Some analysis results associated with the optimization problem for a discrete-time finite-buffer *NT*-policy queue

Miaomiao Yu[1,2,*], **Attahiru Sule Alfa**[2,3]

*Miaomiao Yu, mmyu75@163.com

[1]School of Science, Sichuan University of Science and Engineering, Zigong 643000, Sichuan, China

[2]Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada

[3]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa

Abstract The prime objective of this paperis to give some analysis results concerning the discrete-time finite-buffer *NT*-policy queue, which can be utilized to determine the optimal threshold values. By recording the waiting time of the leading customer in server's vacation period, the model is successfully described as a vector-valued Markov chain. Meanwhile, depending on the special block structure of the one-step transition probability matrix, the equilibrium queue length distri-bution is calculated through a more effective UL-type RG-factorization. Due to the number of customers served in the busy period does not have the structure of a Galton-Watson branching process, analysis of the regeneration cycle is regarded as a difficult problem in establishing the cost structure of the queueing system. However, employing the concept of *i*-busy period and some difference equation solving skills, the explicit expression for the expected length of the regeneration cycle is easily derived, and the stochastic decomposition structure of the busy period is also demonstrated. Finally, numerical results are offered to illustrate how the direct search method can be implemented to obtain the optimal management policy.

Keywords *NT*-policy queue · RG-factorization · *i*-busy period · Stochastic decomposition · Cost optimization

Mathematics Subject Classification 60K25 · 68M20 · 90B22

1 Introduction

Over the past few decades, a large class of policies for control of vacation queues has been discussed in the literature, viz., single and multiple vacation policies (Doshi 1986), *N*-policy (Yadin and Naor 1963), *T*-policy (Heyman 1977), *NT*-policy (Doganata 1990), *D*-policy (Balachandran 1973) and *F*-policy (Gupta 1995). The discrete-time queue with *NT*-policy was first studied by Alfa and Frigui (1996). In such a vacation queueing system, the idle server will be reactivated when either N customers have accumulated in the queue, or the waiting time of the leading customer has reached T time slots. After the server's vacation is interrupted, the busy period starts and it lasts until the queue becomes empty. This process continues so that the server is in either on or off state. As Alfa and Frigui pointed out, unlike previous works such as done by Gakis et al. (1995), Hur et al. (2003), Tadj (2003), Ke (2005), Jiang et al. (2010) and Zhang et al. (2011), the threshold T is measured from the epoch of arrival of the first customer who enters the system during the server vacation. By imposing a time limit T on the accumulation waiting time, the *NT*-policy avoids that customers suffer excessive delays because the server waits to initiate service until enough customers have arrived. Recently, using the probability generating function technique, Feyaerts et al. (2010) also studied the *NT*-policy in a discrete-time queueing system with independent Bernoulli arrivals and deterministic service times. By means of numerical examples, they illustrated that the *NT*-policy achieves better customer delay performance than the *N*-policy in case of a low rate arrival stream. Due to the significant advantage that mentioned above, this sort of queue has received a great deal of attention in recent literature (see Ke 2006a, b; Feyaerts et al. 2014), and from the practical point of view, this type of control policy can be utilized in the area of communication, manufacturing, and transportation systems to minimize the running costs while keeping a high level of customer satisfaction. For example, in a courier service it is desirable that a truck serving a particular origin-destination pair should carry more than one package on a trip. Thus, if the number of packages is less than N, the truck will wait until the Nth package arrives. On the other hand, in order to maintain good customer service, courier company does not want to keep package delayed too long. Hence, after the waiting time of the leading package has reached T time units, the truck will depart even if there is less than N packages. Clearly, according to the characteristics of termination mechanism of idle period, we may see that such phenomenon arising in courier companies can be well approximated by *NT*-policy queue. In addition, another important issue we are concerned about is that how to get an optimal control strategy to realize economical operation of the service facility. Thus, with the cost structure being constructed, Alfa and Li (2000) and Li and Alfa (2000) further studied the optimal *NT*-policy for both $M/G/1$ and $M/M/m$ queues. Based on the renewal reward theory (see Ross 1996), they obtained the optimal policy for minimizing the long-run average operating cost per unit time.

