International Journal of Production Economics 197 (2018) 174–185

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Improving efficiency in service systems by performing and storing "preliminary services"

Gabi Hanukov [a], Tal Avinadav [a,*], Tatyana Chernonog [a], Uriel Spiegel [a,b,d], Uri Yechiali [c]

[a] Department of Management, Bar-Ilan University, Ramat Gan, 5290002, Israel
[b] Department of Economics, University of Pennsylvania, Philadelphia, USA
[c] Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Tel Aviv, 6997801, Israel
[d] Zefat College, Zefat, Israel

ARTICLE INFO

Keywords:
Queueing systems
Server's idle time
Preliminary services
Performance measures
Cost analysis

ABSTRACT

We propose a novel approach to improve efficiency in service systems. The idea is to utilize the server's idle time to perform and store "preliminary services" for customers who will arrive in the future. Such a model is relevant to settings in which service consists of multiple consecutive tasks, some of which are generic and needed by all customers (and thus can be performed even in their absence), while other require the customer's presence. To show the model's benefits, we formulate a two-dimensional single-server queueing-inventory system for which we derive closed-form expressions for the system's steady-state probabilities, as well as for its performance measures. Assuming linear costs for customers waiting in line and for stored preliminary services, a cost analysis determines the optimal maximal number of stored preliminary services in the system. Numerical examples illustrated with graphs demonstrate the advantages of our approach, in terms of cost savings, as compared with the classical M/M/1 model.

1. Introduction

Operations managers frequently face the difficult challenge of reducing service systems' "idle" time in order to improve those systems' efficiency. Two sources of idleness characterize such systems: either customers wait in line to be served, or servers stay idle while waiting for customers to arrive. Because of the stochastic nature of queues, neither of the two sources of idleness can be entirely eliminated. It is estimated that the annual monetary loss due to idleness of employees in organizations reaches billions of dollars per year (Malachowski and Simonini, 2006), which further emphasizes the importance of improving the efficiency of service systems.

The literature discusses two common approaches that might be used to mitigate the two sources of idleness. (i) Increasing the number of servers in order to reduce the waiting times of customers. The drawback of this approach is that it leads to an increase in the servers' idle time and thus reduces each server's utilization. (ii) Increasing servers' utilization. Many studies propose achieving this goal by adopting a so-called vacation model, in which, instead of being allowed to remain idle, servers perform ancillary duties ("vacations") that are not directly related to their main task (see, e.g., Levy and Yechiali, 1975, 1976; Doshi, 1986; Kella and Yechiali, 1988; Takagi, 1991; Rosenberg and Yechiali, 1993; Boxma et al., 2002; Yechiali, 2004; Jain and Jain, 2010; Wei et al., 2013b; Yang and Wu, 2015; Mytalas and Zazanis, 2015; and Guha et al., 2016). However, the need to wait for a server to complete such tasks may increase customers' waiting times. Thus, each of these two approaches (more servers or server vacations) improves one source of idleness at the expense of the other.

Idleness of servers has been analyzed in the literature from additional perspectives. For example, Armony (2005), Armony and Ward (2010, 2013), and Mandelbaum et al. (2012) investigated fair routing of customers to idle servers in large-scale systems with heterogeneous customers. Cachon and Zhang (2007) investigated allocation of jobs to strategic servers (state-dependent as well as state-independent policies) under a capacity choice game played between the servers. They showed that there are cases in which it is beneficial to allocate a job to

* Corresponding author.
E-mail addresses: german.khanukov@live.biu.ac.il (G. Hanukov), tal.avinadav@biu.ac.il (T. Avinadav), tatyana.chernonog@biu.ac.il (T. Chernonog), uriel.spiegel@biu.ac.il (U. Spiegel), uriy@post.tau.ac.il (U. Yechiali).

https://doi.org/10.1016/j.ijpe.2018.01.004
Received 7 August 2016; Received in revised form 5 January 2018; Accepted 6 January 2018
Available online 10 January 2018

a busy fast server rather than to an idle slow server. Clearly, these allocation approaches can serve to mitigate idleness of servers and customers in multi-server systems, but they are not applicable to single-server systems.

In this paper we propose a novel approach to improve the performance of service systems by utilizing servers' idle time in cases where the service can be decomposed into two stages. The first stage, denoted "preliminary service" (PS henceforth), can be performed in the absence of customers, and its outcome can be preserved until an actual service is requested. The second stage, denoted "complementary service" (CS henceforth), requires the presence of the customer to be completed. In such settings, in contrast to the case of a vacation model, in which servers are diverted to ancillary duties during their idle time, an idle server can be utilized to accumulate PSs and store them until customers arrive and require service. This approach leads to a reduction of customers' mean sojourn time, since a certain fraction of the customers receive only a CS upon arrival (as part of their service was prepared prior to their arrival), and do not have to wait for the full service (FS henceforth).

A representative example of an application of our model is a fast food restaurant in which food, e.g., hamburger patties, can be prepared before demand occurs, and only upon the arrival of a customer is a hamburger patty heated up, inserted into a bun and served to the customer. Another example is a bicycle shop, which can assemble parts of a bicycle before a purchase occurs, and subsequently assemble the remaining parts in accordance with the customer's specific requirements and preferences. Handmade nameplates for doors are another example in which service can be split up. The server can produce basic (not necessarily identical) nameplates from wood, clay or glass before an order is placed, and complete a nameplate for a specific customer upon request (e.g., writing the name, adding decorations, etc.).

Hypothetically, a server can produce PSs during its entire idle time to minimize the customers' sojourn time. However, we show, in our model, that when cost considerations are taken into account—such as holding costs of PSs in inventory and costs of customers' presence in the system—there may be a certain number of stored PSs beyond which it is more beneficial to keep the server idle rather than to occupy it with producing additional PSs. Herein, we analyze this innovative queueing-inventory system. Examples of other types of queueing-inventory systems, in which each customer requires a unit from inventory when being served, appear in Zhao and Lin (2011), and in Adacher and Cassandras (2014). We use the classical M/M/1 queue as a baseline for comparison, which is common practice in the literature (e.g., Andritsos and Tang, 2013; Wei et al., 2013a; Güler et al., 2014).

We can summarize the main contributions of this paper as follows:

- A novel single-server queueing-inventory system is formulated as a two-dimensional stochastic process, and a method to derive closed-form expressions for the system's steady-state probabilities and for its performance measures is provided.
- It is shown that under certain conditions related to the duration of the service stages and the cost structure, the performance of a system that produces and stores PSs is superior to that of a similar system but without PSs. Nevertheless, Theorem 1 states that the stability conditions of the two systems (with or without PSs) are the same.
- A condition is established in Proposition 1 under which a server that utilizes some of its idle time to produce PSs actually remains idle for a larger fraction of time compared with a server in a similar system that does not store PSs.
- Assuming linear costs for each waiting customer and each stored PS, results of a cost analysis are provided, demonstrating how the optimal maximal number of stored PSs is affected by the model parameters.

2. Notations and assumptions

The following notations and assumptions are used throughout the paper:

Notations	
FS	Full service rendered continuously
PS	Preliminary service
CS	Complementary service
λ	customers' mean arrival rate
μ	server's mean rate of performing FSs
α	server's mean rate of producing PSs
β	server's mean rate of performing CSs
n	a decision variable denoting the maximal number of stored PSs
L	number of customers in the system in the long run (a random variable)
S	number of PSs in the system in the long run (a random variable)
$p_{i,j}$	steady-state probability of finding the system in state $\{L=i, S=j\}$
R	rate matrix of the matrix geometric analysis
$L(n)$	mean number of customers in the system as a function of n
$L_q(n)$	mean number of customers in queue as a function of n
$W(n)$	mean sojourn time of a customer in the system as a function of n
$W_q(n)$	mean waiting time of a customer in queue as a function of n
$S(n)$	mean number of PSs in the system as a function of n
$S_q(n)$	mean number of PSs in inventory as a function of n
$T(n)$	mean time a PS resides in the system as a function of n
$T_q(n)$	mean time a PS resides in inventory as a function of n
$\alpha_{eff}(n)$	effective production rate of PSs as a function of n
c	cost per unit of time per customer in the system
h	holding cost per unit of time per inventoried PS
$Z(n)$	total expected cost per unit of time as a function of n
η	percentage reduction in total expected cost in comparison to the classical M/M/1 model
ξ	percentage reduction in idle time of the server in comparison to the classical M/M/1 model

Assumptions

1. We consider a single-server system with a Poisson arrival rate λ and exponentially-distributed full-service time with mean $1/\mu$.
2. The service can be split into two consecutive stages. The first stage, PS, can be performed in the absence of customers, and its outcome can be preserved until an actual service is requested. The second stage, CS, requires the actual presence of the customer to be completed.
3. When the system is empty, the server produces PSs at a Poisson rate α. The PSs are stored until the arrival of customers, and can be considered as work-in-process inventory whose aim is to reduce the sojourn time of customers in the system.
4. When the number of stored PSs reaches the value of n, the server stops producing PSs and becomes idle.
5. If a customer arrives at the front of the queue and a PS is available, the server immediately starts rendering a CS for that customer; otherwise, the customer receives an FS.
6. The CS time is assumed to be exponentially distributed with mean $1/\beta(< 1/\mu)$.
7. The decomposition of service into two separate stages (potentially with an intermission between them) does not affect service quality, which implies that customers have no preference between receiving CS or FS. This assumption suits cases in which the storage time of PSs is relatively short in comparison to the shelf-life duration of a PS.

We now justify our assumptions regarding the production rates. We emphasize that although the PSs are standard units, they are not produced in an automatic process (which implies a deterministic preparation time). Specifically, the variability of the PS production durations emerges from three sources: the server, the production process and raw materials. Variability associated with the human server may stem from external

interruptions during preparation (e.g., answering phone calls from customers and suppliers, drinking a cup of coffee, etc.) or variation over time in the server's levels of fatigue and concentration. Variability associated with the production process may stem, for example, from electrical failure, malfunctioning of a machine part, etc. Variability associated with raw materials may stem from these materials' non-homogenous quality, shape or size. These examples imply that, in non-automatic production processes, the variability of PS durations can be large. In line with prior works in the domain of production and inventory systems (Benjaafar et al., 2011; Flapper et al., 2012; Iravani et al., 2012), and in order to ensure mathematical tractability, we adopt the exponential distribution to characterize production and serving times. Second, we note that the actions included in a split service (PS followed by a CS) do not precisely overlap with the actions in a continuous FS. This difference has two main sources: operational and behavioral. The operational source includes additional setup actions, such as PS storage and retrieval, which are not necessary in a continuous FS. The behavioral source includes adding or omitting actions to make a present customer more satisfied, such as explaining the work process to the customer or ignoring phone calls during work. Thus, the duration of a continuous FS is not necessarily equal to the sum of two independent random variables exponentially distributed with parameters α and β. Moreover, even in terms of expected durations, $1/\mu$ is not necessarily equal to $1/\alpha + 1/\beta$, since when no PS is available, the presence of a customer may speed up the work of the server, by creating a positive supervision effect, or delay it, by creating negative interference. Our assumption that the service time of an FS, as well as the service time of each stage in a split service, is exponentially distributed leads to a two-dimensional continuous-time Markov chain for which there are probabilistic solution methods available. In what follows, we use these methods and obtain analytical results for this new service model, which has not been analyzed before.

