overall policy is referred to as a *lot-for-lot replenishment and threshold allocation policy.* The delivery lead times for successive orders form a sequence of independent, identically distributed random variables with mean T. In this paper, we consider simulations of the system using a variety of lead time probability distributions including the constant, the exponential, the Erlang, the gamma, the geometric, and the lognormal distributions.

We assume the demand streams for *gold* and *silver* customers are independent Poisson processes with demand rates λ_g and λ_s, respectively. We further assume that replenishment orders are placed according to an $(S-1, S)$ policy based on inventory position. Hence, the arrival of any demand, either by a *gold* or a *silver* customer, triggers an immediate replenishment order of size 1. The parameters (S, S_g) completely specify the replenishment and allocation policies. The overall policy is referred to as a *lot-for-lot replenishment and threshold allocation policy.* The delivery lead times for successive orders form a sequence of independent, identically distributed random variables with mean T. In this paper, we consider simulations of the system using a variety of lead time probability distributions including the constant, the exponential, the Erlang, the gamma, the geometric, and the lognormal distributions.

Vicil and Jackson (2014) provide a general algorithm to determine optimal levels of the policy parameters S and S_g to minimize inventory investment costs subject to service level constraints for both the *gold* and *silver* customers. This algorithm requires an efficient method for estimating the service levels (fill rates) for both customer demand classes for any given combination of parameters $(S, S_g; \lambda_g, \lambda_s, T)$ and for that method to be robust with regard to the underlying lead time probability distribution. We find that assuming an exponential lead time probability distribution works well and compares favorably to other approximation methods.

Let β_g (respectively, β_s) denote the steady state fill rate for *gold* (respectively, *silver*) customers as functions of the parameters $(S, S_g; \lambda_g, \lambda_s, L)$. Denote the stationary probability distribution of a random process by $P_\infty(\cdot)$. By the PASTA principle (Tijms (1996) p. 51), arriving demands face the stationary distribution of on-hand inventory, OH. A *silver* customer arrival will be served if and only if $OH > S_g$, whereas a *gold* customer arrival will be served if and only if $OH > 0$.

Consequently,

$$\beta_s = 1 - P_\infty(OH \leqslant S_g),$$

and

$$\beta_g = 1 - P_\infty(OH = 0).$$

The silver fill rate, β_s, is easily determined using Palm's Theorem.

Proposition 2.1 *For a general, positively-valued lead time distribution with no probability mass at zero, the silver fill rate is given by*

$$\beta_s = \sum_{k=0}^{S-S_g-1} \frac{(\lambda T)^k \, e^{\lambda T}}{k}$$

To implement the lot-for-lot replenishment and threshold allocation policy at any decision point (i.e. at the arrival of a demand or the delivery of an order), the inventory manager requires current knowledge of the on-hand inventory level, OH, the number of gold backorders, B_g, the number of silver backorders, B_s, and the number of units in re-supply, R. Because the system follows lot-for-lot replenishment, it must be the case at every point in time, t, that:

$$OH(t) = [S - R(t) + B_s(t)]^+ \tag{1}$$

and

$$B_g(t) = [R(t) - B_s(t) - S]^+ . \tag{2}$$

where x^+ is defined to be $\max(x, 0)$. Consequently, this policy can be implemented at any decision point knowing only two state variables, $(R(t), B_s(t))$.

Palm's Theorem implies that the stationary distribution of $R(t)$ for general lead time distributions is identical to that obtained when the lead time is exponentially distributed, with the same mean. A similar result obtains for the stationary distribution of $(R(t), B_s(t))$ if the following condition holds:

Definition 1 *The Independence Condition is said to hold if, whenever the state of the system* $(R, B_s) = (r, b_s)$ *at an arbitrary point in time* t*, the probability of a unit delivery in the interval* $(t, t+h)$ *for an infinitesimally small* $h > 0$ *does not depend on the value of* b_s*.*

Observe that this condition is very like that used in the embedded Markov chain approach for constant lead times. The independence condition holds in the case of exponentially disributed lead times because of the memoryless property of the exponential distribution. Vicil and Jackson (2014) show the following:

Theorem 2.1 *Assuming a general, positively-valued lead time distribution having finite mean,* T*, with no probability mass at zero, then, if the independence condition is true, the steady state distribution of* (R, B_s) *satisfies the same balance equations as a system with an exponential lead time distribution with the same mean.*

Vicil and Jackson (2014) provide an algorithm for computing the stationary distribution of (R, B_s) in the special case of exponentially distributed lead times.

The theorem highlights the essential difficulty of exact analysis for this problem: dependence of the probability distribution of delivery times of units in resupply on B_s, the number of silver backorders. The theorem holds for the trivial case of exponentially distributed lead times. On the other hand, if the dependence is weak, the theorem suggests that the stationary distribution under exponentially distributed lead times might lead to a very good approximation for general lead time distributions. It is this conjecture which motivates the experimental studies of this paper. We refer to using the results from exponential lead time distributions to approximate general lead time distribution situations as the *continuous time Markov chain* (CTMC) approach.

Because the Independence Condition is central to both the embedded Markov chain approach for constant lead times and the continuous time Markov chain approach for general lead time distributions, we investigate it in some detail. It is well-known that if we condition on the total number of Poisson arrivals in the interval $(t - L, t]$, say, r_0, then the unordered demand arrival times would be distributed as r_0 independent random variables, each uniformly distributed on $(t - L, t]$. Under the $(S - 1, S)$ policy, each demand arrival triggers a replenishment order that is

to be received L periods later. Consequently, the replenishment order delivery times in $(t, t+L]$ would be distributed as r_0 uniform random variables on $(t, t+L]$. As Fadiloglu and Bulut note, this property is no longer guaranteed to hold when one conditions also on the value of B_s, the *silver* backorders. In the following section, we report on simulation experiments which demonstrate, indeed, that the distribution of replenishment order delivery times in $(t, t+L]$ is not uniformly distributed, when the value of $B_s(t)$ is known. Nevertheless, Fadiloglu and Bulut report that the embedded Markov chain approach works quite well for constant lead times. The purpose of this paper is to confirm that result and to show how well the continuous time Markov chain approach works.

2.1 Testing the Independence Condition

In this section we describe a simulation study used to test the independence condition. Suppose lead times are constant with value L. As mentioned in the previous section, if it is known only that r units are in resupply at time t, then the unordered demand arrival times will be uniformly distributed over the interval $(t-L, t]$. The independence condition is another way of saying that the silver backorders provide no additional knowledge of arrival times of these units. Consequently, we can test the independence condition by simulating the system with constant lead times and comparing the distribution of the unordered arrival times of units in resupply with the uniform distribution, for different values of B_s, the *silver* backorders.

In this simulation study, we take samples at times whenever a demand of any class occurs or a unit is received from the resupply provided that system state is $\big(R(t) = r_0, B_s(t) = b_0\big)$. In other words, whenever a demand occurs, and this new customer sees the system state as $\big(R(t) = r_0, B_s(t) = b_0\big)$, we look at the pipeline vector and collect these data before any change occurs in system states. In addition, if a unit is received from resupply and the new system state becomes $\big(R(t) = r_0, B_s(t) = b_0\big)$, then we collect this data for the pipeline vector.

Furthermore, to prevent any possible correlation between samples, if a sample is taken at time, say t', and the last unit in the pipeline at time t' would arrive by time t'', then we do not collect any sample until time t'' passes.

For each case, we collect at least $10,000$ samples, each one yielding a $\big(R(t) = r_0\big)$ dimensional

pipeline vector. Note that the components of each vector represent the ***remaining time*** until delivery and must lie in the interval $(0, L]$. Based on these data, we construct a histogram of unordered demand arrival times from the $r_0 \times 10,000$ realizations. If the conjecture is true, then the histogram should resemble that of a uniform distribution.

For the following series of simulations, we set $S = 4$, $S_g = 2$, $\lambda_s = 1.5$, $\lambda_g = 20$ and $L = 0.5$. We also set number of bins as 20. We condition on the system state $R(t) = 12$ and different values of B_s to see what the histograms look like. The resulting histograms are displayed in Figure 1.

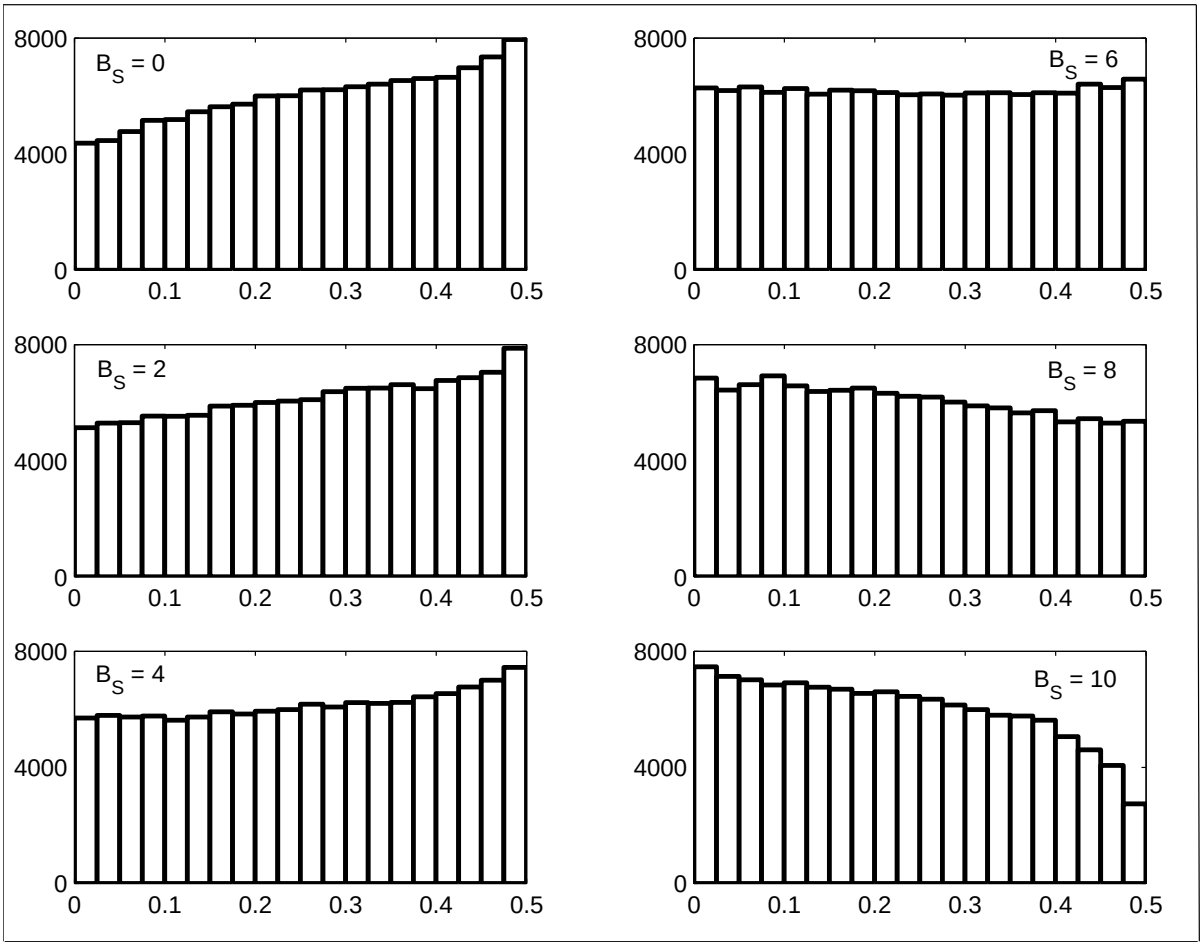

Figure 1: Histograms Conditioned on System States $R = 12$ and $B_s \in \{0, 2, 4, 6, 8, 10\}$

When we analyze the results of this series of simulation scenarios, we can conclude that conditioned on the system state $\big(R(t) = r_0, B_s(t) = b_0\big)$ at a random point in time, although total demand is a Poisson process, the unordered demand arrival times in $(t - L, t]$ are **not** necessarily r_0 independent random variables with uniform distribution on $(t - L, t]$, though for some cases the

distribution looks uniform.