As one can see from above, existing research results on *NT*-policy queue are limited to infinite-buffer system, even though queues with finite-buffers commonly encountered in industrial practice. Especially, as far as we are aware the analysis results concerning the discrete-time finite-buffer *NT*-policy queue which can be used to find the optimal control strategies do not exist until now. Thus, based on the work done by Alfa and Frigui (1996), we will concentrate on solving the performance measures related to the optimal control strategies in this paper. Specifically, our main contributions are as follows: first, a simpler UL-type RG-factorization is applied to provide effective solutions for the block-structured Markov chains. Then, the mean number of customers in the system and the blocking probability of an arbitrary customer can be easily obtained. Second, we develop a first step analysis for the study of the busy period. The explicit expression and the stochastic decomposition structure of the busy period are also given by using some tips and tricks for algebraic manipulations. These topics have not been addressed sufficiently in the existing literature. Actually, analyzing the finite-buffer *NT*-policy queue gives some obvious advantages. The most one is that taking the limit as buffer capacity K approaches infinity allows us to get the result of the corresponding infinite-buffer system. Moreover, in the case of finite-buffer queue, we can analyze the unstable system, that is to say, the average arrival rate is higher than the average service rate. But at the same time, it should also be noted that since the number of customers served in the busy period does not have the structure of a Galton-Watson branching process, the regeneration cycle analysis for the finite-buffer queue is much more complex than the infinite-buffer counterpart. Therefore, getting the optimal threshold values for the *Geo*/*Geo*/1/K queue with *NT*-policy will be a challenging task. It requires some analytic and technical efforts. As a highlight of our study, a concise method for analysis of the regeneration cycle is presented. With a probabilistic argument and some simple algebraic manipulations, we easily obtain the expected length of the regeneration cycle. Therefore, the major obstacle that will encounter in this study can be successfully overcome.

According to the work done by Heyman (1977), the expectation of the queue length and the mean value of the regeneration cycle are essential for establishing the long-run average operating cost function per unit time. Thus, the rest of this paper is organized as follows. By stating the requisite assumptions and notations, the mathematical model is described in Sect. 2. Section 3 is dedicated to the calculation of the stationary distribution. The *Geo*/*Geo*/1/K queue with *NT*-policy is analyzed as a block-structured Markov chain with finite levels. A simpler UL-type RG-factorization is presented to provide effective solutions for the block-structured Markov chains. Furthermore, employing the extended definition of the busy period that might be initiated by multiple customers, the regeneration cycle of this model is analyzed in Sect. 4. In Sect. 5, a long-run expected cost function per unit time for the (N, T)-policy queue is constructed to determine the joint optimum threshold

values. Numerical example is also presented for illustrative purposes. Finally, we conclude with a brief summary in Sect. 6.

2 Queueing model description

In this paper, we consider a discrete-time single-server queueing system with finite storage capacity under the *NT*-policy. In the discrete-time situation, the time axis is divided into fixed-length contiguous period, called slots, and all queueing activities occur around slot boundaries. To be more specific, we suppose that potential departures occur in the time interval (t^-, t), while potential arrivals and the beginning or ending of the vacation take place in the time interval (t, t^+) (see Fig. 1). This also means that the queue is analyzed for the early arrival system (EAS).