3. Model formulation and steady state analysis

We formulate the model as a quasi birth-and-death (QBD) process. At time t, let L_t and S_t denote the number of customers and the number of PSs in the system, respectively. $\{L_t, S_t\}$ defines the state space of the queueing-inventory system at time t. Let $L \equiv \lim_{t\to\infty} L_t$ and $S \equiv \lim_{t\to\infty} S_t$, so the steady-state joint probability distribution of the two-dimensional Markovian process is given by $p_{i,j} = \Pr(L = i, S = j)$, $i = 0, 1, 2, \ldots, \infty$, $j = 0, 1, 2, \ldots, n$. The transition rate diagram is depicted in Fig. 1, and the steady-state equations of the process are given in Table 1.

Herein, we analyze the behavior of the queueing-inventory system in its steady state. Specifically, we obtain the condition for system stability (i.e., for a finite mean queue length), which is required when the population and the queue length are unbounded, and investigate which parameters affect it. Our main goal is calculating performance measures of the system, as given in the next section, which are based on expected values, as well as the fraction of time the server is idle. The first step to achieve this goal is to calculate the steady-state probabilities of the process in the long run. In what follows we provide methods to do so.

The two-dimensional Markov process defined in the previous section can be analyzed either by using probability generating functions (PGFs), as explained and applied in, e.g., Litvak and Yechiali (2003), Perel and Yechiali (2008, 2014); or by using matrix geometric analysis (Neuts, 1981), as applied in Zhao and Lin (2011). Although the two probabilistic methods lead to the same analytical results, in this problem their computational efficiency is not the same, as discussed in Appendix C. In what follows, we use the notations $\vec{0} \equiv (0, 0, \ldots, 0)$, $\vec{e} \equiv (1, 1, \ldots, 1)^T$, $\vec{v} \equiv (0, 1, 2, \ldots, n)^T$ for $n \geq 0$, $\vec{u} \equiv (0, 0, 1, 2, \ldots, n-1)^T$ for $n \geq 1$, $\vec{p}_i \equiv (p_{i,0}, p_{i,1}, p_{i,2}, \ldots, p_{i,n})$ for $i = 0, 1, 2, \ldots, \infty$, and I for the identity matrix.

Define a set of $n+1$ (partial) PGFs as follows: $G_j(z) = \sum_{i=0}^{\infty} p_{i,j} z^i$,

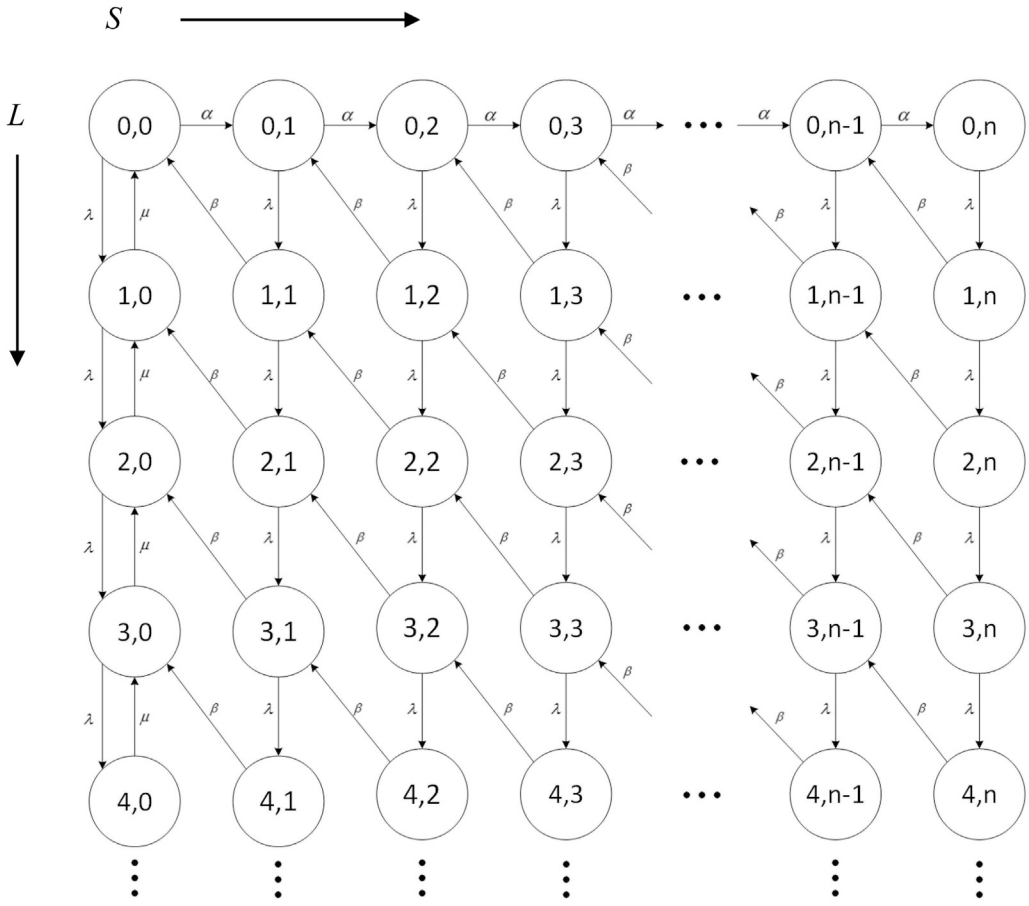


Fig. 1. System states and transition-rate diagram of the two-dimensional Markovian process.

Table 1
The steady-state equations.

	$j=0$	$1\le j\le n-1$	$j=n$
$i=0$	$(\alpha+\lambda)p_{0,0}=\beta p_{1,1}+\mu p_{1,0}$	$(\alpha+\lambda)p_{0,j}=\alpha p_{0,j-1}+\beta p_{1,j+1}$	$\lambda p_{0,n}=\alpha p_{0,n-1}$
$i\ge 1$	$(\mu+\lambda)p_{i,0}=\lambda p_{i-1,0}+\mu p_{i+1,0}+\beta p_{i+1,1}$	$(\beta+\lambda)p_{i,j}=\lambda p_{i-1,j}+\beta p_{i+1,j+1}$	$(\beta+\lambda)p_{i,n}=\lambda p_{i-1,n}$

$j=0,1,\ldots,n$, $|z|\le 1$, so $p_{i,j}=\frac{1}{i!}\left.\frac{d^i G_j(z)}{dz^i}\right|_{z=0}$, $i=0,1,\ldots,\infty$, $j=0,1,\ldots,n$. In essence, this definition means that for a given j, each probability vector $(p_{0,j},p_{1,j},p_{2,j},\ldots,p_{\infty,j})^T$ is "zipped" into the PGF $G_j(z)$. Thus, the infinite number of balance equations given in Table 1 can be replaced with a set of $n+1$ linear equations, in which $G_j(z)$, $j=0,1,\ldots,n$, are the variables. Define $d(z)\equiv(1-z)(\lambda z-\mu)$, $a(z)\equiv\lambda(1-z)+\beta$ and $\vec{g}(z)\equiv(G_0(z),G_1(z),\ldots,G_n(z))^T$. According to a procedure similar to the one presented in Litvak and Yechiali (2003) and in Perel and Yechiali (2008), $\vec{g}(z)$ is the solution of $A(z)\vec{g}(z)=\vec{b}(z)$, where

$$A(z)=\begin{pmatrix} d(z) & -\beta & 0 & 0 & \ldots & 0 & 0\\ 0 & za(z) & -\beta & 0 & \ldots & 0 & 0\\ 0 & 0 & za(z) & -\beta & \ldots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \ldots & za(z) & -\beta\\ 0 & 0 & 0 & 0 & \ldots & 0 & a(z)\end{pmatrix}_{(n+1)\times(n+1)},\quad \vec{b}(z)$$
$$=\begin{pmatrix}(\mu(z-1)-\alpha z)p_{0,0}-\beta p_{0,1}\\ \alpha z p_{0,0}-(\alpha z-\beta z)p_{0,1}-\beta p_{0,2}\\ \alpha z p_{0,1}-(\alpha z-\beta z)p_{0,2}-\beta p_{0,3}\\ \vdots\\ \alpha z p_{0,n-2}-(\alpha z-\beta z)p_{0,n-1}-\beta p_{0,n}\\ \alpha p_{0,n-1}+\beta p_{0,n}\end{pmatrix}.$$

To get $\vec{b}(z)$, we have to calculate the vector of boundary probabilities, $\vec{p}_0$. The common method of doing so is to characterize and use the roots of $|A(z)|$; however, in our problem, this method does not produce sufficiently informative equations to calculate $\vec{p}_0$.

Instead, we apply an alternative approach to obtain $\vec{p}_0$. Let $\Psi\equiv\{(i,j): i=0,1,\ldots,n-1; j=i+1,i+2,\ldots,n\}$. Then, the set of $N\equiv n(n+1)/2$ steady-state equations, which correspond to states $(i,j)\in\Psi$ in Table 1, includes $N+1$ steady-state probabilities: $p_{0,0}$ and $p_{i,j}$ with $(i,j)\in\Psi$. To obtain an additional independent (normalization) equation with the same probabilities, we use horizontal and vertical cuts. Let $p_{i,\bullet}\equiv\sum_{j=0}^{n}p_{i,j}$, $i=0,1,2,\ldots,\infty$, and $p_{\bullet,j}\equiv\sum_{i=0}^{\infty}p_{i,j}$, $j=0,\ldots,n$. Then, as can be observed in Fig. 1, for all horizontal cuts between row i and row $i+1$, the equilibrium is obtained by $\lambda p_{i,\bullet}=\mu p_{i+1,0}+\beta(p_{i+1,\bullet}-p_{i+1,0})$, $i=0,1,2,\ldots,\infty$, and for all vertical cuts between column j and column $j+1$, the equilibrium is obtained by $\alpha p_{0,j}=\beta(p_{\bullet,j+1}-p_{0,j+1})$, $j=0,\ldots,n-1$. Summing the horizontal equilibrium equations over $i=0,1,2,\ldots,\infty$, and the vertical equilibrium equations over $j=0,\ldots,n-1$, yields the following two equations: $\lambda=(\mu-\beta)(p_{\bullet,0}-p_{0,0})+\beta(1-p_{0,\bullet})$ and $\alpha(p_{0,\bullet}-p_{0,n})=\beta(1-p_{\bullet,0}-(p_{0,\bullet}-p_{0,0}))$, respectively. By extracting $p_{\bullet,0}$ from the latter equation, substituting it in the former, and using algebraic manipulations, we obtain the additional equation,

$$\left(\frac{1}{\alpha}+\frac{1}{\beta}-\frac{1}{\mu}\right)\sum_{j=0}^{n-1}p_{0,j}=\frac{1}{\alpha}\left(1-\frac{\lambda}{\mu}-p_{0,n}\right),\tag{1}$$

which includes only the boundary probabilities $p_{0,j}$, $j=0,1,2,\ldots,n$. Hence, a set of $N+1$ independent linear equations with $N+1$ steady-state probabilities exists, from which $\vec{p}_0$ can be extracted.