Observe that for a moderate level of silver backorders, $B_s = 6$, the distribution of unordered arrival times appear uniform. However, for low values of B_s, there is a bias toward younger units in resupply (longer remaining delivery times) and for high values of B_s there is a bias toward older units in resupply (shorter remaining delivery times).

3 The Three Demand-Class Model

It is straightforward, but tedious, to extend the model to consider three demand classes, adding a *platinum* demand class to the previously described *gold* and *silver* demand classes. Let λ_p denote the arrival rate for *platinum* customers. *Platinum* customers are assumed to require a higher level of service than both *gold* or *silver* customers. We extend the rationing policy to include a threshold $S_p \leqslant S_g$ at and below which only *platinum* customers are served. The state space must be expanded to include gold backorders: (R, B_s, B_g), but in the case of exponentially distributed lead times it is not difficult to derive the balance equations which can be solved for the steady state probabilities. We omit the derivation in order to focus on the numerical accuracy of the resulting probabilities when the lead time distribution is other than exponential. The balance equations for three demand-class model are included in the next section.

An algorithm for solving the balance equations can be found in Vicil (2006). It is an extension of the Bridge algorithm described in Vicil and Jackson (2014).

3.1 Balance Equations for Three Customer Demand-Class

Under the setting with three customer demand classes, (R, B_s, B_g) is sufficient to characterize the system state. Let us denote the steady state probabilities as $\lim_{t\to\infty} P_{(0,0,0),(i,j,k)}(0,t) = \pi_{(i,j,k)}$.

Hence, for exponentially distributed lead times with rate μ, where $\mu = 1/T$, the balance equations for $S > S_g > S_p$ are as follows:

1. $i = 0$:

$$\pi_{(i,0,0)} \cdot \lambda \;=\; \pi_{(i+1,0,0)} \cdot \mu \cdot (i+1)$$

2. $1 \leqslant i < S - S_g$:

$$\pi_{(i,0,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,0,0)} \cdot \lambda + \pi_{(i+1,0,0)} \cdot \mu \cdot (i+1)$$

3. $i = S - S_g,\ j = 0$:

$$\pi_{(i,j,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,j,0)} \cdot \lambda + \pi_{(i+1,j,0)} \cdot \mu \cdot (i+1) + \pi_{(i+1,j+1,0)} \cdot \mu \cdot (i+1)$$

4. $i = S - S_g + j,\ j \geqslant 1$:

$$\pi_{(i,j,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,j-1,0)} \cdot \lambda_s + \big[\pi_{(i+1,j+1,0)} + \pi_{(i+1,j,0)}\big] \cdot \mu \cdot (i+1)$$

5. $S - S_g < i < S - S_p,\ j = 0$:

$$\pi_{(i,j,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,j,0)} \cdot \big(\lambda_g + \lambda_p\big) + \pi_{(i+1,j,0)} \cdot \mu \cdot (i+1)$$

6. $S - S_g < i - j < S - S_p,\ j \geqslant 1$:

$$\pi_{(i,j,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,j,0)} \cdot \big(\lambda_g + \lambda_p\big) + \pi_{(i+1,j,0)} \cdot \mu \cdot (i+1) + \pi_{(i-1,j-1,0)} \cdot \lambda_s$$

7. $i = S - S_p$:

$$\pi_{(i,0,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,0,0)} \cdot \big(\lambda_g + \lambda_p\big) + \big[\pi_{(i+1,0,0)} + \pi_{(i+1,1,0)} + \pi_{(i+1,0,1)}\big] \cdot \mu \cdot (i+1)$$

8. $i = S - S_p + j,\ j \geqslant 1$:

$$\pi_{(i,j,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,j-1,0)} \cdot \lambda_s + \pi_{(i-1,j,0)} \cdot \big(\lambda_g + \lambda_p\big) + \big[\pi_{(i+1,j,0)} + \pi_{(i+1,j,1)}\big] \cdot \mu \cdot (i+1)$$

9. $i = S - S_p + k,\ k \geqslant 1$:

$$\pi_{(i,0,k)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,0,k-1)} \cdot \lambda_g + \big[\pi_{(i+1,0,k+1)} + \pi_{(i+1,0,k)}\big] \cdot \mu \cdot (i+1)$$

10. $i = S - S_p + j + k,\ j \geqslant 1,\ k \geqslant 1$:

$$\pi_{(i,j,k)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,j-1,k)} \cdot \lambda_s + \pi_{(i-1,j,k-1)} \cdot \lambda_g + \big[\pi_{(i+1,j,k+1)} + \pi_{(i+1,j,k)}\big] \cdot \mu \cdot (i+1)$$

11. $i > S - S_p$:

$$\pi_{(i,0,0)} \cdot \big[\lambda + \mu \cdot i\big] = \pi_{(i-1,0,0)} \cdot \lambda_p + \pi_{(i+1,0,0)} \cdot \mu \cdot (i+1)$$

12. $i > S - S_p + k,\ k \geqslant 1$:

$$\pi_{(i,0,k)} \cdot \left[\lambda \ + \ \mu \cdot i\right] \ = \ \pi_{(i-1,0,k)} \cdot \lambda_p \ + \ \pi_{(i-1,0,k-1)} \cdot \lambda_g \ + \pi_{(i+1,0,k)} \cdot \mu \cdot (i+1)$$

13. $i > S - S_p + j,\ j \geqslant 1$:

$$\pi_{(i,j,0)} \cdot \left[\lambda \ + \ \mu \cdot i\right] \ = \ \pi_{(i-1,j-1,0)} \cdot \lambda_s \ + \ \pi_{(i-1,j,0)} \cdot \lambda_p \ + \ \pi_{(i+1,j,0)} \cdot \mu \cdot (i+1)$$

14. $i > S - S_p + j + k,\ j \geqslant 1,\ k \geqslant 1$:

$$\pi_{(i,j,k)} \cdot \left[\lambda \ + \ \mu \cdot i\right] \ = \ \pi_{(i-1,j-1,k)} \cdot \lambda_s \ + \ \pi_{(i-1,j,k-1)} \cdot \lambda_g \ + \ \pi_{(i-1,j,k)} \cdot \lambda_p \ + \ \pi_{(i+1,j,k)} \cdot \mu \cdot (i+1)$$

4 Performance Analysis Using Numerical Simulation

For the balance of the paper, we concentrate on using numerical simulation to evaluate the performance of the continuous time Markov chain (CTMC) approach under a variety of lead time probability distributions and for both two and three-demand class models. We explore a wide range of system parameters and, where possible, compare the results of the CTMC approach with competing heuristics. Unless otherwise stated, the duration of each simulation is 200,000 time periods and 10 independent simulations are performed for each parameter scenario. We use the mean of the *gold* fill rate, β_g, from each of the 10 simulations to construct confidence intervals around the performance metric. The confidence intervals are constructed according to the *t -distribution* because the sample size is small. In each scenario, the *silver* fill rate, β_s, can be determined analytically. Parameters for most scenarios are chosen so that $\beta_s \geqslant 60\%$, which is at least what we would anticipate in practice. When reporting the performance of competing heuristics, we use the confidence intervals from our simulations and we attempt to compute the *gold* fill rate using the alternative methodology. However, in the case of the embedded Markov chain approach for constant lead times, we use the *gold* fill rates reported in Fadiloglu and Bulut (2010) because the approach is non-trivial to implement. Our simulations confirm the high quality of the embedded Markov chain approach for constant lead times.

Our numerical study is divided into two major sections, one dealing with constant lead times and the other dealing with general lead time distributions.

4.1 The Performance of the Exponential Approach Under Constant Lead Times

4.1.1 A Comparison of the CTMC Approach with the Single-Cycle Approach

To compare the CTMC approach with the single-cycle approach of Dekker et al. (1998), we construct a series of experiments for which λ_s and λ_g values vary and we assume order lead times are constant. The parameters are chosen in such a way that $\beta_s \geqslant 60\,\%$ and $\beta_g \geqslant 85\,\%$, that is, *gold* customers contract for substantially higher service levels than *silver* customers.

In Table 1, thirty different cases are presented in order to compare the accuracy of approximations with respect to various system parameters. From these results, we conclude that several factors affect the performance of the Dekker et al. heuristic. First, it is clear that as long as the expected lead time demand is sufficiently low, the Dekker et al. heuristic provides a good approximation. However, as soon as the expected lead time demand exceeds some threshold (e.g. 15 units) in these experiments, we start observing significant deviations from the simulated fill-rate figures (case (19) through case (24) are good examples of this pattern). Second, it is also apparent that the accuracy of Dekker et al. heuristic improves for high *gold* fill-rates (i.e. 95%). Third, we also observe that beside *gold* fill-rates, *silver* fill-rates are also driving factors in the quality of Dekker et al. (1998) approximation. For example, cases (11) and (12) both correspond to high *gold* fill-rates, 98.84 % and 97.23 % respectively. However, the former has 82.17 % *silver* fill-rate while the latter has 65.32 %. Although both cases correspond to high *gold* fill-rates, the quality of Dekker et al. approximation is lower for the lower *silver* fill-rate (compare cases (17) and (18)).

On the other hand, it can be concluded that the *Independence Condition* holds well for these system parameters and the CTMC approach works well for all cases. It is important to note that, the CTMC approach provides a very high quality approximation across all the scenarios considered. The predicted *gold* fill-rate differs from the center of the confidence interval by no more than 1%. However, it is apparent that the CTMC approach consistently but slightly overestimates the simulated *gold* fill-rate, in contrast to the single-cycle approach which underestimates the *gold* fill-rate, often by a substantial amount.