The *NT*-policy queue is assumed to operate as follows. Inter-arrival times $\{A_i, i \geq 1\}$ are independent identically distributed random variables, and follow a geometric distribution: $\Pr\{A_i = j\} = \lambda\bar{\lambda}^{j-1}, 0 < \lambda < 1, j \geq 1$, where we use symbol $\bar{x} = 1 - x$, for any real number $x(0 < x < 1)$. The service times of the customers, denoted by S, are independent and geometrically distributed with parameter μ, where μ is the probability that a customer finishes his service in a time slot. The server starts service when a first customer arrives in an empty system and either one of the following happens:

(i) If the time elapsed since the first arrival during the vacation period reaches the predefined threshold T, the server will return to the system to begin service immediately (see Fig. 2a);

(ii) If N customers have accumulated in the queue before the timer expires, the vacation will be interrupted and the server immediately resumes the queue service (see Fig. 2b).

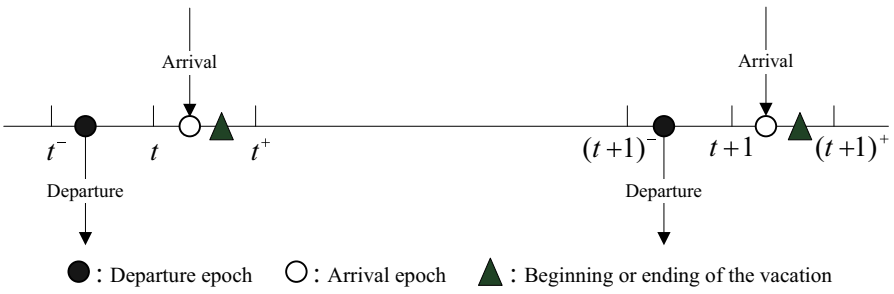

Fig. 1 Various time epochs in early arrival system

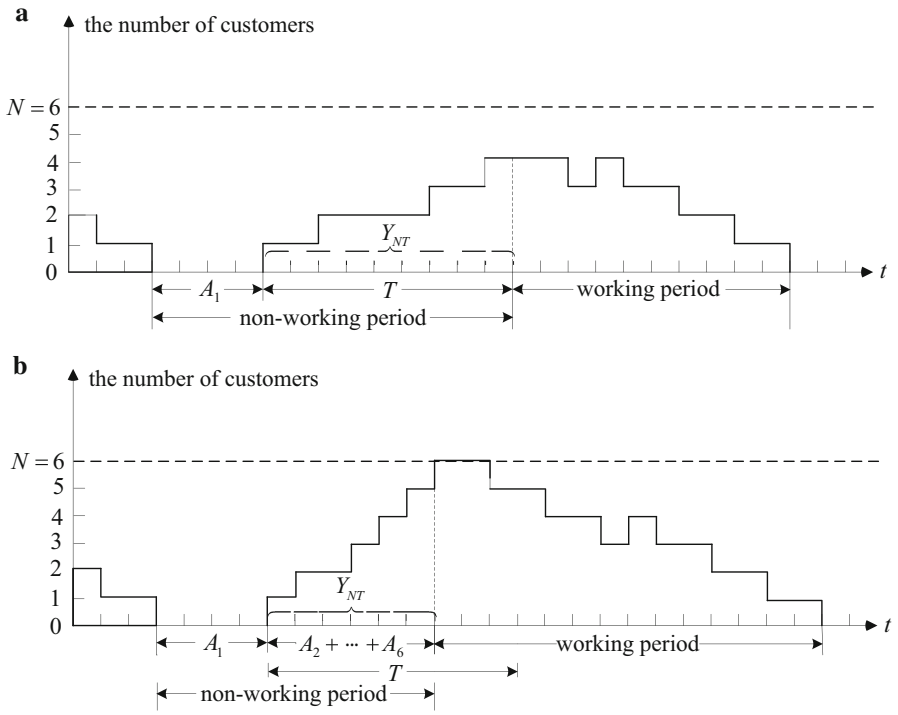

Fig. 2 Two possible evolution courses of the system content for an *NT*-policy queue with $N = 6$ and $T = 9$

Under the *NT*-policy, the server's status alternates between working and non-working states. Thus, a nonworking period followed by a working period together constitutes a regeneration cycle. Here, non-working period contains an accumulation process, denoted by Y_{NT}. The length of Y_{NT} is measured from the instant a first customer arrives to an empty queue to the time when the server is reactivated again. In Sect. 4, for establishing the long-run average operating cost function, we will use the concept of *i*-busy period (see Pacheco and Ribeiro 2008) to derive the mean value of the regeneration cycle.