In our model, $p_{0,n}$ expresses the fraction of time during which the server is idle; this is because, given that there are no customers in the system, the server stops producing PSs and becomes idle once the number of stored PSs reaches the value n.

Proposition 1. *The fraction of time the server is idle, $p_{0,n}$, is smaller than that in the classical M/M/1 queue (i.e., $1-\lambda/\mu$) if and only if the total mean duration of a split service is larger than the mean duration of a continuous FS.*

Proof. Since $\sum_{j=0}^{n-1}p_{0,j}>0$ and $\frac{1}{\alpha}>0$, then, by (1), $1-\frac{\lambda}{\mu}-p_{0,n}>0$ *if and only if* $\frac{1}{\alpha}+\frac{1}{\beta}-\frac{1}{\mu}>0$, i.e., $p_{0,n}<1-\frac{\lambda}{\mu}$ *if and only if* $\frac{1}{\alpha}+\frac{1}{\beta}>\frac{1}{\mu}$.

Proposition 1 can be explained as follows: It is likely that, due to loss of efficiency, the total mean duration of a split service is longer than that of a continuous FS. In some cases, however, the opposite is true, e.g., when a customer interferes with the server or distracts him during the service and thus slows down the work. The latter case leads to what may be considered a paradox: utilizing some of the server's idle time to increase productivity actually increases the fraction of time during which the server is idle.

We turn now to a matrix geometric analysis. Consider a lexicographic order of the system's states, $\{(0,0),(0,1),\ldots,(0,n);(1,0),(1,1),\ldots,(1,n);\ldots;(i,0),(i,1),\ldots,(i,n);\ldots\}$. We construct an infinitesimal generator matrix, denoted Q:

$$Q=\begin{pmatrix} B & A_0 & 0 & 0 & \cdots\\ A_2 & A_1 & A_0 & 0 & \cdots\\ 0 & A_2 & A_1 & A_0 & \\ \vdots & & \ddots & \ddots & \ddots\end{pmatrix},$$

where the matrices B, A_0, A_1 and A_2, each of order $(n+1)\times(n+1)$, are given by

$$B=\begin{pmatrix}-(\alpha+\lambda) & \alpha & 0 & \cdots & 0 & 0\\ 0 & -(\alpha+\lambda) & \alpha & & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & 0 & & -(\alpha+\lambda) & \alpha\\ 0 & 0 & 0 & \cdots & 0 & -\lambda\end{pmatrix},\quad A_0$$
$$=\begin{pmatrix}\lambda & 0 & \cdots & 0 & 0\\ 0 & \lambda & & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & & \lambda & 0\\ 0 & 0 & \cdots & 0 & \lambda\end{pmatrix}$$

$$A_1=\begin{pmatrix}-(\mu+\lambda) & 0 & \cdots & 0 & 0\\ 0 & -(\beta+\lambda) & & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & & -(\beta+\lambda) & 0\\ 0 & 0 & \cdots & 0 & -(\beta+\lambda)\end{pmatrix},\quad A_2$$
$$=\begin{pmatrix}\mu & 0 & \cdots & 0 & 0\\ \beta & 0 & \cdots & 0 & 0\\ 0 & \beta & \ddots & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & \cdots & \beta & 0\end{pmatrix}.$$

Theorem 1. *The stability condition of the queuing-inventory system is $\lambda<\mu$ (exactly as in the classical M/M/1 queue), and it is independent of the PS production rate (α) and of the CS production rate (β).*

Proof. According to Neuts (1981, p. 83), the stability condition is $\vec{\pi}A_0\vec{e}<\vec{\pi}A_2\vec{e}$, where $\vec{\pi}=(\pi_0,\pi_1,\ldots,\pi_n)$ is the unique solution of the linear system $\vec{\pi}[A_0+A_1+A_2]=\vec{0}$ and $\vec{\pi}\cdot\vec{e}=1$. In our case, $\vec{\pi}=(1,0,\ldots,0)$, and the stability condition translates into $\lambda<\mu$.

Theorem 1 can be intuitively explained as follows: the CS production rate β is exercised only when there are PSs available in the system. At the

moment the PSs are exhausted, the service returns to its regular rate, μ. Thus, when the number of customers in the system becomes sufficiently large, all PSs will be used, and the system will imitate a classical M/M/1 queue, resulting in the stability condition of the latter queue, which is not influenced either by β or by the production rate of PSs, α.

As before, let $\vec{p}_i \equiv (p_{i,0}, p_{i,1}, \ldots, p_{i,n-1}, p_{i,n})$, and let R be a matrix of size $(n+1)\times(n+1)$ that satisfies

$$A_0 + RA_1 + R^2A_2 = 0_{n+1,n+1}. \tag{2}$$

Since there may be several values for each entry in R, only the smallest positive value should be taken (Neuts, 1981, p. 82). In most cases, the entries $r_{i,j}$ of R can be found only by numerical calculations (see Chapter 8 in Latouche and Ramaswami, 1999). A common method for computing the entries of R is by *successive substitutions* (Neuts, 1981, p. 37). For our problem, however, we have succeeded in obtaining closed-form expressions for *all* $r_{i,j}$, as given in Theorem 2 below. Such an explicit *complete* solution is rare in the literature. The solution is related to Catalan numbers (Koshy, 2008), which are a sequence of natural numbers (1, 1, 2, 5, 14, 42, 132, 429,…), defined by $C_m = (2m)!/((m+1)!m!)$, $m = 0, 1, 2, \ldots$. These numbers are associated with various counting problems, often involving recursively-defined objects. An example of C_m, which is closely related to our problem, is the number of monotonic lattice paths (Dershowitz and Rinderknecht, 2015) along the edges of a grid with $m \times m$ square cells that do not pass above the diagonal.

Theorem 2.

$$r_{i,0} = \begin{cases} \lambda/\mu & i = 0 \\ \dfrac{C_i\beta^{i-1}\lambda^{i+1}}{\mu(\beta+\lambda)^{2i-1}} + \displaystyle\sum_{k=1}^{i-1}\frac{C_{i-k}\beta^{i-k-1}\lambda^{i-k+1}}{(\beta+\lambda)^{2(i-k)}}r_{k,0} & 1 \le i \le n \end{cases} \quad \text{and} \quad r_{i,j} = \begin{cases} 0 & 0 \le i < j \le n \\ \dfrac{C_{i-j}\beta^{i-j}\lambda^{i-j+1}}{(\beta+\lambda)^{2(i-j)+1}} & 0 < j \le i \le n \end{cases}.$$

Proof. The proof is based on calculating the explicit entries of the left-hand side of (2) and using induction. The details are given in Appendix A.

Note: The explicit representation of the matrix R for $n = 7$ is given in Appendix B.

As a consequence of Theorem 2 and its Proof in Appendix A, we state:

Corollary 1. (*i*) *All the entries of R above the main diagonal are zero.* (*ii*) *Except for the first column, all the entries in any diagonal, whether the main diagonal or any diagonal below it, are equal to one another.*

In order to calculate $p_{i,j}$, $i = 0, 1, \ldots, \infty$, $j = 0, 1, \ldots, n$, we first have to obtain the vector of boundary probabilities $\vec{p}_0$. This is accomplished (Latouche and Ramaswami, 1999, p. 144) by solving the following linear system: $\vec{p}_0[B + RA_2] = \vec{0}$ and $\vec{p}_0[I-R]^{-1}\vec{e} = 1$. Finally, the rest of the steady-state probabilities are calculated by $\vec{p}_i = \vec{p}_0R^i$, $i = 1, 2, \ldots, \infty$.

4. Performance measures

Two types of measures are used to evaluate the performance of our queueing-inventory system. The first type refers to the customers in the service system, and the second refers to the inventory of PSs. We present each of these measures as a function of the decision variable n, which can be used to minimize the total expected cost of the system (see Section 5). The first type includes the mean number of customers in the service system, $L(n) = \sum_{i=1}^{\infty} i p_{i,\bullet}$, and in queue, $L_q(n) = \sum_{i=1}^{\infty}(i-1)p_{i,\bullet} = L(n) - (1 - p_{0,\bullet})$, from which we immediately derive the mean sojourn time, $W(n) = L(n)/\lambda$, and waiting time in queue, $W_q(n) = L_q(n)/\lambda$, by applying Little's law. The second type includes the mean number of PSs in the system, $S(n) = \sum_{j=1}^{n} j p_{\bullet,j}$, and in inventory,
$S_q(n) = \sum_{j=1}^{n}[jp_{0,j} + (j-1)(p_{\bullet,j} - p_{0,j})] = S(n) - 1 - p_{0,0} + p_{0,\bullet} + p_{\bullet,0}$,
from which we immediately derive the mean durations of time for which a PS resides in the system, $T(n) = S(n)/\alpha_{eff}(n)$, or in inventory, $T_q(n) = S_q(n)/\alpha_{eff}(n)$, by applying Little's law with the effective production rate of PSs, $\alpha_{eff}(n) = \alpha(p_{0,\bullet} - p_{0,n})$. Note that we distinguish between $i = 0$ and $i \ge 1$ in the summands of $S_q(n)$, since when customers are present, one of the PSs is removed from inventory (becomes a CS).