Table 1: Comparison of the CTMC approximation to the single-cycle heuristic

Case	S	S_g	$\lambda_g/(\lambda_s+\lambda_g)$	λL	β_s	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	$\beta_{g\,(single-cycle)}$
(1)	5	2	1/2	1.5	80.88 %	99.53 ± 0.02 %	99.57 %	97.40 %
(2)	7	2	1/2	3	81.52 %	99.17 ± 0.03 %	99.23 %	98.78 %
(3)	10	2	1/2	6	74.40 %	97.90 ± 0.04 %	98.08 %	96.31 %
(4)	19	2	1/2	15	66.41 %	95.38 ± 0.07 %	95.80 %	89.76 %
(5)	29	3	1/2	24	63.19 %	97.78 ± 0.05 %	98.01 %	91.46 %
(6)	37	4	1/2	30	68.34 %	99.26 ± 0.03 %	99.35 %	95.50 %
(7)	5	1	1/3	2.25	80.94 %	97.41 ± 0.04 %	97.51 %	96.64 %
(8)	7	1	1/3	4.50	70.29 %	94.32 ± 0.08 %	94.63 %	91.62 %
(9)	13	2	1/3	9	70.60 %	98.60 ± 0.03 %	98.75 %	96.43 %
(10)	27	1	1/3	22.5	74.33 %	93.42 ± 0.13 %	93.59 %	87.29 %
(11)	44	2	1/3	36	82.17 %	98.84 ± 0.04 %	98.85 %	95.33 %
(12)	50	2	1/3	45	65.32 %	97.23 ± 0.08 %	97.37 %	87.20 %
(13)	6	2	2/3	2.25	80.94 %	98.79 ± 0.02 %	98.86 %	98.50 %
(14)	8	2	2/3	4.5	70.29 %	96.16 ± 0.06 %	96.44 %	94.65 %
(15)	13	2	2/3	9	70.60 %	94.50 ± 0.09 %	94.83 %	91.49 %
(16)	27	1	2/3	22.5	74.33 %	86.84 ± 0.21 %	87.10 %	82.63 %
(17)	44	2	2/3	36	82.17 %	95.36 ± 0.14 %	95.34 %	91.46 %
(18)	50	2	2/3	45	65.32 %	89.01 ± 0.16 %	89.37 %	79.25 %
(19)	8	2	1/5	3.75	82.29 %	99.86 ± 0.01 %	99.87 %	99.69 %
(20)	11	2	1/5	7.5	66.20 %	99.47 ± 0.03 %	99.51 %	98.29 %
(21)	19	2	1/5	15	66.41 %	99.26 ± 0.05 %	99.34 %	96.83 %
(22)	42	2	1/5	37.5	63.71 %	98.97 ± 0.05 %	99.04 %	92.86 %
(23)	65	2	1/5	60	63.38 %	98.89 ± 0.05 %	98.93 %	90.32 %

Table 1 - continued from previous page

Case	S	S_g	$\lambda_g/(\lambda_s+\lambda_g)$	λL	β_s	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	$\beta_{g\,(single-cycle)}$
(24)	81	2	1/5	75	66.28 %	98.94 ± 0.05 %	98.99 %	90.28 %
(25)	9	3	4/5	3.75	82.29 %	99.22 ± 0.03 %	99.30 %	99.05 %
(26)	12	2	4/5	7.5	77.64 %	94.98 ± 0.09 %	95.14 %	93.65 %
(27)	20	2	4/5	15	74.89 %	92.09 ± 0.11 %	92.31 %	89.40 %
(28)	43	2	4/5	37.5	69.52 %	87.07 ± 0.18 %	87.26 %	81.51 %
(29)	66	3	4/5	60	63.38 %	88.02 ± 0.31 %	88.43 %	79.06 %
(30)	82	3	4/5	75	66.28 %	88.59 ± 0.31 %	88.93 %	79.90 %

On Hand Probability Estimation: Besides fill-rate performance, we are also concerned with the quality of approximations of the stationary distributions for OH, the on hand inventory.

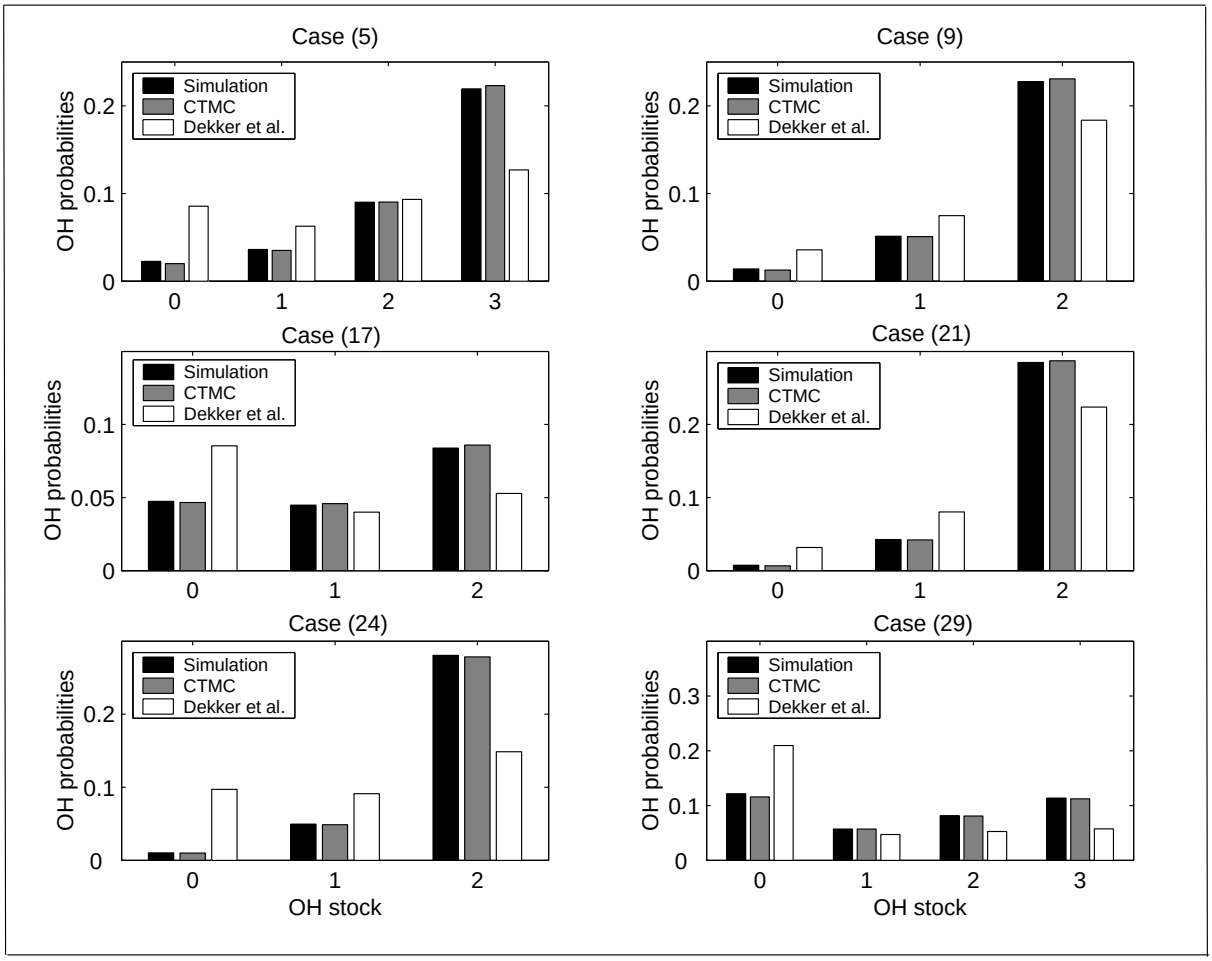

Figure 2: Comparison of CTMC approximation to the single-cycle (Dekker et al.) heuristic for OH probability distribution.

In Figure 2, we analyze the performance of approximations for the steady state OH probabilities for a subset of scenarios selected from Table 1, cases (5), (9), (17, (21), (24) and (29). Cases (5), (9) and (17) corresponds to *gold* fill-rates greater than 95 %, while cases (21) and (24) have fill-rates greater than 99 %. The horizontal axis in the graphs of Figure 2 measures OH. Since steady state OH probabilities for $OH > S_g$ can be calculated from *Palm's Theorem*, only probabilities for $0 \leqslant OH \leqslant S_g$ are presented. It is clear that the CTMC approximation performs much better than the single-cycle (Dekker et al.) heuristic for all scenarios. Also, for the cases considered, there are significant deviations from the simulated OH probabilities under the single-cycle approximation, even though simulated *gold* fill-rates are above 95% in most of the scenarios. In case (21), for example, even though the *gold* fill-rate is very high (99%) and single-cycle approach does not deviate greatly from the simulated fill-rate, there is a large deviation from the simulated OH probabilities. We can conclude that the CTMC approximation not only performs well with respect to gold fill-rate approximation but also performs quite well with respect to approximating OH probabilities, even for high expected total lead time demands (i.e. cases (17), (24) and (29)).

The Impact of Lead Time Demand Changes: In the next study, we set the ratio $\frac{\lambda_g}{\lambda_s+\lambda_g} = 0.5$ and select S in such a way that β_s is maintained at a high level of approximately 80 %. Our aim in this part is to analyze the effect of lead time (and hence expected lead time demand) on the performance of approximations, while trying to keep the *silver* customer service fixed. The results are presented in Table 2. We also provide absolute errors with respect to (mean) simulated *gold* fill-rates.

Table 2: Performance of approximations with respect to an increase in expected lead time demand

Case	S	S_g	λL	β_S	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(sgl-cycle)}$	AE$_{(sgl-cycle)}$
(I)	6	1	3	81.53 %	95.83 ± 0.08 %	95.95 %	0.12 %	94.79 %	1.04 %
(II)	10	1	6	84.72 %	95.81 ± 0.09 %	95.92 %	0.11 %	94.42 %	1.39 %
(III)	16	1	12	77.20 %	92.28 ± 0.10 %	92.41 %	0.13 %	88.58 %	3.70 %
(IV)	36	1	30	79.73 %	92.11 ± 0.06 %	92.25 %	0.14 %	87.40 %	4.71 %
(V)	56	1	48	82.68 %	93.06 ± 0.16 %	93.07 %	0.01 %	88.40 %	4.66 %
(VI)	68	1	60	80.12 %	91.67 ± 0.14 %	91.78 %	0.11 %	88.95 %	2.72 %

From these results, similar to the previous study, we can conclude that the performance of the CTMC approximation is much better than the single-cycle approach for all the cases considered. We can also observe that as the expected lead time demand increases, the single-cycle approach deviates more from the simulated fill rate. The absolute error of the CTMC approach increases slightly up to some point, and then starts to decrease. However, we can observe that, provided that β_s does not change much, an increase in expected lead time demand has only a small effect on the quality of the CTMC approximation. It is also important to note that, when all the cases in Table 1 and Table 2 are considered, the absolute error for the CTMC approximation is less than 0.5 %, while the absolute error for the single-cycle approach ranges from 1 % to 10 %.

The Impact of Absolute Demand Rate Changes: Next, we fix $\frac{\lambda_g}{\lambda_s+\lambda_g} = 0.5$, $S = 5$, and $S_g = 2$ and vary total work load λL. Our aim in this part is to analyze the effect of total work load on the performance of approximations, while keeping all other system parameters fixed. The results are presented in Table 3.