Furthermore, we note that for $N = 1$, the system becomes a traditional work-conserving queue, without any threshold. If on the other hand $T < N$, only the time threshold T is relevant since it takes at least N time slots to accumulate N customers in the queue. Thus, as for this case, the service facility would only implement a *T*-policy. In this paper, we focus our attention on situations where both thresholds actually contribute to the scheduling discipline, i.e. we assume $1 < N \leq T < K$ in model analysis.

3 Steady-state queue length distribution

From Fig. 2, we observe that the operation of the system exhibits a cyclic behavior. When a first customer arrives in an empty queue, the system commences an accumulation process until at least one of the thresholds is reached. If we record the waiting time of the first customer in the accumulation process, the system formulated in Sect. 2 can be considered as a discrete-time Markov chain (DTMC). Thus, the state of the system at time t^+ is described by the following random variables:

- $X(t)$: The waiting time of the first customer at time t^+ during the accumulation process;
- $Q(t)$: The number of customers in the system at time t^+ (including the one being served, if any).

Consider

$$\begin{aligned}
\Omega_0 =&\{(X(t)=i,Q(t)=j)|i=0,j=0,1\},\\
\Omega_1 =&\{(X(t)=i,Q(t)=j)|i=1,2,\ldots,T-1,j=1,\ldots,\min(i+1,N-1)\},\\
\Omega_2 =&\{\,(Q(t)=j)|j=1,2,\ldots,K\},
\end{aligned}$$

where (0, 0) denotes the state with no customers in the queue and the server is on vacation, (0, 1) denotes the state with one customer in the queue and his elapsed waiting time is zero. Each state (i, j) of Ω_1 represents that the server is idle, there are j customers in the queue and the first customer has been waiting i time units. Therefore, the state space of the NT-policy queue is given by Ω, where $\Omega=\Omega_0\bigcup\Omega_1\bigcup\Omega_2$. Here, to give the reader a good intuitive understanding, we enumerate the states in the following order with assumption that $N=5,T=9$ and $K=12$,

$$\Omega=\{l(0),l(1),l(2),\ldots,l(8),1,2,\ldots,11,12\}.$$

For the sake of notational convenience, we use the symbol $l(i)$ to denote the union of $\min(i+1,N-1)$ states, namely

$$l(i)=\{(i,1),(i,2),\ldots,(i,\min(i+1,N-1))\},\quad i=1,2,\ldots,T-1,$$

and $l(0)$ represents a collection of states (0, 0) and (0, 1). Furthermore, we may find that the transition probability matrix $\boldsymbol{P}$ of this queueing system can be written as a partitioned matrix corresponding to the transitions among these above states, and exhibits the following block-structured form

$$\boldsymbol{P}=\begin{pmatrix}\boldsymbol{P}_{V,V} & \boldsymbol{P}_{V,W}\\ \boldsymbol{P}_{W,V} & \boldsymbol{P}_{W,W}\end{pmatrix},$$