When applying PGFs, we use the relations $\frac{d}{dz}G_j(z)\big|_{z=1} = \sum_{i=0}^{\infty} i p_{i,j}$ and $G_j(1) = p_{\bullet,j}$ to obtain $L(n) = \sum_{j=0}^{n}\frac{d}{dz}G_j(z)\big|_{z=1}$ and $S(n) = \sum_{j=1}^{n} jG_j(1)$, respectively. On the other hand, when applying the matrix geometric analysis approach, we use the relations $\sum_{i=1}^{\infty} iR^{i-1} = [I-R]^{-2}$ and $\sum_{i=0}^{\infty} R^i = [I-R]^{-1}$ to obtain

$$L(n) = \sum_{i=1}^{\infty} i\,\vec{p}_i\cdot\vec{e} = \sum_{i=1}^{\infty} i\left(\vec{p}_1R^{i-1}\right)\vec{e} = \vec{p}_1\left(\sum_{i=1}^{\infty} iR^{i-1}\right)\vec{e} = \vec{p}_0R[I-R]^{-2}\vec{e} \tag{3}$$

and $S(n) = \sum_{i=0}^{\infty}(\vec{p}_i\cdot\vec{v}) = \vec{p}_0(\sum_{i=0}^{\infty}R^i)\vec{v} = \vec{p}_0[I-R]^{-1}\vec{v}$, respectively.

The mean number of PSs in inventory is calculated as

$$S_q(n) = \sum_{j=1}^{n} p_{0,j} + \sum_{j=1}^{n}\left[(j-1)p_{\bullet,j}\right] = \vec{p}_0\cdot\vec{e} - p_{0,0} + \sum_{i=0}^{\infty}\vec{p}_i\cdot\vec{u} = \vec{p}_0\cdot\vec{e} - p_{0,0} + \vec{p}_0\left(\sum_{i=0}^{\infty}R^i\right)\vec{u} = \vec{p}_0\cdot\vec{e} - p_{0,0} + \vec{p}_0[I-R]^{-1}\vec{u}. \tag{4}$$

Explicit expressions of the performance measures $L(n), L_q(n), S(n), S_q(n), T(n)$ and $T_q(n)$ for $n = 1$ and $n = 2$ are presented in Appendix C.

5. Cost analysis: finding the optimal value of *n*

In this section we provide the results of a numerical study demonstrating how to determine the optimal value of n, the maximum number of preliminary services that can be stored at any given time, using a model that assumes linear costs for each waiting customer and stored preliminary service. We note that the problem of analytically obtaining the optimal value of the control parameter n seems to be intractable. We consider two types of cost rates: one is proportional to the number of customers in the system, $L(n)$; the other is proportional to the number of PSs held in inventory, $S_q(n)$. Let c be the cost per unit of time per customer in the system, and let h be the holding cost per unit of time per inventoried PS (assuming no holding cost for a PS that moves on to the CS phase). Note that c takes into account the cost of the space required for a

customer who is waiting for or currently receiving service, as well as loss of goodwill due to prolonging the customer's stay in the system. The term h takes into account only the additional holding cost of a PS beyond the cost of holding the raw materials used to produce it. For example, in a bicycle shop, preassembled bicycle parts take up considerably less storage volume than an assembled bike does. In line with the dominant approach in the domain of inventory systems, we assume that holding costs are linear in the number of stored units (see, e.g., Avinadav and Henig, 2015; Avinadav et al., 2016).

The objective is to minimize the total expected cost per time unit by controlling the maximal number of stored PSs, i.e.,

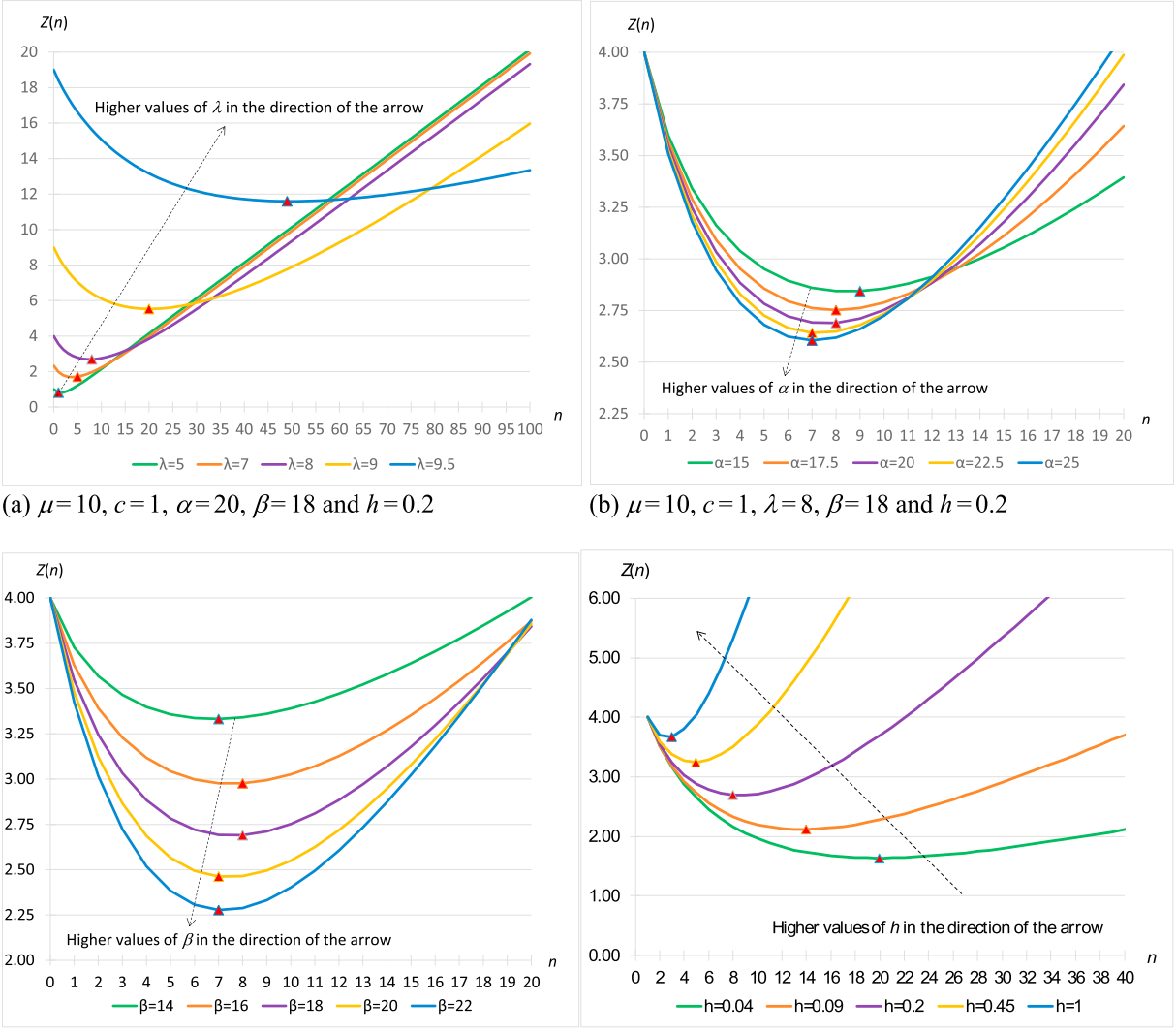
$$\min_{n\in\{0,1,2,\ldots\}} \{Z(n) = cL(n) + hS_q(n)\}. \quad (5)$$

Equations (3) and (4) provide expressions for $L(n)$ and $S_q(n)$, respectively, so a line search can be readily applied to find the optimal value of n over a closed interval. However, it is difficult to derive the properties of $Z(n)$ analytically, especially with regard to convexity. Therefore, using an efficient line-search method, such as the golden section search (see Bazaraa et al., 2006), does not guarantee finding the global minimum of $Z(n)$. Since $L(n)$ decreases in n, and $S_q(n)$ increases in n, we conjecture that $Z(n)$ is convex in n. In this section, we examine this conjecture using numerical examples and conduct a sensitivity analysis. Indeed, the numerical analysis for values of n up to 100 supports our conjecture.

Table 2
Parameter values in the examples presented in Fig. 2.

Parameter	Fig. 2(a)	Fig. 2(b)	Fig. 2(c)	Fig. 2(d)
λ	5, 7, 8, 9, 9.5	8	8	8
μ	10	10	10	10
α	20	15, 17.5, 20, 22.5, 25	20	20
β	18	18	14, 16, 18, 20, 22	18
c	1	1	1	1
h	0.2	0.2	0.2	0.04, 0.09, 0.2, 0.45, 1

We use the following parameter values as a base-example: $\lambda = 8$, $\mu = 10$, $\alpha = 20$, $\beta = 18$, $c = 1$ and $h = 0.2$. These values were chosen such that (i) $\lambda < \mu$, (ii) $1/\alpha + 1/\beta > 1/\mu$, and (iii) h is smaller than c to reflect the fact that the cost of making a customer wait for service is higher than the cost of storing a PS. In the case of a bicycle shop, for example, this assumption seems reasonable, given that such shops are usually constructed to accommodate numerous new bicycles