Table 3: Performance of approximations with respect to an increase in expected lead time demand

Case	S	S_g	λL	β_s	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(sgl-cycle)}$	AE$_{(sgl-cycle)}$
(I)	5	2	1.5	80.88 %	99.53 ± 0.02 %	99.57 %	0.04 %	99.40 %	0.13 %
(II)	5	2	3	42.41 %	95.42 ± 0.04 %	96.05 %	0.63 %	92.47 %	2.95 %
(III)	5	2	6	6.33 %	82.59 ± 0.13 %	85.92 %	3.33 %	55.94 %	26.65 %
(IV)	5	2	15	∼ 0 %	75.45 ± 0.23 %	78.93 %	3.48 %	2.44 %	73.01 %
(V)	5	2	24	∼ 0 %	75.12 ± 0.13 %	77.64 %	2.52 %	∼ 0 %	75.12 %
(VI)	5	2	30	∼ 0 %	75.26 ± 0.34 %	77.17 %	1.91 %	∼ 0 %	75.26 %

We can observe a similar pattern as in Table 2: the absolute error of approximation increases up to some point, and then starts to decrease. It is also interesting to observe that that as expected lead time increases, while rest of the system parameters are kept fixed, the *gold* customer fill-rate is not significantly affected after $\lambda L = 15$ in these experiments. This might be counter-intuitive. One explanation to this phenomenon is that for $\lambda L \geqslant 15$, *silver* customers do not get any service at all despite the existence of *silver* customer demands. On the other hand, all the replenishment orders due to silver customer demands are used to satisfy gold customers. Hence, this offsets the

negative effect of an increase in lead time demand on *gold* customer fill-rate. However, the degree of such an offset would vary depending on the ratio $\lambda_g/(\lambda_s + \lambda_g)$. We investigate the impact of that ratio next.

Varying the Demand Rate for Gold Service: In the following series of experiments, we set $S = 8$, $S_g = 2$, and $\lambda L = 5$. Our aim is to analyze the performance of approximations under a fixed workload while varying the ratio $\lambda_g/(\lambda_s + \lambda_g)$. The results are presented in Table 4. According to the numerical results, we see that CTMC approximation provides higher quality approximation in all cases than the single-cycle heuristic. We also observe that as the ratio $\lambda_g/(\lambda_s + \lambda_g)$ increases up to 2/3, the performance of both the CTMC approximation and the single-cycle heuristic are affected negatively. As the ratio increases beyond this point, the quality of both approximations increases. One explanation to this is that as the ratio approaches 0, the system behaves more like a single-customer system with silver demands, while as the ratio approaches 1, the system moves towards a single-customer system with gold demands. Hence, the effect of rationing decreases and therefore both approximations provide higher quality results at the extremes.

Table 4: Performance of approximations with respect to an increase in ratio $\lambda_g/(\lambda_s + \lambda_g)$ under fixed work load

Case	$\lambda_g/(\lambda_s + \lambda_g)$	β_s	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(sgl-cycle)}$	AE$_{(sgl-cycle)}$
(I)	1/10	61.60 %	99.86 ± 0.02 %	99.89%	0.03 %	99.60 %	0.26 %
(II)	1/5	61.60 %	99.47 ± 0.03 %	99.54%	0.07 %	98.61 %	0.86 %
(III)	1/3	61.60 %	98.54 ± 0.04 %	98.70 %	0.16 %	96.77%	1.77 %
(IV)	1/2	61.60 %	96.66 ± 0.07 %	97.02 %	0.36 %	94.14 %	2.52 %
(V)	2/3	61.60 %	94.10 ± 0.08 %	94.57 %	0.47 %	91.48 %	2.62 %
(VI)	4/5	61.60 %	91.50 ± 0.12 %	91.96 %	0.46 %	89.45 %	2.05 %
(VII)	9/10	61.60 %	89.21 ± 0.17 %	89.56%	0.35 %	88.01 %	1.20 %

Next, we analyze the performance of approximations for very low *gold* fill-rate levels. Although very low fill-rates should not be observed in real life applications, our main aim is to investigate the behavior of the system with respect to the *Independence Condition*. In this series of experiments, we set $S = 4$, $S_g = 2$, $L = 0.5$ and for the first set, we fix $\lambda_s = 1.5$ and then vary λ_g. The results

are presented in Table 5. In these cases, absolute errors for the CTMC approach are as high as 4.3 % but they are less than a third of the alternative approach. Even for very low *gold* fill-rates, our approximation is reasonably accurate, while the single-cycle approximation provides very poor performance. We also observe that as λ_g increases, the *gold* fill-rate decreases, and the absolute error for the CTMC approximation increases until certain point (i.e. $\lambda_g = 8$). Beyond this point, (cases (V) and (VI)), the absolute error begins to decrease as λ_g increases further. We can also observe a similar situation in Table 6 when we vary only λ_s.

Table 5: Performance of approximations for low *gold* fill-rates and varying *gold* demand rates

Case	S	S_g	λ_s	$\lambda_{\mathbf{g}}$	L	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(sgl-cycle)}$	AE$_{(sgl-cycle)}$
(I)	4	2	1.5	**2**	0.5	96.67 ± 0.07 %	97.11 %	0.44 %	95.60 %	1.07 %
(II)	4	2	1.5	**3**	0.5	91.61 ± 0.08 %	92.61 %	1.00 %	88.75 %	2.86 %
(III)	4	2	1.5	**4**	0.5	84.62 ± 0.15 %	86.38 %	1.76 %	79.29 %	5.33 %
(IV)	4	2	1.5	**8**	0.5	53.33 ± 0.13 %	57.73 %	4.40 %	36.91 %	16.42 %
(V)	4	2	1.5	**15**	0.5	27.21 ± 0.18 %	30.85 %	3.64 %	4.68 %	22.53 %
(VI)	4	2	1.5	**25**	0.5	16.55 ± 0.18 %	18.62 %	2.07 %	0.12 %	16.43 %

Varying the Demand Rate for Silver Service: We next fix $\lambda_g = 15$ and vary λ_s. The results are presented in Table 6, which are similar to the results in Table 5. For the cases considered, the absolute error of the CTMC approximation is less than 5.1 % which shows that the CTMC approach provides a reasonable approximation even for extreme cases, while the single-cycle heuristic performs poorly in these extreme settings.

Note in Table 6 that, as we increase λ_s while everything else is fixed, β_g increases. This is not what we might expect. This occurs because as λ_s increases, more units are ordered and when those units are received, they are used mainly for *gold* customers rather than *silver* customers due to the threshold rationing policy. *Gold* customers thus benefit from increased *silver* demand.

Table 6: Performance of approximations for low *gold* fill-rates and varying *silver* demand rates

Case	S	S_g	λ_s	$\lambda_{\mathbf{g}}$	L	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(Dekker\ et\ al.)}$	AE$_{(D...)}$
(I)	4	2	**2**	15	0.5	31.46 ± 0.17 %	35.56 %	4.10 %	4.36 %	27.10 %
(II)	4	2	**4**	15	0.5	44.20 ± 0.11 %	49.25 %	5.05 %	3.35 %	40.85 %
(III)	4	2	**8**	15	0.5	60.25 ± 0.20 %	65.16 %	4.91 %	2.24 %	58.01 %
(IV)	4	2	**15**	15	0.5	75.46 ± 0.11 %	78.93 %	3.47 %	1.44 %	74.02 %
(V)	4	2	**25**	15	0.5	85.96 ± 0.22 %	87.73 %	1.77 %	1.03 %	84.93 %
(VI)	4	2	**35**	15	0.5	90.97 ± 0.13 %	91.96 %	0.99 %	0.86 %	90.11 %

4.1.2 A Comparison of the Continuous Time Markov Chain Approach with the Embedded Markov Chain Approach

Finally, in this section, we compare the performance of CTMC approach with respect to the most recent approximation provided by Fadiloglu and Bulut (2010). Recall that the embedded Markov chain approach assumes the independence condition holds for constant lead times where as the continuous time Markov chain approach assumes the same condition holds for general lead time distributions. In the following series of experiments, we refer to the same numerical examples considered in Fadiloglu and Bulut (2010). The lead time is constant for each example. In the first series of experiments, the threshold levels are set as $S = 4$ and $S_g = 1$, while in the second series they are set as $S = 4$ and $S_g = 2$. The results are presented in Tables 7 and 8, respectively.

For both series of numerical studies, we can observe that the *independence condition* appears to hold as long as the *silver* fill-rate is not too low. For $\beta_s \geqslant 90\%$, we observe that the absolute error for the estimated gold fill-rate under CTMC approach is zero, while for $\beta_s \geqslant 73.58\%$, the absolute error is still less than 0.05%. On the other hand, for β_s as low as 19.91%, the absolute error increases up to 2.19%. The error is greatest when the *silver* fill-rate is lowest ($\beta_s = 1.74\%$).

Table 7: Comparison of CTMC approximation vs. Fadiloglu et. al. approximation, $S = 4, S_g = 1$.

λL	$\lambda_g/(\lambda_s + \lambda_g)$	β_s	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(Fadiloglu\ et\ al.)}$	AE$_{(F...)}$
	1/4		99.54 ± 0.01 %	99.54 %	0 %	99.5 %	0 %
1	1/2	91.97 %	99.07 ± 0.02 %	99.07 %	0 %	99.1 %	0 %
	3/4		98.59 ± 0.04 %	98.59 %	0 %	98.6 %	0 %
	1/4		91.13 ± 0.10 %	91.87 %	0.74 %	91.2 %	0.1 %
3	1/2	42.32 %	82.38 ± 0.13 %	83.47 %	1.09 %	82.4 %	0 %
	3/4		73.67 ± 0.10 %	74.59 %	0.92 %	73.7 %	0 %
	1/4		78.89 ± 0.09 %	80.90 %	2.01 %	78.7 %	0.2 %
6	1/2	6.20 %	58.05 ± 0.06 %	61.27 %	3.22 %	58.1 %	0 %
	3/4		37.60 ± 0.11 %	40.49 %	2.89 %	38.1 %	0.5 %

Table 8: Comparison of CTMC approximation vs. Fadiloglu et. al. approximation, $S = 4, S_g = 2$.

λL	$\lambda_g/(\lambda_s + \lambda_g)$	β_s	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$	AE $_{CTMC}$	$\beta_{g\,(Fadiloglu\ et\ al.)}$	AE$_{(F...)}$
	1/4		99.88 ± 0.01 %	99.89 %	0.01 %	99.9 %	0 %
1	1/2	73.58 %	99.52 ± 0.01 %	99.57 %	0.05 %	99.5 %	0 %
	3/4		98.95 ± 0.02 %	98.98 %	0.03 %	98.9 %	0.1 %
	1/4		97.80 ± 0.07 %	98.32 %	0.52 %	97.8 %	0 %
3	1/2	19.91 %	91.35 ± 0.09 %	92.93 %	1.58 %	91.3 %	0.1 %
	3/4		80.65 ± 0.09 %	82.84 %	2.19 %	80.7 %	0 %
	1/4		94.81 ± 0.09 %	96.16 %	1.35 %	94.7 %	0.1 %
6	1/2	1.74 %	79.85 ± 0.20 %	84.06 %	4.21 %	79.9 %	0 %
	3/4		56.20 ± 0.13 %	61.70 %	5.50 %	56.9 %	0.7 %

On the other hand, for those cases considered, the embedded Markov chain approach approximates the *gold* fill-rate extremely well, even when the *silver* fill-rate is small. It is surprising that it outperforms the CTMC approach given that the Independence Condition is the basis for both approaches. It is noteworthy that the differences are most pronounced in scenarios for which the Independence Condition is least likely to hold (see Section 2.1).