where

$$\boldsymbol{P}_{V,V}=\begin{array}{c} \\ l(0) \\ l(1) \\ \vdots \\ l(N-3) \\ l(N-2) \\ \vdots \\ l(T-2) \\ l(T-1) \end{array}\begin{array}{c} \begin{array}{ccccccccc} l(0) & l(1) & l(2) & \cdots & l(N-2) & l(N-1) & \cdots & l(T-1) \end{array} \\ \begin{pmatrix} \boldsymbol{H}_1 \\ \boldsymbol{0}_{1\times 2} \end{pmatrix} \quad \begin{pmatrix} \boldsymbol{0}_{1\times 2} \\ \boldsymbol{H}_1 \end{pmatrix} \quad \boldsymbol{0}_{2\times 3} \quad \cdots \quad \boldsymbol{0}_{2\times(N-1)} \quad \boldsymbol{0}_{2\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{2\times(N-1)} \\ \boldsymbol{0}_{2\times 2} \quad \boldsymbol{0}_{2\times 2} \quad \boldsymbol{H}_2 \quad \cdots \quad \boldsymbol{0}_{2\times(N-1)} \quad \boldsymbol{0}_{2\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{2\times(N-1)} \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \\ \boldsymbol{0}_{(N-2)\times 2} \quad \boldsymbol{0}_{(N-2)\times 2} \quad \boldsymbol{0}_{(N-2)\times 3} \quad \cdots \quad \boldsymbol{H}_{N-2} \quad \boldsymbol{0}_{(N-2)\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{(N-2)\times(N-1)} \\ \boldsymbol{0}_{(N-1)\times 2} \quad \boldsymbol{0}_{(N-1)\times 2} \quad \boldsymbol{0}_{(N-1)\times 3} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times(N-1)} \quad \boldsymbol{H}_{N-1} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times(N-1)} \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \\ \boldsymbol{0}_{(N-1)\times 2} \quad \boldsymbol{0}_{(N-1)\times 2} \quad \boldsymbol{0}_{(N-1)\times 3} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times(N-1)} \quad \boldsymbol{0}_{(N-1)\times(N-1)} \quad \cdots \quad \boldsymbol{H}_{T-1} \\ \boldsymbol{0}_{(N-1)\times 2} \quad \boldsymbol{0}_{(N-1)\times 2} \quad \boldsymbol{0}_{(N-1)\times 3} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times(N-1)} \quad \boldsymbol{0}_{(N-1)\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times(N-1)} \end{array},$$

$$\boldsymbol{P}_{V,W}=\begin{array}{c} \\ l(0) \\ l(1) \\ l(2) \\ \vdots \\ l(N-3) \\ l(N-2) \\ \vdots \\ l(T-2) \\ l(T-1) \end{array}\begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & \cdots & N-1 & N & N+1 & \cdots & K-1 & K \end{array} \\ \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \cdots \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \cdots \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \\ \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \cdots \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \quad \cdots \quad \boldsymbol{0}_{2\times 1} \quad \boldsymbol{0}_{2\times 1} \\ \boldsymbol{0}_{3\times 1} \quad \boldsymbol{0}_{3\times 1} \quad \boldsymbol{0}_{3\times 1} \quad \cdots \quad \boldsymbol{0}_{3\times 1} \quad \boldsymbol{0}_{3\times 1} \quad \boldsymbol{0}_{3\times 1} \quad \cdots \quad \boldsymbol{0}_{3\times 1} \quad \boldsymbol{0}_{3\times 1} \\ \vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \quad \vdots \\ \boldsymbol{0}_{(N-2)\times 1} \quad \boldsymbol{0}_{(N-2)\times 1} \quad \boldsymbol{0}_{(N-2)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-2)\times 1} \quad \boldsymbol{0}_{(N-2)\times 1} \quad \boldsymbol{0}_{(N-2)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-2)\times 1} \quad \boldsymbol{0}_{(N-2)\times 1} \\ \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{C} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \\ \vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \quad \vdots \\ \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{C} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \\ \boldsymbol{D}_1 \quad \boldsymbol{D}_2 \quad \boldsymbol{D}_3 \quad \cdots \quad \boldsymbol{D}_{N-1} \quad \boldsymbol{C} \quad \boldsymbol{0}_{(N-1)\times 1} \quad \cdots \quad \boldsymbol{0}_{(N-1)\times 1} \quad \boldsymbol{0}_{(N-1)\times 1} \end{array},$$