(a) $\mu = 10$, $c = 1$, $\alpha = 20$, $\beta = 18$ and $h = 0.2$

(b) $\mu = 10$, $c = 1$, $\lambda = 8$, $\beta = 18$ and $h = 0.2$

(c) $\mu = 10$, $c = 1$, $\lambda = 8$, $\alpha = 20$ and $h = 0.2$

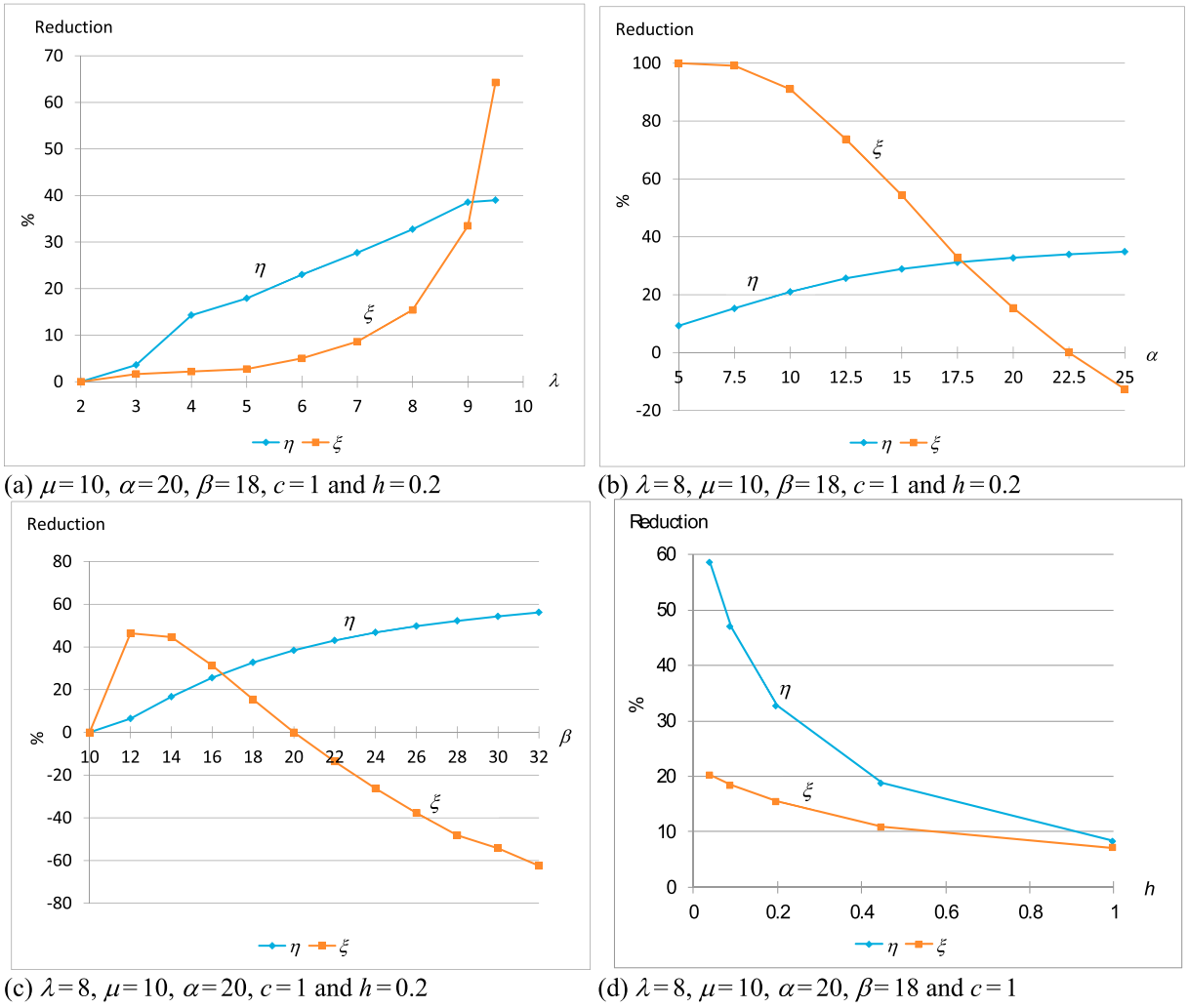
(d) $\mu = 10$, $c = 1$, $\lambda = 8$, $\alpha = 20$ and $\beta = 18$

Fig. 2. Total expected cost per time unit, $Z(n)$, as a function of n.

comfortably, whereas a waiting customer contributes to crowding in the shop and is likely to grow impatient as time passes. We calculate $Z(n)$ for $n = \{0, 1, 2, ..., 100\}$ and repeat this process for other parameter values as follows: we keep the values of μ and c fixed, and use four additional values (two above and two below the base value) for each parameter, where the other parameter values of the base-example are held constant. Specifically, we use the following additional parameter values: $\lambda = \{5, 7, 9, 9.5\}$, $\alpha = \{15, 17.5, 22.5, 25\}$ and $\beta = \{14, 16, 20, 22\}$. For the holding cost we use $h = \{0.04, 0.09, 0.2, 0.45, 1\}$; these values follow a logarithmic scale with coefficient $\sqrt{5}$, to allow ratios of c/h to be between 1 and 25. Since in the numerical examples the difference series $\{Z(n+1) - Z(n)\}$ for each value of λ, α, β and h are all monotonic increasing over the domain $n = \{0, 1, 2, ..., 99\}$, the objective function $Z(n)$ is convex on the integers over this domain. In order to emphasize the differences among the plots for different parameter values, we limit the n axis in Fig. 2(b) and (c) and (d) to $n = 20$. In what follows we define n^* as the optimal value of n in each numerical example, and depict the optimal point $(n^*, Z(n^*))$ as a triangle on the corresponding curve in Fig. 2. The parameter values are summarized in Table 2, where each column refers to a subgraph (a-d) in Fig. 2.

Fig. 2(a) presents $Z(n)$ for five different values of λ, and shows that n^* and $Z(n^*)$ increase in λ, and that when λ gets closer to μ, $Z(n)$ becomes only slightly sensitive to changes in n in the vicinity of n^*. Fig. 2(b) presents $Z(n)$ for five different values of α, and shows that n^* does not increase in α, whereas $Z(n^*)$ decreases in α. Fig. 2(c) presents $Z(n)$ for five different values of β, and shows, interestingly, that n^*, as a function of β, first increases from 7 to 8 and then decreases from 8 to 7, whereas $Z(n^*)$ decreases in β. We investigated the effect of β for additional parameter values: $\alpha = \{15, 25\}$ and $\mu = \{9, 12, 14\}$, which are not presented in Fig. 2(c), and obtained the same qualitative result. Fig. 2(d) presents $Z(n)$ for five different values of h, and shows that n^* decreases in h, whereas $Z(n^*)$ increases in h.

The results outlined above point to the following important practical implications. (i) The observed convexity of the total expected cost in n implies that the decision maker can use efficient line-search techniques, characterized by low computational complexity, to find the optimal value of n. (ii) The faster the server produces PSs (a larger value of α), the smaller the optimal inventory of PSs. This relationship reflects the fact that when the server can replenish the PS inventory quickly, even during brief periods of idle time, there is less of a necessity to store large numbers of PSs, and by storing fewer PSs the decision maker can benefit from the savings in holding costs. (iii) When the CS rate (β) is either low or high, it is economically beneficial to reduce the maximal number of



(a) $\mu = 10$, $\alpha = 20$, $\beta = 18$, $c = 1$ and $h = 0.2$

(b) $\lambda = 8$, $\mu = 10$, $\beta = 18$, $c = 1$ and $h = 0.2$

(c) $\lambda = 8$, $\mu = 10$, $\alpha = 20$, $c = 1$ and $h = 0.2$

(d) $\lambda = 8$, $\mu = 10$, $\alpha = 20$, $\beta = 18$ and $c = 1$

Fig. 3. Plots of η and ξ (the percentage reduction in the total expected cost and in the server's idle time, respectively, compared with an M/M/1 queue) for various parameter values.

Table 3
Parameter values in the examples presented in Fig. 3.

Parameter	Fig. 3(a)	Fig. 3(b)	Fig. 3(c)	Fig. 3(d)
λ	2, 3,…,9,9.5	8	8	8
μ	10	10	10	10
α	20	5,7.5,…,22.5, 25	20	20
β	18	18	10, 12,…, 30, 32	18
c	1	1	1	1
h	0.2	0.2	0.2	0.04, 0.09, 0.2, 0.45, 1

stored PSs, since, in the first case, the PSs stay a longer time in storage, thereby accumulating holding costs, and in the second case, the server has additional idle time, and thus has more opportunities to refill the PSs inventory. (iv) When the holding cost increases, the optimal maximal number of stored PSs decreases, as expected.

Clearly, the proposed queuing-inventory model under the optimal value of n is expected to perform at least as well as the M/M/1 model (which is equivalent to our model with $n = 0$), since our approach utilizes the server's idle time to increase productivity. Thus, in what follows, we investigate how different parameter values affect the extent to which our approach is advantageous over the M/M/1 model in terms of the percentage reduction in the total expected cost, denoted as $\eta = \frac{Z(0)-Z(n^*)}{Z(0)} \times 100\%$. Moreover, we evaluate the extent to which our model produces a percentage reduction in the server's idle time, denoted as $\xi = \frac{(1-\lambda/\mu)-p_{0,n^*}}{1-\lambda/\mu} \times 100\%$. Fig. 3(a)–(d) present η and ξ for different values of the parameters λ, α, β and h, where the other parameter values are held constant as in the base-example. The parameter values are summarized in Table 3, where each column refers to a subgraph (a-d) in Fig. 3.

Fig. 3(a)–(d) show that our model is always less costly than the M/M/1 queue ($0 \le \eta \le 100\%$), whereas the server's fraction of idle time may be smaller than ($0 < \xi \le 100\%$), equal to ($\xi = 0$), or larger than ($\xi < 0$) that in the M/M/1 queue. These results are due to the objective of minimizing the total expected cost per time unit given in (5). Fig. 3(a) shows that, for our model, the percentage reduction obtained for either the total expected cost or the fraction of the server's idle time is higher for higher values of the customer arrival rate λ. Fig. 3(b) shows that a higher PS production rate α results in a higher percentage reduction in the total expected cost, whereas the percentage reduction in the fraction of the server's idle time decreases. Moreover, as is claimed in Proposition 1, we see that for $\alpha \ge 1/(1/\mu - 1/\beta) = 22.5$ the fraction of the server's idle time in our model is even larger than that in the M/M/1 queue, as reflected in negative values of ξ. Fig. 3(c) shows that a higher CS production rate β results in a higher percentage reduction in the total expected cost. An interesting observation is that ξ is not monotonic in β: first it increases and then it decreases. Moreover, as is claimed in Proposition 1, we see that for $\beta \ge 1/(1/\mu - 1/\alpha) = 20$ the fraction of the server's idle time in the proposed model is even larger than that in the M/M/1 queue, as reflected in negative values of ξ. Fig. 3(d) shows that the percentage reductions in both the total expected cost and the fraction of the server's idle time are lower for higher values of the holding cost h. This observation is explained by the lower incentive to prepare and store PSs, as is shown in Fig. 2(d). The mechanism at work in our example (in which $\frac{1}{\alpha} + \frac{1}{\beta} > \frac{1}{\mu}$) is that as n increases, more jobs have a longer mean total service time, and thus the server's idle time decreases.

The practical implications of the above analysis are as follows: (i) The proposed approach produces greater cost savings when customers' arrival rate is higher, or when the server produces PSs and/or renders CSs more quickly, or when the holding costs are lower. (ii) When the rate at which the server produces PSs and/or renders CSs exceeds a certain value, implementing the proposed approach actually gives the server more idle time than it would have under the M/M/1 approach. The latter result points to a win-win-win situation, where the service owner, the server and the customer benefit from implementing the proposed approach.

6. Conclusions

This study investigated an innovative approach to increase the efficiency of queueing systems by utilizing servers' idle time to produce preliminary services for future incoming customers. In order to investigate such a system, we used a single-server Markovian queue and constructed a two-dimensional state space that considers both queue sizes and inventory levels. Using probabilistic methods, we calculated the steady-state probabilities of the system states and various performance measures. Application of the matrix geometric method allowed us to solve problems for $n \le 100$ within a time frame of few minutes, thereby enabling us to carry out cost analysis.

We have shown that the stability condition of our model is identical to that of the classical M/M/1 queue, which means that the maximal arrival rate of customers that the server can handle does not differ between the two approaches. Moreover, when the total average duration of a split service is smaller than that of a full service, the server is idle for a larger fraction of time than it would be in the classical M/M/1 queue. Numerical examples reveal two major insights: (i) the total expected cost function is convex over n; thus, efficient line-search methods can be used to find its optimal value; (ii) in extreme cases in which the duration of a CS is either very low or very high, it is economically more beneficial to reduce the maximal number of stored PSs. Specifically, for low CS rendering rates, the PSs remain in storage for longer periods of time, thereby accumulating holding costs. For high CS rendering rates, the server has ample idle time to refill the PSs inventory, so a smaller n is required.