We conclude that as long as *silver* fill-rates are high (say, in excess of 60%), the CTMC approach will provide estimates of the *gold* fill-rate which are almost as good as the best heuristic for the constant lead time case.

4.1.3 The Performance of the Exponential Approach for Three Demand-Classes

In previous sections, under the setting with two priority demand-classes, we have shown the high quality of CTMC approximation, especially for sufficiently high *silver* fill-rates, which should be the case for most real life scenarios.

Next, we consider the extension of the model to three customer demand classes. To evaluate the performance of CTMC approximation, we conduct two series of numerical experiments. For both series, we set $S = 10, S_g = 2$, and $S_p = 1$. In the first one, we set $\lambda_s = \lambda_g$, and vary the ratio λ_p/λ among different work loads λL, where λ is the total demand arrival rate. The results are presented in Table 9. It can be observed that for a given work load, as the ratio λ_p/λ increases, the absolute error also increases up to some point, but then starts to decrease. On the other hand, for a given ratio λ_p/λ, as the work load increases, the absolute error also increases for both *gold* and *platinum* fill-rate approximations. Again we observe a similar pattern as in two priority demand classes setting: the CTMC approximation overestimates the true *gold* and *platinum* fill-rates. But for sufficiently high silver fill-rates (i.e. $\beta_s \geqslant 50\%$), the absolute errors for CTMC approximation with respect to (mean) simulated *gold* and *platinum* fill-rates are less than 0.75%. Even for silver fill-rates as low as 22%, the absolute errors are less than 4%. As noted earlier, such low service levels are unlikely in real life applications.

Table 9: Comparison of CTMC approximation vs. simulation, $S = 10, S_g = 2, S_p = 1$; $\lambda_s = \lambda_g$.

λL	λ_p/λ	β_s	$\beta_{g\,(Sim)}$	$\beta_{g\,(CTMC)}$	AE	$\beta_{p\,(Sim)}$	$\beta_{p\,(CTMC)}$	AE
	0.10		96.24 ± 0.06 %	96.35 %	0.11 %	99.82 % ± 0.02	99.84 %	0.02 %
	0.25		95.73 ± 0.05 %	95.84 %	0.11 %	99.49 % ± 0.02	99.54 %	0.05 %
5	0.50	86.66 %	94.85 ± 0.08 %	94.98 %	0.13 %	98.78 % ± 0.04	98.87 %	0.09 %
	0.75		94.19 ± 0.15 %	94.10 %	0.09 %	97.93 % ± 0.05	97.98 %	0.05 %
	0.90		93.41 ± 0.16 %	93.56 %	0.15 %	97.29 % ± 0.06	97.32 %	0.03 %
	0.10		81.45 ± 0.18 %	82.13 %	0.68 %	98.77 % ± 0.05	98.91 %	0.14 %
	0.25		78.88 ± 0.18 %	79.61 %	0.73 %	96.50 % ± 0.09	96.86 %	0.36 %
7.5	0.50	52.46 %	74.64 ± 0.16 %	75.34 %	0.70 %	91.62 % ± 0.09	92.30 %	0.68 %
	0.75		70.46 ± 0.26 %	70.91 %	0.45 %	85.47 % ± 0.16	86.09 %	0.62 %
	0.90		67.82 ± 0.34 %	68.13 %	0.31 %	80.89 % ± 0.08	81.36 %	0.47 %
	0.10		63.31 ± 0.20 %	65.05 %	1.74 %	97.03 % ± 0.09	97.42 %	0.39 %
	0.25		58.52 ± 0.20 %	60.13 %	1.61 %	91.48 % ± 0.10	92.59 %	1.11 %
10	0.50	22.02 %	50.01 ± 0.30 %	51.76 %	1.75 %	79.82 % ± 0.15	81.82 %	2.00 %
	0.75		41.92 ± 0.31 %	43.00 %	1.08 %	64.71 % ± 0.25	67.02 %	2.31 %
	0.90		36.90 ± 0.34 %	40.39 %	3.49 %	54.17 % ± 0.19	58.13 %	3.96 %

In the second series of experiments, we analyze the effect of the ratio λ_g/λ on the fill-rates as total work load varies. The results are presented in Table 10, where we can observe similar patterns as in the previous series.

Furthermore, it is interesting to observe the effect of two-level rationing on the *platinum* customer service levels. Although there is only 1 stock reserved for the use of platinum customers, we can see that the platinum fill-rate can be as much as 30% higher than the gold fill-rate (i.e. for the case $\lambda L = 10$, $\lambda_p/\lambda = 0.50$). We conclude from these experiments that two-step rationing provides even larger protection from being backordered for the highest priority demand class. Additionally, we can also observe that for sufficiently large silver fill-rates, the platinum fill-rates are quite high.

These experiments demonstrate that it is possible to provide distinctly different levels of service for multiple demand classes using threshold rationing policies.

Table 10: Comparison of CTMC approximation vs. simulation, $S = 10, S_g = 2, S_p = 1$; $\lambda_s = \lambda_p$.

λL	λ_g/λ	β_s	$\beta_{g\,(Sim)}$	$\beta_{g\,(CTMC)}$	AE	$\beta_{p\,(Sim)}$	$\beta_{p\,(CTMC)}$	AE
	0.10		96.14 ± 0.13 %	96.35 %	0.21%	99.20 % ± 0.02	99.27 %	0.07 %
	0.25		95.70 ± 0.04 %	95.84 %	0.14 %	99.25 % ± 0.03	99.30 %	0.05 %
5	0.50	86.66 %	94.88 ± 0.08 %	94.98 %	0.10 %	99.41 % ± 0.03	99.44 %	0.03 %
	0.75		94.06 ± 0.06 %	94.10 %	0.04 %	99.63 % ± 0.03	99.67 %	0.04 %
	0.90		93.55 ± 0.08 %	93.56 %	0.01 %	99.84 % ± 0.03	99.86 %	0.02 %
	0.10		81.39 ± 0.27 %	82.13 %	0.74 %	94.49 ± 0.09 %	95.04 %	0.55 %
	0.25		78.89 ± 0.21 %	79.61 %	0.72 %	94.77 ± 0.07 %	95.28 %	0.51 %
7.5	0.50	52.46 %	74.81 ± 0.22 %	75.34 %	0.53 %	95.82 ± 0.05 %	96.18 %	0.36 %
	0.75		70.55 ± 0.16 %	70.91 %	0.36 %	97.55 ± 0.06 %	97.74 %	0.19 %
	0.90		67.95 ± 0.18 %	68.13 %	0.18 %	98.97 ± 0.08 %	99.00 %	0.03 %
	0.10		63.26 ± 0.25 %	65.05 %	1.79 %	86.68 % ± 0.15	88.28 %	1.60 %
	0.25		58.44 ± 0.22 %	60.13 %	1.69 %	87.31 % ± 0.15	88.85 %	1.54 %
10	0.50	22.02 %	50.27 ± 0.15 %	51.76 %	1.49 %	89.94 % ± 0.15	90.98 %	1.04 %
	0.75		41.88 ± 0.18 %	43.00 %	1.12 %	94.08 % ± 0.12	94.63. %	0.55 %
	0.90		36.86 ± 0.23 %	40.39 %	3.53 %	97.41 % ± 0.16	99.31 %	1.90 %

4.2 The Performance of the CTMC Approach Under General Lead Times

4.2.1 The Performance of the CTMC Approach Under Erlang-Distributed Lead Times

The Erlang distribution can be used to model very different lead time behaviors. In this section, we perform simulation studies considering Erlang distributed lead times with several shape parameters $(20, 10, 5$ and $2)$. We consider the same cases as in Table 1. Mean lead times are set equal to the corresponding constant lead times in those cases. We investigate the performance of the CTMC approximation when lead times are Erlang distributed.

The results are presented in Table 11. We see that the CTMC approximation performs even better for Erlang distributed lead times than it does for constant lead times. Furthermore, even though there can be some irregularities due to the nature of the simulation study, we can also observe that the quality of CTMC approximation increases as the coefficient of variation of lead time approaches 1 (For the Erlang distribution, the coefficient of variation CV $= \sqrt{\frac{1}{k}}$, where k is

the shape parameter. Hence, the CV of the Erlang distribution varies between 0 and 1). This is expected since for $k = 1$, the Erlang distribution is identical to the exponential distribution, for which the CTMC approach is exact.

It is also important to point out that when Tables 11 is considered, the performance levels are all very close to each other as long as the expected lead times are equivalent. With respect to the quality of approximations, we observe similar relationships as in the constant lead time scenarios discussed in the previous section. Since the arguments would be similar, they are omitted here.

Since Erlang-20 will result in nearly constant lead times, it is not surprising to see the similarity of these results to Table 1. It can also be deduced from the results that as CV deviates from 1 towards 0, the CTMC approach tends to overestimate the gold fill-rate.

Table 11: Comparison of CTMC approximation to Erlang distributed lead times

		Simulations of Erlang lead times				
Case	λT	$\beta_{g(Erl-20)}$	$\beta_{g(Erl-10)}$	$\beta_{g(Erl-5)}$	$\beta_{g(Erl-2)}$	$\beta_{g(CTMC)}$
(1)	1.5	99.54 ± 0.02 %	99.55 ± 0.02 %	99.56 ± 0.01 %	99.56 ± 0.01 %	99.57 %
(2)	3	99.16 ± 0.03 %	99.16 ± 0.02 %	99.19 ± 0.03 %	99.17 ± 0.03 %	99.23 %
(3)	6	97.88 ± 0.03 %	97.90 ± 0.06 %	97.90 ± 0.05 %	97.98 ± 0.05 %	98.08 %
(4)	15	95.51 ± 0.08 %	95.45 ± 0.10 %	95.56 ± 0.12 %	95.59 ± 0.12 %	95.80 %
(5)	24	97.78 ± 0.08 %	97.79 ± 0.08 %	97.76 ± 0.06 %	97.86 ± 0.07 %	98.01 %
(6)	30	99.27 ± 0.04 %	99.27 ± 0.04 %	99.29 ± 0.03 %	99.28 ± 0.04 %	99.35 %
(7)	2.25	97.39 ± 0.05 %	97.39 ± 0.05 %	97.40 ± 0.04 %	97.44 ± 0.06 %	97.51%
(8)	4.5	94.34 ± 0.06 %	94.30 ± 0.05 %	94.33 ± 0.09 %	94.43 ± 0.10 %	94.63%
(9)	9	98.60 ± 0.06 %	98.61 ± 0.04 %	98.60 ± 0.07 %	98.62 ± 0.05 %	98.75%
(10)	22.5	93.40 ± 0.11 %	93.40 ± 0.12 %	93.44 ± 0.11 %	93.52 ± 0.06 %	93.59%
(11)	36	98.81 ± 0.05 %	98.80 ± 0.04 %	98.81 ± 0.05 %	98.76 ± 0.03 %	98.85%
(12)	45	97.26 ± 0.06 %	97.25 ± 0.07 %	97.23 ± 0.06 %	97.26 ± 0.06 %	97.37%
(13)	2.25	98.80 ± 0.05 %	98.81 ± 0.02 %	98.80 ± 0.04 %	98.81 ± 0.04 %	98.86 %
(14)	4.5	96.20 ± 0.06 %	96.19 ± 0.05 %	96.19 ± 0.05 %	96.30 ± 0.07 %	96.44 %
(15)	9	94.47 ± 0.08 %	94.49 ± 0.12 %	94.49 ± 0.10 %	94.62 ± 0.08 %	94.83 %
(16)	22.5	86.84 ± 0.23 %	86.85 ± 0.20 %	86.83 ± 0.19 %	86.84 ± 0.26 %	87.10 %
(17)	36	95.33 ± 0.15 %	95.37 ± 0.15 %	95.31 ± 0.14 %	95.24 ± 0.12 %	95.34 %