There are various possible directions for further research in this domain. For example, it would be interesting to extend the model to multiple servers or to limited-capacity service systems. Another direction for future investigation would be to relax the exponential distribution assumptions. We expect that analyzing such a model will necessitate the use of simulation-based analysis, and it will be interesting to compare the results to those obtained in this paper. An important extension of our model would be to take into account the possibility that PSs can spoil or deteriorate in quality while being stored. This extension might be achieved, for example, by assuming that the PS shelf life duration is a random variable. In addition, we suggest analyzing a service system with two types of customers: one type that agrees to use a preliminary service prepared without him or her being present, and another type that insists on obtaining a continuous full service without any interruption. Another idea is to investigate how the splitting of the service could be done in practice, for example, what proportion of the full service should be allocated to the preliminary service considering the various associated costs. Finally, our approach has the potential to assist practitioners in evaluating the benefits (e.g., in terms of return on investment) of modifying certain continuous services such that they can be split up into two separate components.

Acknowledgments

The authors thank two anonymous referees for their insightful comments, which helped to improve the paper. This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1448/17).

Appendix A. Proof of Theorem 2

The explicit entries of the left-hand side of (2) are:

$$A_0+RA_1+R^2A_2=\begin{pmatrix}
\lambda-(\mu+\lambda)r_{0,0}+\mu\sum_{k=0}^{n}r_{0,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{0,k}r_{k,1} & -(\beta+\lambda)r_{0,1}+\beta\sum_{k=0}^{n}r_{0,k}r_{k,2} & -(\beta+\lambda)r_{0,2}+\beta\sum_{k=0}^{n}r_{0,k}r_{k,3} & \dots & -(\beta+\lambda)r_{0,n-2}+\beta\sum_{k=0}^{n}r_{0,k}r_{k,n-1} & -(\beta+\lambda)r_{0,n-1}+\beta\sum_{k=0}^{n}r_{0,k}r_{k,n} & -(\beta+\lambda)r_{0,n}\\
-(\mu+\lambda)r_{1,0}+\mu\sum_{k=0}^{n}r_{1,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{1,k}r_{k,1} & \lambda-(\beta+\lambda)r_{1,1}+\beta\sum_{k=0}^{n}r_{1,k}r_{k,2} & -(\beta+\lambda)r_{1,2}+\beta\sum_{k=0}^{n}r_{1,k}r_{k,3} & \dots & -(\beta+\lambda)r_{1,n-2}+\beta\sum_{k=0}^{n}r_{1,k}r_{k,n-1} & -(\beta+\lambda)r_{1,n-1}+\beta\sum_{k=0}^{n}r_{1,k}r_{k,n} & -(\beta+\lambda)r_{1,n}\\
-(\mu+\lambda)r_{2,0}+\mu\sum_{k=0}^{n}r_{2,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{2,k}r_{k,1} & -(\beta+\lambda)r_{2,1}+\beta\sum_{k=0}^{n}r_{2,k}r_{k,2} & \lambda-(\beta+\lambda)r_{2,2}+\beta\sum_{k=0}^{n}r_{2,k}r_{k,3} & \dots & -(\beta+\lambda)r_{2,n-2}+\beta\sum_{k=0}^{n}r_{2,k}r_{k,n-1} & -(\beta+\lambda)r_{2,n-1}+\beta\sum_{k=0}^{n}r_{2,k}r_{k,n} & -(\beta+\lambda)r_{2,n}\\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
-(\mu+\lambda)r_{n-2,0}+\mu\sum_{k=0}^{n}r_{n-2,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{n-2,k}r_{k,1} & -(\beta+\lambda)r_{n-2,1}+\beta\sum_{k=0}^{n}r_{n-2,k}r_{k,2} & -(\beta+\lambda)r_{n-2,2}+\beta\sum_{k=0}^{n}r_{n-2,k}r_{k,3} & \dots & \lambda-(\beta+\lambda)r_{n-2,n-2}+\beta\sum_{k=0}^{n}r_{n-2,k}r_{k,n-1} & -(\beta+\lambda)r_{n-2,n-1}+\beta\sum_{k=0}^{n}r_{n-2,k}r_{k,n} & -(\beta+\lambda)r_{n-2,n}\\
-(\mu+\lambda)r_{n-1,0}+\mu\sum_{k=0}^{n}r_{n-1,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{n-1,k}r_{k,1} & -(\beta+\lambda)r_{n-1,1}+\beta\sum_{k=0}^{n}r_{n-1,k}r_{k,2} & -(\beta+\lambda)r_{n-1,2}+\beta\sum_{k=0}^{n}r_{n-1,k}r_{k,3} & \dots & -(\beta+\lambda)r_{n-1,n-2}+\beta\sum_{k=0}^{n}r_{n-1,k}r_{k,n-1} & \lambda-(\beta+\lambda)r_{n-1,n-1}+\beta\sum_{k=0}^{n}r_{n-1,k}r_{k,n} & -(\beta+\lambda)r_{n-1,n}\\
-(\mu+\lambda)r_{n,0}+\mu\sum_{k=0}^{n}r_{n,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{n,k}r_{k,1} & -(\beta+\lambda)r_{n,1}+\beta\sum_{k=0}^{n}r_{n,k}r_{k,2} & -(\beta+\lambda)r_{n,2}+\beta\sum_{k=0}^{n}r_{n,k}r_{k,3} & \dots & -(\beta+\lambda)r_{n,n-2}+\beta\sum_{k=0}^{n}r_{n,k}r_{k,n-1} & -(\beta+\lambda)r_{n,n-1}+\beta\sum_{k=0}^{n}r_{n,k}r_{k,n} & \lambda-(\beta+\lambda)r_{n,n}
\end{pmatrix} \tag{A.1}$$

The following Proof of the formulae in Theorem 2 is based on induction. According to the structure of matrix R, as presented in Theorem 2 (and demonstrated in Appendix B for n=7), we split the proof into three stages, according to the location of the entries (i,j) within R. In stage (i), we deal with $0\le i<j\le n$, where $r_{i,j}=0$. In stage 2, we deal with $0<j\le i\le n$, and finally, in stage 3, we deal with the left column of matrix R.

Stage (i) We first prove by induction that $r_{i,j}=0,\ 0\le i<j\le n$. Starting with the last column of (A.1) above, i.e., column $j=n$, we obtain from (2):

$$-(\beta+\lambda)r_{i,n}=0,\quad i=0,1,2,\dots,n-1, \tag{A.2}$$

implying that $r_{i,n}=0$ for $i=0,1,2,\dots,n-1$.

We now assume that $r_{i,j}=0$ for all $0\le i<j\le n, j\ge n-m$, and show that $r_{i,n-m-1}=0$ for $0\le i\le n-m-2$. By (2) and the entries above the main diagonal in column $j=n-m-1$ of equation (A.1), $-(\beta+\lambda)r_{i,n-m-1}+\beta\sum_{k=0}^{n}r_{i,k}r_{k,n-m}=0,\ i=0,1,2,\dots,n-m-2$. By the induction assumption, $r_{k,n-m}=0$ for $k\le n-m-1$, and $r_{i,k}=0$ for $k>n-m-1$ since $i\le n-m-2$. Thus, $\sum_{k=0}^{n}r_{i,k}r_{k,n-m}=0$. Hence, $r_{i,n-m-1}=0$ for $0\le i\le n-m-2$, which proves the claim.

Stage (ii) We now prove by induction that $r_{i,j}=\frac{C_{i-j}\beta^{i-j}\lambda^{i-j+1}}{(\beta+\lambda)^{2(i-j)+1}}, 0<j\le i\le n$, where C_m is the m-th Catalan number. Starting with the main diagonal of (A.1), i.e., $i=j\ge 1$, we have from (2):

$$\lambda-(\beta+\lambda)r_{i,i}+\beta\sum_{k=0}^{n}r_{i,k}r_{k,i+1}=0,\quad i=1,2,\dots,n-1,\quad \text{and}\quad \lambda-(\beta+\lambda)r_{n,n}=0. \tag{A.3}$$

By (i), $r_{k,i+1}=0$ for $k\le i$, and $r_{i,k}=0$ for $k\ge i+1$, so $\sum_{k=0}^{n}r_{i,k}r_{k,i+1}=0$. Thus, from (A.3),

$$r_{i,i}=\lambda/(\beta+\lambda)=\frac{C_0\beta^0\lambda^1}{(\beta+\lambda)^{2(i-i)+1}},\quad i=1,2,\dots,n.$$

Next, we assume that $r_{i,j}=\frac{C_{i-j}\beta^{i-j}\lambda^{i-j+1}}{(\beta+\lambda)^{2(i-j)+1}}$ for all $0\le i-j\le m$, $1\le j\le i\le n$ (i.e., expressions for the entries in the main diagonal and in the $m-1$ diagonals below it) and show that $r_{i,j}=\frac{C_{i-j}\beta^{i-j}\lambda^{i-j+1}}{(\beta+\lambda)^{2(i-j)+1}}$ for $i-j=m+1$, $1\le j\le i\le n$. By (2) and the entries in the diagonal $i-j=m+1$ of (A.1),

$$-(\beta+\lambda)r_{i,i-m-1}+\beta\sum_{k=0}^{i}r_{i,k}r_{k,i-m}=0,\quad i=m+2,\dots,n. \tag{A.4}$$

From (i), $r_{k,i-m}=0$ for $k\le i-m-1$, so (A.4) can be written as $-(\beta+\lambda)r_{i,i-m-1}+\beta\sum_{k=i-m}^{i}r_{i,k}r_{k,i-m}=0,\ i=m+2,\dots,n$. By the induction assumption, $r_{i,k}=\frac{C_{i-k}\beta^{i-k}\lambda^{i-k+1}}{(\beta+\lambda)^{2(i-k)+1}}$ for $i-k\le m$ (and thus for $i-m\le k\le i$) and $r_{k,i-m}=\frac{C_{k-i+m}\beta^{k-i+m}\lambda^{k-i+m+1}}{(\beta+\lambda)^{2(k-i+m)+1}}$ for $k-i+m\le m$ (and thus for $i-m\le k\le i$), so $\sum_{k=i-m}^{i}r_{i,k}r_{k,i-m}=\frac{\beta^m\lambda^{m+2}}{(\beta+\lambda)^{2(m+1)}}\sum_{k=i-m}^{i}C_{i-k}C_{k-i+m}$. By modifying the indices in the summation, $\sum_{k=i-m}^{i}C_{i-k}C_{k-i+m}=\sum_{k=0}^{m}C_kC_{m-k}$, and from using the recurrence relation of the Catalan numbers, $\sum_{k=0}^{m}C_kC_{m-k}=C_{m+1}$. Thus, (A.4) can be written as

$$-(\beta+\lambda)r_{i,i-m-1}+\frac{C_{m+1}\beta^{m+1}\lambda^{m+2}}{(\beta+\lambda)^{2(m+1)}}=0,\quad i=m+2,\dots,n, \tag{A.5}$$

from which the claim is proved, i.e., $r_{i,j}=\frac{C_{i-j}\beta^{i-j}\lambda^{i-j+1}}{(\beta+\lambda)^{2(i-j)+1}}$, $i-j=m+1$, $1\le j\le i\le n$.

Stage (iii) Finally, we show $r_{0,0}=\lambda/\mu$ and $r_{i,0}=\frac{C_i\beta^{i-1}\lambda^{i+1}}{\mu(\beta+\lambda)^{2i-1}}+\sum_{k=1}^{i-1}\frac{C_{i-k}\beta^{i-k-1}\lambda^{i-k+1}}{(\beta+\lambda)^{2(i-k)}}r_{k,0}$, $1\le i\le n$. Starting with the top left entry of (A.1) above, we obtain the following:

$$\lambda-(\mu+\lambda)r_{0,0}+\mu\sum_{k=0}^{n}r_{0,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{0,k}r_{k,1}=0. \tag{A.6}$$

By (i), $r_{0,k}=0$ for $1\le k\le n$, so $\sum_{k=0}^{n}r_{0,k}r_{k,0}=(r_{0,0})^2$ and $\sum_{k=0}^{n}r_{0,k}r_{k,1}=0$. Thus, (A.6) is reduced to

$$\lambda-(\mu+\lambda)r_{0,0}+\mu(r_{0,0})^2=0. \tag{A.7}$$

The quadratic equation in (A.7) results in two roots, $r_{0,0}=1$ and $r_{0,0}=\lambda/\mu<1$, where the latter is the relevant root. We continue with calculating $r_{i,0}$ for $i=1,2,\ldots,n$, by solving (see (A.1))

$$-(\mu+\lambda)r_{i,0}+\mu\sum_{k=0}^{n}r_{i,k}r_{k,0}+\beta\sum_{k=0}^{n}r_{i,k}r_{k,1}=0,\quad 1\le i\le n. \tag{A.8}$$

By (i), $r_{i,k}=0$ for $k\ge i+1$, so $\sum_{k=0}^{n}r_{i,k}r_{k,0}=\sum_{k=0}^{i}r_{i,k}r_{k,0}$ and $\sum_{k=0}^{n}r_{i,k}r_{k,1}=\sum_{k=1}^{i}r_{i,k}r_{k,1}$. Thus, (A.8) can be written as

$$-(\mu+\lambda)r_{i,0}+\mu\left(r_{i,0}r_{0,0}+\sum_{k=1}^{i-1}r_{i,k}r_{k,0}+r_{i,i}r_{i,0}\right)+\beta\sum_{k=1}^{i}r_{i,k}r_{k,1}=0,\quad 1\le i\le n. \tag{A.9}$$

Since $r_{0,0}=\lambda/\mu$, $r_{i,i}=\lambda/(\beta+\lambda)$, $1\le i\le n$, and, by (ii), $r_{i,k}=\frac{C_{i-k}\beta^{i-k}\lambda^{i-k+1}}{(\beta+\lambda)^{2(i-k)+1}}$, $1\le k\le i$ and $r_{k,1}=\frac{C_{k-1}\beta^{k-1}\lambda^{k}}{(\beta+\lambda)^{2k-1}}$, $1\le k\le i$, then (A.9) can be written as

$$\frac{-\mu\beta}{\beta+\lambda}r_{i,0}+\mu\sum_{k=1}^{i-1}\frac{C_{i-k}\beta^{i-k}\lambda^{i-k+1}}{(\beta+\lambda)^{2(i-k)+1}}r_{k,0}+\frac{\beta^{i}\lambda^{i+1}}{(\beta+\lambda)^{2i}}\sum_{k=1}^{i}C_{i-k}C_{k-1}=0,\quad 1\le i\le n. \tag{A.10}$$

Multiplying (A.10) by $(\beta+\lambda)/(\mu\beta)$ and replacing $\sum_{k=1}^{i}C_{i-k}C_{k-1}$ with $\sum_{k=0}^{i-1}C_{i-1-k}C_{k}=C_i$ results in

$$r_{i,0}=\frac{C_i\beta^{i-1}\lambda^{i+1}}{\mu(\beta+\lambda)^{2i-1}}+\sum_{k=1}^{i-1}\frac{C_{i-k}\beta^{i-k-1}\lambda^{i-k+1}}{(\beta+\lambda)^{2(i-k)}}r_{k,0},\quad 1\le i\le n. \tag{A.11}$$

This completes the Proof.

Appendix B. Explicit calculation of the matrix R for $n=7$

By Theorem 2, the explicit representation of the matrix R for $n=7$ is:

$$R=\begin{pmatrix}
\frac{\lambda}{\mu} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\lambda^2}{\mu(\beta+\lambda)} & \frac{\lambda}{\beta+\lambda} & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\lambda^3(2\beta+\lambda)}{\mu(\beta+\lambda)^3} & \frac{\beta\lambda^2}{(\beta+\lambda)^3} & \frac{\lambda}{\beta+\lambda} & 0 & 0 & 0 & 0 & 0\\
\frac{\lambda^4(5\beta^2+4\beta\lambda+\lambda^2)}{\mu(\beta+\lambda)^5} & \frac{2\beta^2\lambda^3}{(\beta+\lambda)^5} & \frac{\beta\lambda^2}{(\beta+\lambda)^3} & \frac{\lambda}{\beta+\lambda} & 0 & 0 & 0 & 0\\
\frac{\lambda^5(14\beta^3+14\beta^2\lambda+6\beta\lambda^2+\lambda^3)}{\mu(\beta+\lambda)^7} & \frac{5\beta^3\lambda^4}{(\beta+\lambda)^7} & \frac{2\beta^2\lambda^3}{(\beta+\lambda)^5} & \frac{\beta\lambda^2}{(\beta+\lambda)^3} & \frac{\lambda}{\beta+\lambda} & 0 & 0 & 0\\
\frac{\lambda^6(42\beta^4+48\beta^3\lambda+27\beta^2\lambda^2+8\beta\lambda^3+\lambda^4)}{\mu(\beta+\lambda)^9} & \frac{14\beta^4\lambda^5}{(\beta+\lambda)^9} & \frac{5\beta^3\lambda^4}{(\beta+\lambda)^7} & \frac{2\beta^2\lambda^3}{(\beta+\lambda)^5} & \frac{\beta\lambda^2}{(\beta+\lambda)^3} & \frac{\lambda}{\beta+\lambda} & 0 & 0\\
\frac{\lambda^7(132\beta^5+165\beta^4\lambda+110\beta^3\lambda^2+44\beta^2\lambda^3+10\beta\lambda^4+\lambda^5)}{\mu(\beta+\lambda)^{11}} & \frac{42\beta^5\lambda^6}{(\beta+\lambda)^{11}} & \frac{14\beta^4\lambda^5}{(\beta+\lambda)^9} & \frac{5\beta^3\lambda^4}{(\beta+\lambda)^7} & \frac{2\beta^2\lambda^3}{(\beta+\lambda)^5} & \frac{\beta\lambda^2}{(\beta+\lambda)^3} & \frac{\lambda}{\beta+\lambda} & 0\\
\frac{\lambda^8(429\beta^6+572\beta^5\lambda+429\beta^4\lambda^2+208\beta^3\lambda^3+65\beta^2\lambda^4+12\beta\lambda^5+\lambda^6)}{\mu(\beta+\lambda)^{13}} & \frac{132\beta^6\lambda^7}{(\beta+\lambda)^{13}} & \frac{42\beta^5\lambda^6}{(\beta+\lambda)^{11}} & \frac{14\beta^4\lambda^5}{(\beta+\lambda)^9} & \frac{5\beta^3\lambda^4}{(\beta+\lambda)^7} & \frac{2\beta^2\lambda^3}{(\beta+\lambda)^5} & \frac{\beta\lambda^2}{(\beta+\lambda)^3} & \frac{\lambda}{\beta+\lambda}
\end{pmatrix}.$$

Appendix C. Explicit expressions for the cases $n=1$ and $n=2$

Here, we apply PGFs to obtain explicit expressions of the performance measures $L(n), L_q(n), S(n), S_q(n), T(n)$ and $T_q(n)$ for $n=1$ and $n=2$. We omit the explicit presentation for higher values of n since the expressions are cumbersome. The results are obtained by Maple 2015 software. Although the two probabilistic methods lead to the same results, in calculations the matrix geometric analysis has a significant advantage in our problem. Whereas the PGF method allows us to solve problems only up to $n=16$, due to computational limitations of the software, the matrix geometric analysis enables us to solve problems for $n\le 100$ within a time frame of few minutes with the same software. This computational-effort advantage is achieved due to Theorem 2, in which we obtain explicit expressions for all the entries of the matrix R.

The case of $n=1$

In order to calculate the boundary probabilities, $p_{0,0}$ and $p_{0,1}$, we solve the balance equation of state (0,1), $\lambda p_{0,1}=\alpha p_{0,0}$, combined with the

normalization equation (1), $\left(\frac{1}{\alpha}+\frac{1}{\beta}-\frac{1}{\mu}\right)p_{0,0}=\frac{1}{\alpha}\left(1-\frac{\lambda}{\mu}-p_{0,1}\right)$. Solving the above set results in $p_{0,0}=1/\left(\frac{\alpha}{\lambda}+\frac{1+\alpha/\beta}{1-\lambda/\mu}\right)$ and $p_{0,1}=1/\left(1+\frac{1/\alpha+1/\beta}{1/\lambda-1/\mu}\right)$.