Table 11 - continued from previous page

		Simulations of Erlang lead times				
Case	λT	$\beta_{g(Erl-20)}$	$\beta_{g(Erl-10)}$	$\beta_{g(Erl-5)}$	$\beta_{g(Erl-2)}$	$\beta_{g(CTMC)}$
(18)	45	89.12 ± 0.24 %	89.25 ± 0.21 %	89.18 ± 0.25 %	89.21 ± 0.32 %	89.37 %
(19)	3.75	99.86 ± 0.01 %	99.86 ± 0.01 %	99.84 ± 0.01 %	99.86 ± 0.01 %	99.87%
(20)	7.5	99.44 ± 0.04 %	99.45 ± 0.03 %	99.46 ± 0.03 %	99.48 ± 0.02 %	99.51%
(21)	15	99.27 ± 0.03 %	99.27 ± 0.05 %	99.28 ± 0.03 %	99.29 ± 0.04 %	99.34%
(22)	37.5	99.00 ± 0.05 %	98.99 ± 0.05 %	99.03 ± 0.04 %	99.01 ± 0.06 %	99.04%
(23)	60	98.91 ± 0.09 %	98.90 ± 0.06 %	98.88 ± 0.07 %	98.89 ± 0.07 %	98.93%
(24)	75	98.99 ± 0.05 %	98.98 ± 0.04 %	98.94 ± 0.04 %	98.96 ± 0.05 %	98.99%
(25)	3.75	99.25 ± 0.03 %	99.25 ± 0.03 %	99.26 ± 0.02 %	99.26 ± 0.03 %	99.30%
(26)	7.5	94.85 ± 0.09 %	94.86 ± 0.06 %	94.85 ± 0.08 %	95.00 ± 0.10 %	95.14%
(27)	15	91.93 ± 0.14 %	92.00 ± 0.11 %	92.00 ± 0.14 %	92.08 ± 0.17 %	92.31%
(28)	37.5	87.00 ± 0.17 %	87.02± 0.17 %	86.98 ± 0.16 %	86.99 ± 0.20 %	87.26%
(29)	60	87.95 ± 0.27 %	87.98 ± 0.26 %	88.08 ± 0.23 %	87.96 ± 0.26 %	88.43%
(30)	75	88.52 ± 0.19 %	88.49 ± 0.20 %	88.49 ± 0.22 %	88.66 ± 0.38 %	88.93%

The Impact of Lead Time Demand Changes: In the next study, we set $\frac{\lambda_g}{\lambda_s+\lambda_g} = 0.5$, $S = 5$, and $S_g = 2$. First we vary total work load λT. Our aim in this part is to analyze the effect of total work load on the performance of approximations under Erlang distributed lead times with varying coefficient of variations. The results are presented in Table 12.

Table 12: Performance of Erlang lead time approximations with respect to an increase in expected lead time demand λT

			Simulations of Erlang lead times				
Case	λT	β_s	$\beta_{g(Erl-20)}$	$\beta_{g(Erl-10)}$	$\beta_{g(Erl-5)}$	$\beta_{g(Erl-2)}$	$\beta_{g(CTMC)}$
(I)	1.5	80.88 %	99.54 ± 0.02 %	99.55 ± 0.02 %	99.56 ± 0.01 %	99.56 ± 0.01 %	99.57 %
(II)	3	42.41 %	95.40 ± 0.07 %	95.48 ± 0.05 %	95.54 ± 0.04 %	95.70 ± 0.07 %	96.05 %
(III)	6	6.33 %	82.64 ± 0.10 %	82.76 ± 0.13 %	83.08 ± 0.10 %	84.30 ± 0.11 %	85.92 %
(IV)	15	∼ 0 %	75.64 ± 0.18 %	75.71 ± 0.25 %	75.99 ± 0.21 %	76.86 ± 0.16 %	78.93 %
(V)	24	∼ 0 %	75.19 ± 0.15 %	75.20 ± 0.21 %	75.42 ± 0.14 %	76.11 ± 0.13 %	77.64 %
(VI)	30	∼ 0 %	75.13 ± 0.12 %	74.99 ± 0.23 %	75.06 ± 0.17 %	75.62 ± 0.21 %	77.17 %

We observe that the quality of the approximation deteriorates as the silver fill-rate decreases.

The Impact of Relative Demand Rate Changes: Next, we set $S = 8$, $S_g = 2$, and $\lambda T = 5$ and vary the ratio $\lambda_g/(\lambda_s+\lambda_g)$. Our aim in this part is to analyze the effect of the ratio $\lambda_g/(\lambda_s+\lambda_g)$ on the quality of CTMC approximation. The results are presented in Table 13. For all those cases, $\beta_s = 61.60\%$, and the quality of approximation is high.

Table 13: Performance of Erlang lead time approximations with respect to a change in ratio $\frac{\lambda_g}{\lambda_s+\lambda_g}$

		Simulations of Erlang lead times				
Case	$\lambda_g/(\lambda_s+\lambda_g)$	$\beta_{g(Erl-20)}$	$\beta_{g(Erl-10)}$	$\beta_{g(Erl-5)}$	$\beta_{g(Erl-2)}$	$\beta_{g(CTMC)}$
(I)	1/10	99.88± 0.02 %	99.86 ± 0.02 %	99.87 ± 0.01 %	99.87 ± 0.02 %	99.89 %
(II)	1/5	99.47 ± 0.04 %	99.47 ± 0.04 %	99.47 ± 0.03 %	99.51 ± 0.03 %	99.54 %
(III)	1/3	98.49 ± 0.03 %	98.51 ± 0.03 %	98.52 ± 0.04 %	98.60 ± 0.04 %	98.70 %
(IV)	1/2	96.73 ± 0.04 %	96.73 ± 0.08 %	96.72 ± 0.07 %	96.86 ± 0.07 %	97.02 %
(V)	2/3	94.05 ± 0.06 %	94.07 ± 0.09 %	94.09 ± 0.08 %	94.21 ± 0.10 %	94.57 %
(VI)	4/5	91.49 ± 0.12 %	91.50 ± 0.11 %	91.48 ± 0.12 %	91.65 ± 0.13 %	91.96 %
(VII)	9/10	89.24 ± 0.08 %	89.26 ± 0.11 %	89.27 ± 0.11 %	89.42 ± 0.12 %	89.56 %

From the results of both Table 12 and Table 13, we can observe that the CTMC approximation shows modal behavior as λT or the ratio $\lambda_g/(\lambda_s + \lambda_g)$ increases. The pattern is similar to the constant lead time cases studied previously: the absolute error increases up to a point, and then starts to decrease.

4.2.2 The Performance of the CTMC Approach Under Gamma-Distributed Lead Times

Although the Erlang distribution is widely used to model systems in inventory management, it may be important to study systems with lead times that are highly variable. In this part of the study, we consider Gamma distributed lead times with CV values of 1.25, 1.50, 2.00 and 3.00.

Again we perform a simulation study with the same system parameters as in Table 1. The results are presented in Table 14. It can be observed from the results that for given λ_s and λ_g, as the CV greatly exceeds 1, the quality of approximation starts to diminish and the approximation

underestimates true gold fill-rates. This suggests that the dependence of the probability distribution of future unit deliveries on the number of silver backorders gets stronger.

Nevertheless, the CTMC approximation still provides satisfactory results for most of the settings. However, it can be seen from the results that as the expected lead time demand increases (for fixed S), the CTMC approximation deviates more from the simulated fill-rate values. In addition, we also observe that as the ratio $\lambda_g/\lambda_s + \lambda_g$ increases, the quality of CTMC approximation diminishes (case (18) versus case (12)). It is also important to note that for the cases considered in Table 14 for which the CV is less than or equal to 2, the absolute error for the CTMC approximation is less than 2.40%.

Table 14: Comparison of CTMC approximation to Gamma distributed lead times

		Simulations of Gamma lead times				
Case	λT	$\beta_{g(CV=3.00)}$	$\beta_{g(CV=2.00)}$	$\beta_{g(CV=1.50)}$	$\beta_{g(CV=1.25)}$	$\beta_{g(CTMC)}$
(1)	1.5	99.77 ± 0.02 %	99.69 ± 0.01 %	99.63 ± 0.02 %	99.59 ± 0.02 %	99.57 %
(2)	3	99.64 ± 0.03 %	99.44 ± 0.02 %	99.32 ± 0.03 %	99.30 ± 0.02 %	99.23 %
(3)	6	99.15 ± 0.02 %	98.68 ± 0.05 %	98.35 ± 0.05 %	98.26 ± 0.04 %	98.08 %
(4)	15	98.24 ± 0.04 %	97.12 ± 0.08 %	96.41 ± 0.14 %	96.06 ± 0.09 %	95.80 %
(5)	24	99.48 ± 0.03 %	98.91 ± 0.04 %	98.43 ± 0.05 %	98.15 ± 0.05 %	98.01 %
(6)	30	99.90 ± 0.01 %	99.70 ± 0.02 %	99.53 ± 0.03 %	99.45 ± 0.03 %	99.35 %
(7)	2.25	98.37 ± 0.04 %	97.95 ± 0.03 %	97.70 ± 0.05 %	97.59 ± 0.04 %	97.51%
(8)	4.5	96.76 ± 0.11 %	95.74 ± 0.10 %	95.11 ± 0.06 %	94.83 ± 0.09 %	94.63%
(9)	9	99.56 ± 0.02 %	99.22 ± 0.04 %	98.99 ± 0.03 %	98.87 ± 0.04 %	98.75%
(10)	22.5	95.83 ± 0.18 %	94.70 ± 0.10 %	93.97 ± 0.17 %	93.73 ± 0.13 %	93.59%
(11)	36	99.58 ± 0.05 %	99.20 ± 0.06 %	98.99 ± 0.04 %	98.93 ± 0.06 %	98.85%
(12)	45	99.05 ± 0.05 %	98.28 ± 0.09 %	97.73 ± 0.07 %	97.54 ± 0.09 %	97.37%
(13)	2.25	99.32 ± 0.02 %	99.06 ± 0.02 %	98.96 ± 0.03 %	98.90 ± 0.04 %	98.86 %
(14)	4.5	98.17 ± 0.08 %	97.36 ± 0.05 %	96.90 ± 0.06 %	96.64 ± 0.07 %	96.44 %
(15)	9	97.26 ± 0.08 %	96.10 ± 0.12 %	95.39 ± 0.08 %	95.14 ± 0.11 %	94.83 %
(16)	22.5	90.01 ± 0.42 %	88.35 ± 0.23 %	87.47 ± 0.25 %	87.19 ± 0.23 %	87.10 %
(17)	36	97.48 ± 0.19 %	96.20 ± 0.22 %	95.71 ± 0.08 %	95.57 ± 0.13 %	95.34 %
(18)	45	94.42 ± 0.24 %	91.74 ± 0.49 %	90.36 ± 0.17 %	89.88 ± 0.33 %	89.37 %
(19)	3.75	99.95 ± 0.01 %	99.91 ± 0.01 %	99.89 ± 0.01 %	99.88 ± 0.01 %	99.87%