Solving $A(z)(G_0(z)\ ,\ G_1(z))^T=\overrightarrow{b}(z)$ with $A(z)$ and $\overrightarrow{b}(z)$ from Section 3, and using algebraic manipulations, results in

$$G_0(z)=\frac{\frac{\lambda}{\alpha}\left(1-\frac{\lambda}{\mu}\right)\left(1-\frac{\lambda}{\lambda+\beta}\left(1-\frac{\alpha}{\mu}\right)z\right)}{\left(1+\lambda\left(\frac{1}{\alpha}+\frac{1}{\beta}-\frac{1}{\mu}\right)\right)\left(1-\frac{\lambda}{\mu}z\right)\left(1-\frac{\lambda}{\lambda+\beta}z\right)}\quad G_1(z)=\frac{1-\frac{\lambda}{\mu}}{\left(1+\lambda\left(\frac{1}{\alpha}+\frac{1}{\beta}-\frac{1}{\mu}\right)\right)\left(1-\frac{\lambda}{\lambda+\beta}z\right)}.$$

Then, according to Section 4, we derive

$$L(1)=\frac{\lambda(\beta\mu(\alpha\mu+\beta\lambda)-\alpha\lambda(\mu-\lambda)(\beta-\mu))}{\beta(\mu-\lambda)(\lambda\mu(\alpha+\beta)+\alpha\beta(\mu-\lambda))}\quad L_q(1)=\frac{\lambda^2(\beta\lambda(\alpha+\beta)+\alpha\mu(\mu-\lambda))}{\beta(\mu-\lambda)(\lambda\mu(\alpha+\beta)+\alpha\beta(\mu-\lambda))},$$

$$S(1)=\frac{\beta+\lambda}{\beta+\lambda\frac{\mu(\alpha+\beta)}{\alpha(\mu-\lambda)}}\quad S_q(1)=\frac{1}{1+\frac{\lambda\mu(\alpha+\beta)}{\alpha\beta(\mu-\lambda)}}.$$

To obtain $T(1)$ and $T_q(1)$, we first calculate the effective production rate of PSs, $\alpha_{eff}(1)=\alpha p_{0,0}=1/\left(\frac{1}{\lambda}+\frac{1/\alpha+1/\beta}{1-\lambda/\mu}\right)$. Then, we obtain $T(1)=1/\beta+1/\lambda$ and $T_q(1)=1/\lambda$, which can be interpreted as follows: the mean time of one unit in inventory equals the mean time between consecutive customer arrivals, whereas the mean time of one unit in the system includes, in addition, the mean CS production time. Note that when α approaches zero, the server does not succeed in performing any PS, and the system becomes the classical M/M/1.

The case of $n=2$

Similarly to the case of $n=1$, we obtain the boundary probabilities:

$$p_{0,0}=\frac{(\mu-\lambda)\beta(\beta+\alpha+\lambda)\lambda^2}{((\mu-\alpha)\beta+\alpha\mu)\lambda^3+((\mu-\alpha)\beta+\alpha\mu)(\beta+2\alpha)\lambda^2+((\mu-\alpha)\beta+2\alpha\mu)\beta\alpha\lambda+\alpha^2\beta^2\mu},$$

$$p_{0,1}=\frac{\lambda\alpha\beta(\beta+\lambda)(\mu-\lambda)}{((\mu-\alpha)\beta+\alpha\mu)\lambda^3+((\mu-\alpha)\beta+\alpha\mu)(\beta+2\alpha)\lambda^2+((\mu-\alpha)\beta+2\alpha\mu)\beta\alpha\lambda+\alpha^2\beta^2\mu},$$

$$p_{0,2}=\frac{\alpha^2\beta(\beta+\lambda)(\mu-\lambda)}{((\mu-\alpha)\beta+\alpha\mu)\lambda^3+((\mu-\alpha)\beta+\alpha\mu)(\beta+2\alpha)\lambda^2+((\mu-\alpha)\beta+2\alpha\mu)\beta\alpha\lambda+\alpha^2\beta^2\mu}.$$

Then, we obtain

$$L(2)=\frac{\left(\left(\lambda^3\left(\mu\beta^2+\alpha(\beta-\mu)(\lambda+3\alpha-\beta+\mu)\right)+\left(\left(\beta^2-4\beta\mu+3\mu^2\right)\alpha^2+\mu\beta\left(2\mu\alpha+\beta^2\right)\right)\lambda^2+\mu\alpha((3\mu-2\beta)\alpha+\beta\mu)\beta\lambda+\mu^2\beta^2\alpha^2\right)\lambda\right)}{(\mu-\lambda)\beta\left(\lambda^2(\alpha(\mu-\beta)+\beta\mu)(\lambda+\beta+2\alpha)+\alpha\beta((2\mu-\beta)+\beta\mu)\lambda+\mu\beta^2\alpha^2\right)},$$

$$S(2)=\frac{2\alpha(\mu-\lambda)\left(0.5\lambda^3+(\beta+1.5\alpha)\lambda^2+0.5\beta\lambda(\beta+4\alpha)+\alpha\beta^2\right)}{\lambda^2(\alpha(\mu-\beta)+\beta\mu)(\lambda+\beta+2\alpha)+\beta((2\mu-\beta)\alpha+\beta\mu)\alpha\lambda+\alpha^2\beta^2\mu},$$

from which it is easy to calculate $L_q(2)$ and $S_q(2)$, as well as $T(2)$ and $T_q(2)$, by using $\alpha_{eff}(2)=\alpha(p_{0,0}+p_{0,1})$.

References

Adacher, L., Cassandras, C.G., 2014. Lot size optimization in manufacturing systems: the surrogate method. Int. J. Prod. Econ. 155, 418–426.

Andritsos, D.A., Tang, C.S., 2013. The impact of cross-border patient movement on the delivery of healthcare services. Int. J. Prod. Econ. 145 (2), 702–712.

Armony, M., 2005. Dynamic routing in large-scale service systems with heterogeneous servers. Queueing Syst. 51, 287–329.

Armony, M., Ward, A.R., 2010. Fair dynamic routing in large-scale heterogeneous-server systems. Oper. Res. 58, 624–637.

Armony, M., Ward, A.R., 2013. Blind fair routing in large-scale service systems with heterogeneous customers and servers. Oper. Res. 61, 228–243.

Avinadav, T., Chernonog, T., Lahav, Y., Spiegel, U., 2016. Dynamic pricing and promotion expenditures in an EOQ model of perishable products. Ann. Oper. Res. 1–17.

Avinadav, T., Henig, M., 2015. Exact accounting of inventory costs in stochastic periodic-review models. Int. J. Prod. Econ. 169, 89–98.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M., 2006. Nonlinear Programming, Theory and Algorithms. John Wiley, Hoboken, NJ.

Benjaafar, S., Cooper, W.L., Mardan, S., 2011. Production-inventory systems with imperfect advance demand information and updating. Nav. Res. Logist. 58 (2), 88–106.

Boxma, O.J., Schlegel, S., Yechiali, U., 2002. A note on the M/G/1 queue with a waiting server, timer and vacations. Am. Math. Soc. Transl. Series 2 (207), 25–35.

Cachon, G.P., Zhang, F., 2007. Obtaining fast service in a queueing system via performance-based allocation of demand. Manage. Sci. 53, 408–420.

Dershowitz, N., Rinderknecht, C., 2015. The average height of Catalan trees by counting lattice paths. Math. Mag. 88 (3), 187–195.

Doshi, B.T., 1986. Queueing systems with vacations—a survey. Queueing Syst. 1, 29–66.

Flapper, S.D.P., Gayon, J.P., Vercraene, S., 2012. Control of a production–inventory system with returns under imperfect advance return information. Eur. J. Oper. Res. 218 (2), 392–400.

Guha, D., Goswami, V., Banik, A.D., 2016. Algorithmic computation of steady-state probabilities in an almost observable GI/M/c queue with or without vacations under state dependent balking and reneging. Appl. Math. Model. 40 (5), 4199–4219.

Güler, M.G., Bilgiç, T., Güllü, R., 2014. Joint inventory and pricing decisions when customers are delay sensitive. Int. J. Prod. Econ. 157, 302–312.

Iravani, S.M., Liu, T., Simchi-Levi, D., 2012. Optimal production and admission policies in make-to-stock/make-to-order manufacturing systems. Prod. Oper. Manage. 21 (2), 224–235.

Jain, M., Jain, A., 2010. Working vacations queueing model with multiple types of server breakdowns. Appl. Math. Model. 34 (1), 1–13.

Kella, O., Yechiali, U., 1988. Priorities in M/G/1 queue with server vacations. Nav. Res. Logist. 35, 23–34.

Koshy, T., 2008. Catalan Numbers with Applications. Oxford University Press, Oxford.

Latouche, G., Ramaswami, V., 1999. Introduction to Matrix Analytic Methods in Stochastic Modeling. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, PA.

Levy, Y., Yechiali, U., 1975. Utilization of idle time in an M/G/1 queueing system. Manage. Sci. 22, 202–211.

Levy, Y., Yechiali, U., 1976. An M/M/s queue with servers' vacations. INFOR 14 (2), 153–163.

Litvak, N., Yechiali, U., 2003. Routing in queues with delayed information. Queueing Syst. 43, 147–165.

Mandelbaum, A., Momcilovic, P., Tseytlin, Y., 2012. On fair routing from emergency departments to hospital wards: QED queues with heterogeneous servers. Manage. Sci. 58, 1273–1291.

Malachowski, D., Simonini, J. Wasted time at work costing companies billions in 2006. http://www.salary.com/wasted-time-at-work-still-costing-companies-billions-in-2006/

Mytalas, G.C., Zazanis, M.A., 2015. An M^X/G/1 queueing system with disasters and repairs under a multiple adapted vacation policy. Nav. Res. Logist. 62, 171–189.

Neuts, M.F., 1981. Matrix-geometric Solutions in Stochastic Models: an Algorithmic Approach. Johns Hopkins University Press, Baltimore, MD.

Perel, E., Yechiali, U., 2008. Queues where customers of one queue act as servers of the other queue. Queueing Syst. 60, 271–288.

Perel, N., Yechiali, U., 2014. The Israeli queue with retrials. Queueing Syst. 78, 31–56.

Rosenberg, E., Yechiali, U., 1993. The M^X/G/1 queue with single and multiple vacations under the LIFO service regime. Oper. Res. Lett. 14 (3), 171–179.

Takagi, H., 1991. Queueing Analysis, vol. 1. Vacation and Priority Systems, North-Holland, Amsterdam.

Wei, Y., Xu, C., Hu, Q., 2013a. Transformation of optimization problems in revenue management, queueing system, and supply chain management. Int. J. Prod. Econ. 146 (2), 588–597.

Wei, Y., Yu, M., Tang, Y., Gu, J., 2013b. Queue size distribution and capacity optimum design for N-policy Geo (λ1, λ2, λ3)/G/1 queue with setup time and variable input rate. Math. Comput. Model. 57, 1559–1571.

Yang, D.Y., Wu, C.H., 2015. Cost-minimization analysis of a working vacation queue with N-policy and server breakdowns. Comput. Ind. Eng. 82, 151–158.

Yechiali, U., 2004. On the M^X/G/1 queue with a waiting server and vacations. Sankhya 66 (1), 159–174.

Zhao, N., Lin, Z.T., 2011. A queueing-inventory system with two classes of customers. Int. J. Prod. Econ. 129, 225–231.