Table 14 - continued from previous page

		Simulations of Gamma lead times				
Case	λT	$\beta_{g(CV=3.00)}$	$\beta_{g(CV=2.00)}$	$\beta_{g(CV=1.50)}$	$\beta_{g(CV=1.25)}$	$\beta_{g(CTMC)}$
(20)	7.5	99.87 ± 0.02 %	99.72 ± 0.02 %	99.60 ± 0.03 %	99.56 ± 0.03 %	99.51%
(21)	15	99.79 ± 0.01 %	99.61 ± 0.02 %	99.50 ± 0.03 %	99.42 ± 0.04 %	99.34%
(22)	37.5	99.68 ± 0.04 %	99.38 ± 0.04 %	99.20 ± 0.04 %	99.09 ± 0.06 %	99.04%
(23)	60	99.66 ± 0.03 %	99.32 ± 0.06 %	99.13 ± 0.05 %	99.01 ± 0.05 %	98.93%
(24)	75	99.66 ± 0.03 %	99.34 ± 0.04 %	99.11 ± 0.07 %	99.04 ± 0.03 %	98.99%
(25)	3.75	99.62 ± 0.02 %	99.46 ± 0.03 %	99.37 ± 0.04 %	99.32 ± 0.03 %	99.30%
(26)	7.5	96.84 ± 0.11 %	95.96 ± 0.08 %	95.47 ± 0.09 %	95.24 ± 0.09 %	95.14%
(27)	15	95.18 ± 0.25 %	93.70 ± 0.16 %	92.93 ± 0.18 %	92.63 ± 0.15 %	92.31%
(28)	37.5	91.84 ± 0.32 %	89.31 ± 0.39 %	88.20 ± 0.32 %	87.69 ± 0.13 %	87.26%
(29)	60	94.17 ± 0.27 %	91.20 ± 0.28 %	89.58 ± 0.27 %	88.79 ± 0.26 %	88.43%
(30)	75	94.05 ± 0.50 %	91.43 ± 0.25 %	89.94 ± 0.32 %	89.28 ± 0.41 %	88.93%

In the next two studies, as we did for Erlang distributed lead times, first we set $\frac{\lambda_g}{\lambda_s+\lambda_g} = 0.5$, $S = 5$, and $S_g = 2$ and vary total work load λT. Second, we set $S = 8$, $S_g = 2$, and $\lambda T = 5$ and vary the ratio $\lambda_g/(\lambda_s + \lambda_g)$. The results are presented in Table 15 and in Table 16. We can observe similar patterns as in the previous cases where lead times are Erlang distributed.

Table 15: Performance of Gamma lead time approximations with respect to an increase in expected lead time demand

			Simulations of Gamma lead times				
Case	λT	β_s	$\beta_{g(CV=3.00)}$	$\beta_{g(CV=2.00)}$	$\beta_{g(CV=1.50)}$	$\beta_{g(CV=1.25)}$	$\beta_{g(CTMC)}$
(I)	1.5	80.88 %	99.78 ± 0.02 %	99.68 ± 0.02 %	99.63 ± 0.02 %	99.60 ± 0.02 %	99.57 %
(II)	3	42.41 %	98.45 ± 0.03 %	97.41 ± 0.05 %	96.71 ± 0.06 %	96.40 ± 0.07 %	96.05 %
(III)	6	6.33 %	95.59 ± 0.04 %	91.93 ± 0.11 %	89.08 ± 0.11 %	87.49 ± 0.10 %	85.92 %
(IV)	15	∼ 0 %	93.21 ± 0.05 %	87.50 ± 0.14 %	83.34 ± 0.14 %	81.10 ± 0.19 %	78.93 %
(V)	24	∼ 0 %	92.33 ± 0.08 %	86.12 ± 0.12 %	81.80 ± 0.10 %	79.42 ± 0.15 %	77.64 %
(VI)	30	∼ 0 %	91.82 ± 0.12 %	85.45 ± 0.14 %	81.08 ± 0.16 %	78.86 ± 0.21 %	77.17 %

Table 16: Performance of Gamma lead time approximations with respect to a change in ratio $\frac{\lambda_g}{\lambda_s+\lambda_g}$

		Simulations of Gamma lead times				
Case	$\lambda_g/(\lambda_s+\lambda_g)$	$\beta_{g(CV=3.00)}$	$\beta_{g(CV=2.00)}$	$\beta_{g(CV=1.50)}$	$\beta_{g(CV=1.25)}$	$\beta_{g(CTMC)}$
(I)	1/10	99.98± 0.01 %	99.94 ± 0.02 %	99.92 ± 0.02 %	99.91 ± 0.02 %	99.89 %
(II)	1/5	99.86 ± 0.01 %	99.72 ± 0.03 %	99.64 ± 0.03 %	99.60 ± 0.02 %	99.54 %
(III)	1/3	99.54 ± 0.03 %	99.23 ± 0.03 %	98.97 ± 0.03 %	98.83 ± 0.03 %	98.70 %
(IV)	1/2	98.78 ± 0.04 %	97.97 ± 0.05 %	97.48 ± 0.07 %	97.22 ± 0.07 %	97.02 %
(V)	2/3	97.25 ± 0.07 %	95.96 ± 0.08 %	95.27 ± 0.04 %	94.94 ± 0.05 %	94.57 %
(VI)	4/5	94.95 ± 0.15 %	93.41 ± 0.06 %	92.62 ± 0.15 %	92.24 ± 0.11 %	91.96 %
(VII)	9/10	92.09 ± 0.21 %	90.66 ± 0.15 %	90.06 ± 0.14 %	89.81 ± 0.11 %	89.56 %

4.2.3 The Performance of the CTMC Approach Under Lognormal-Distributed Lead Times

In this part, we study the systems with lead times such that the distribution is believed to be skewed and continuous and different than gamma distribution. Therefore we consider Lognormal distributed lead times with CV values of 0.5, 1.5, 2 and 3. We set $S = 8$, $S_g = 2$, and $\lambda T = 5$ and vary the ratio $\lambda_g/(\lambda_s + \lambda_g)$. The results are presented in Table 17. We observe that for CV=0.5, CTMC approximation overestimates the simulated gold fill rates while for CV > 1, CTMC approximation underestimates. However, the quality of approximation is quite high in almost all cases. Even for CV as high as 3, we observe that the maximum absolute error is less than 0.75%.

Table 17: Performance of Lognormal lead time approximations with respect to a change in ratio $\frac{\lambda_g}{\lambda_s+\lambda_g}$

		Simulations of Lognormal lead times				
Case	$\lambda_g/(\lambda_s+\lambda_g)$	$\beta_{g(CV=3.00)}$	$\beta_{g(CV=2.00)}$	$\beta_{g(CV=1.50)}$	$\beta_{g(CV=0.5)}$	$\beta_{g(CTMC)}$
(I)	1/10	99.91± 0.02 %	99.89 ± 0.02 %	99.88 ± 0.03 %	99.87 ± 0.02 %	99.89 %
(II)	1/5	99.60 ± 0.02 %	99.58 ± 0.02 %	99.54 ± 0.02 %	99.48 ± 0.04 %	99.54 %
(III)	1/3	98.95 ± 0.04 %	98.85 ± 0.03 %	98.69 ± 0.04 %	98.53 ± 0.02 %	98.70 %
(IV)	1/2	97.53 ± 0.06 %	97.23 ± 0.06 %	97.05 ± 0.06 %	96.73 ± 0.06 %	97.02 %
(V)	2/3	95.29 ± 0.12 %	94.79 ± 0.08 %	94.57 ± 0.06 %	94.13 ± 0.09 %	94.57 %
(VI)	4/5	92.69 ± 0.09 %	92.21 ± 0.16 %	92.05 ± 0.13 %	91.56 ± 0.09 %	91.96 %
(VII)	9/10	89.96 ± 0.15 %	89.64 ± 0.17 %	89.43 ± 0.17 %	89.92 ± 0.09 %	89.56 %

4.2.4 The Performance of the CTMC Approach Under Geometric Distributed Lead Times

In this part, we study the systems with lead times having geometric distribution. We set $S = 8$, $S_g = 2$, and $\lambda T = 5$ and vary the ratio $\lambda_g/(\lambda_s + \lambda_g)$. The results are presented in Table 18. We observe that the quality of approximation is quite high in almost all cases. Even for CV as high as 3, we observe that the maximum absolute error is less than 0.75%.

Table 18: Performance of Geometric lead time approximations with respect to a change in ratio $\frac{\lambda_g}{\lambda_s+\lambda_g}$

Case	$\lambda_g/(\lambda_s+\lambda_g)$	$\beta_{g\,(Simulation)}$	$\beta_{g\,(CTMC)}$
(I)	1/10	99.87 ± 0.03 %	99.89%
(II)	1/5	99.49 ± 0.03 %	99.54%
(III)	1/3	98.61 ± 0.03 %	98.70 %
(IV)	1/2	96.90 ± 0.03 %	97.02 %
(V)	2/3	94.36 ± 0.03 %	94.57 %
(VI)	4/5	91.74 ± 0.11 %	91.96 %
(VII)	9/10	89.41 ± 0.14 %	89.56%

4.2.5 A Comparison of the Continuous Time Markov Chain Approach with the Embedded Markov Chain Approach

Although the approximation given by Fadiloglu and Bulut (2010) applies to constant lead time cases, as we have shown theoretically by Theorem 2.1 and emprically by exhaustive numerical examples, as long as the expected lead time is kept fixed, the *gold* service levels are not expected to vary much and any valid approximation might be used to approximate general lead time cases as well. Therefore, by using this idea, in this part of the the study we compare the performance of CTMC approach with respect to the most recent approximation provided by Fadiloglu and Bulut (2010) under Lognormal and Geometric lead time distributions. In the following series of experiments, we refer to the same numerical examples considered in Fadiloglu and Bulut (2010). The expected lead time is constant for each example.

Table 19: Comparison of CTMC approximation vs. Fadiloglu et. al. approximation, $S = 4, S_g = 1$.

				Erlang				
λT	λ_g/λ	β_s	$\beta_{g\,(constant)}$	$\beta_{g(CV=0.25)}$	$\beta_{g(CV=0.50)}$	$\beta_{g(CV=0.707)}$	$\beta_{g\,(CTMC)}$	$\beta_{g\,(Fadiloglu\ et\ al.)}$
1	1/4	91.97 %	99.54 ± 0.01 %	99.54 ± 0.02 %	99.53 ± 0.02 %	99.51 ± 0.03 %	99.54 %	99.5 %
	1/2		99.07 ± 0.02 %	99.06 ± 0.02 %	99.04 ± 0.03 %	99.07 ± 0.04 %	99.07 %	99.1 %
	3/4		98.59 ± 0.04 %	98.58 ± 0.02 %	98.59 ± 0.02 %	98.57 ± 0.02 %	98.59 %	98.6 %
3	1/4	42.32 %	91.13 ± 0.10 %	91.15 ± 0.13 %	91.30 ± 0.12 %	91.48 ± 0.09 %	91.87 %	91.2 %
	1/2		82.38 ± 0.13 %	82.38 ± 0.06 %	82.69 ± 0.18 %	82.84 ± 0.12 %	83.47 %	82.4 %
	3/4		73.67 ± 0.10 %	73.62 ± 0.11 %	73.74 ± 0.14 %	74.04 ± 0.14 %	74.59 %	73.7 %
6	1/4	6.20 %	78.89 ± 0.09 %	78.93 ± 0.17 %	79.19 ± 0.12 %	79.74 ± 0.21 %	80.90 %	78.7 %
	1/2		58.05 ± 0.06 %	57.98 ± 0.08 %	58.62 ± 0.16 %	59.43 ± 0.15 %	61.27 %	58.1 %
	3/4		37.60 ± 0.11 %	37.70 ± 0.091 %	38.00 ± 0.14 %	38.89 ± 0.19 %	40.49 %	38.1 %

In Table 19, we report simulation studies which consider a constant lead time and Erlang distributed lead times with shape parameters $16, 4$, and 2. (For the Erlang distribution, the coefficient of variation $\text{CV} = \sqrt{\frac{1}{k}}$, where k is the shape parameter. Hence, the CV of the Erlang distribution varies between 0 and 1 for $k \geqslant 1$). For the constant lead time case, we observe that the *independence condition* appears to hold as long as the *silver* fill-rate is not too low. In particular, for $\beta_s \geqslant 90\%$, we observe that the absolute error for the estimated gold fill-rate under CTMC approach is zero, while for $\beta_s \geqslant 42.32\%$, the absolute error is still less than 1.15%. On the other hand, for β_s

as low as 6.20%, the absolute error increases up to 3.22%. However, for the cases considered, the embedded Markov chain approach approximates the *gold* fill-rate extremely well, even when the *silver* fill-rate is small.

On the other hand, for Erlang distributed lead times, it is interesting to observe that as CV increases, the quality of CTMC approach increases while the quality of embedded Markov chain approach diminishes. For the cases with CV=0.707 and $\beta_s = 42.32\%$, the maximum absolute error for the CTMC approach drops to 0.63% and for β_s as low as 6.20% the maximum absolute error drops to 1.84%. On the other hand, the maximum absolute error for the embedded Markov chain approach can be as high as 1.33%.

These results drive our motivation to study other cases to observe how the quality of approximation changes as coefficient of variation increases. To do so, we use the same setting as before but this time with Lognormal and geometric lead time distributions. We study the cases with CV equal to $1.50, 2.00$ and 3.00. The results are presented in Table 20. For all the cases with Lognormal lead time distributions, the CTMC approach is superior to the embedded Markov chain approach. Furthermore, we also see that embedded Markov chain approach underestimates the simulated *gold* fill-rates. However, this situation varies for CTMC approach depending on CV values and other system parameters. For $CV \leqslant 2.00$ and β_s as low as 42.32%, the maximum absolute error for CTMC approach is 0.7% while the error can be as high as 1.6% for the embedded Markov chain approach. For the geometric lead time distribution cases, for $\beta_s = 91.97\%$, both methods provide excellent approximations. For the cases with $\beta_s = 42.32\%$, both methods tie in terms of approximation performance. On the other hand, for β_s as low as 6.20%, the CTMC approach provides better approximations than the embedded Markov chain approach. It is surprising that as the CV increases, the CTMC approach outperforms the embedded Markov chain approach given that the Independence Condition is the basis for both approaches. It is noteworthy that the differences are most pronounced in scenarios for which the Independence Condition is least likely to hold.

Table 20: Comparison of CTMC approximation vs. Fadiloglu et. al. approximation, $S = 4, S_g = 1$.

			Lognormal					
λT	λ_g/λ	β_s	$\beta_{g(CV=3.00)}$	$\beta_{g(CV=2.00)}$	$\beta_{g(CV=1.50)}$	$\beta_{g\,(geometric)}$	$\beta_{g\,(CTMC)}$	$\beta_{g\,(Fadiloglu\ et\ al.)}$
1	1/4	91.97 %	99.58 ± 0.03 %	99.55 ± 0.02 %	99.55 ± 0.02 %	99.52 ± 0.02 %	99.54 %	99.5 %
	1/2		99.12 ± 0.02 %	99.07 ± 0.03 %	99.06 ± 0.02 %	99.06 ± 0.03 %	99.07 %	99.1 %
	3/4		98.66 ± 0.03 %	98.62 ± 0.04 %	98.57 ± 0.02 %	98.58 ± 0.03 %	98.59 %	98.6 %
3	1/4	42.32 %	92.87 ± 0.06 %	92.30 ± 0.09 %	91.79 ± 0.13 %	91.39 ± 0.12 %	91.87 %	91.2 %
	1/2		85.06 ± 0.10 %	84.08 ± 0.14 %	83.23 ± 0.16 %	82.82 ± 0.13 %	83.47 %	82.4 %
	3/4		76.34 ± 0.20 %	75.29 ± 0.21 %	74.50 ± 0.11 %	74.11 ± 0.22 %	74.59 %	73.7 %
6	1/4	6.20 %	83.25 ± 0.13 %	81.75 ± 0.12 %	80.40 ± 0.12 %	80.02 ± 0.11 %	80.90 %	78.7 %
	1/2		65.07 ± 0.18 %	62.73 ± 0.16 %	60.62 ± 0.16 %	60.08 ± 0.15 %	61.27 %	58.1 %
	3/4		44.72 ± 0.30 %	42.21 ± 0.21 %	40.08 ± 0.11 %	39.49 ± 0.22 %	40.49 %	38.1 %

5 Conclusion

In this paper, we consider a model in which there are two priority demand classes exhibiting mutually independent, stationary, Poisson demand processes with non-zero order lead times that are independent and identically distributed. We assume an (S-1,S) ordering policy and a threshold level-based allocation and backorder clearing policy are followed.

There is no exact solution as yet for this rationing policy in the literature, except for the special case of exponentially distributed lead times. We pinpoint the difficulty for exact steady state analysis, and then show why a continuous time Markov chain approach might provide a good approximation to the calculation of stationary probabilities under general lead time distributions. We are the first to provide approximations for general lead time distributions. We first compare our results with known heuristics for the constant lead time case. We demonstrate that for constant lead time, the resulting solution outperforms the single-cycle approach of Dekker et al. (1998). According to the simulation study, for realistic scenarios as considered in Table 1 and Table 2, the absolute error for the CTMC approximation is less than 0.5 %, while the absolute error for the Dekker et al. heuristic can be as high as 10 %.

Even for unrealistic scenarios where *gold* customers receive very low service, almost for all cases we observe that ,the absolute error for the CTMC approximation is less than 5 % which shows that

CTMC approximation provides rough approximations even for extreme cases, while the single-cycle approach performs poorly (for constant lead time) for such cases.

Furthermore, we also compare the quality of approximations with respect to approximating steady state OH probabilities. We show that there can be significant deviations from the simulated OH probabilities under the single-cycle approach, although simulated *gold* fill-rates are above 95 %, while CTMC approximation provides high quality approximations.

We then compare the performance of CTMC approximation with respect to the most recent approximation provided by Fadiloglu and Bulut (2010). Although they use the same assumption as ours, their method is customized to the constant lead time case. For the numerical examples considered, their method is clearly superior. However, for $\beta_s \geqslant 73.58\%$, both approximations yield very close results.

We also apply CTMC approximation to a three priority demand-class setting and show the quality of approximation in this setting. We are the first in literature to provide such approximation for more than two demand classes.

We then compare the performance of our approach with respect to general lead time distributions. To do so, we use both Erlang and gamma distributions. We can see that as the CV deviates from 1, the dependence of the probability distribution of future unit deliveries on the number of silver backorders gets stronger. Hence, the quality of approximation starts to diminish (as expected). For CV higher than 1, the approximation underestimates true gold fill-rates, while for CV lower than 1, the approximation overestimates true gold fill-rates. On the other hand, for the cases considered in this paper, we can see that CTMC approximation still provides satisfactory results for most of the settings under general lead time distributions.

For practical applications, it is important to provide simple and accurate approximations, and to investigate their behavior under different system settings. Therefore, our proposed method, which requires only the knowledge of the mean value of the lead time distributions, performs well over a wide range of parameter settings for general lead time distributions, provided the silver fill-rate is maintained in excess of 60 %.

As a suggestion for future research, since the CTMC approximation provides quite satisfactory results under a static rationing policy for general lead time distributions, it may be interesting to

explore the performance of this approach under dynamic replenishment policies.

References

[1] Dekker, R., M. J. Kleijn, P. J. de Rooij. 1998. A spare parts stocking policy based on equipment criticality. *International Journal of Production Economics* 56-57: 69-77

[2] Dekker, R., R. M. Hill, M. J. Kleijn, R. H. Teunter. 2002. On the (S-1, S) lost sales inventory model with priority demand classes. *Naval Research Logistics* 49: 593-610

[3] Deshpande, V., M. A. Cohen, K. Donohue. 2003. A threshold rationing policy for service differentiated demand classes. *Management Science* 49(6): 683-703

[4] Deshpande, V. and M. A. Cohen, 2005. A Nested Threshold Inventory Rationing Policy for Multiple Demand Classes in Inventory Systems with Replenishment. *Working paper,* Krannert School of Management, Purdue University, West Lafayette, IN.

[5] Fadiloglu, M. M. and O. Bulut, 2010. An Embedded Markov Chain Approach to Stock Rationing. *Operations Research Letters* 38(6): 510-515.

[6] Tijms, H.C. 1986. Stochastic Modelling and Analysis: A Computational Approach. John Wiley & Sons, Great Britain, 1986.

[7] Vicil, O. 2006. Threshold Inventory Rationing Model Analysis and Optimization. *Ph.D. Thesis*, School of ORIE, Cornell University, Ithaca, NY.

[8] Vicil, O., P. Jackson. 2014. Rationing Inventories Among Demand Classes Under General Lead Time Distributions. *Working paper*, School of ORIE, Cornell University,Ithaca, NY.