$$\boldsymbol{P}_{W,V}=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ K-1 \\ K \end{array}\begin{array}{c} \begin{array}{ccccccccc} l(0) & l(1) & l(2) & \cdots & l(N-3) & l(N-2) & l(N-1) & \cdots & l(T-1) \end{array} \\ (\mu\bar{\lambda} \quad 0) \quad \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 3} \quad \cdots \quad \boldsymbol{0}_{1\times(N-2)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{1\times(N-1)} \\ \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 3} \quad \cdots \quad \boldsymbol{0}_{1\times(N-2)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{1\times(N-1)} \\ \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 3} \quad \cdots \quad \boldsymbol{0}_{1\times(N-2)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{1\times(N-1)} \\ \vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \\ \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 3} \quad \cdots \quad \boldsymbol{0}_{1\times(N-2)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{1\times(N-1)} \\ \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 2} \quad \boldsymbol{0}_{1\times 3} \quad \cdots \quad \boldsymbol{0}_{1\times(N-2)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \boldsymbol{0}_{1\times(N-1)} \quad \cdots \quad \boldsymbol{0}_{1\times(N-1)} \end{array},$$

$$\boldsymbol{P}_{W,W}=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \vdots \\ K-2 \\ K-1 \\ K \end{array}\begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & \cdots & K-2 & K-1 & K \end{array} \\ \mu\lambda+\bar{\mu}\bar{\lambda} \quad \bar{\mu}\lambda \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \\ \mu\bar{\lambda} \quad \mu\lambda+\bar{\mu}\bar{\lambda} \quad \bar{\mu}\lambda \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \\ 0 \quad \mu\bar{\lambda} \quad \mu\lambda+\bar{\mu}\bar{\lambda} \quad \bar{\mu}\lambda \quad \cdots \quad 0 \quad 0 \quad 0 \\ 0 \quad 0 \quad \mu\bar{\lambda} \quad \mu\lambda+\bar{\mu}\bar{\lambda} \quad \cdots \quad 0 \quad 0 \quad 0 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots \quad \vdots \\ 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad \mu\lambda+\bar{\mu}\bar{\lambda} \quad \bar{\mu}\lambda \quad 0 \\ 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad \mu\bar{\lambda} \quad \mu\lambda+\bar{\mu}\bar{\lambda} \quad \bar{\mu}\lambda \\ 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad \mu\bar{\lambda} \quad \mu\lambda+\bar{\mu} \end{array},$$

in which $\boldsymbol{0}_{m\times n}$ denotes a zero matrix of size $m \times n$. The block matrices appearing in $\boldsymbol{P}_{V,V}$ and $\boldsymbol{P}_{V,W}$ are given as follows:

For $i = 1, 2, \ldots, N-2$,

$$\boldsymbol{H}_i = \begin{pmatrix} \bar{\lambda} & \lambda & 0 & 0 & \cdots & 0 \\ 0 & \bar{\lambda} & \lambda & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \bar{\lambda} & \lambda & 0 \\ 0 & 0 & \cdots & 0 & \bar{\lambda} & \lambda \end{pmatrix}, \text{ and } \boldsymbol{H}_i \text{ has dimensions } i \times (i+1);$$

For $i = N-1, \ldots, T-1$,

$$\boldsymbol{H}_i = \boldsymbol{H} = \begin{pmatrix} \bar{\lambda} & \lambda & 0 & \cdots & 0 \\ 0 & \bar{\lambda} & \lambda & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \bar{\lambda} & \lambda \\ 0 & 0 & 0 & 0 & \bar{\lambda} \end{pmatrix}, \text{ and } \boldsymbol{H} \text{ is of dimension } (N-1) \times (N-1);$$

For $i = 1, 2, \ldots, N-1$,

$\boldsymbol{D}_i = \begin{cases} \bar{\lambda}\boldsymbol{e}_{N-1}^1, & i = 1 \\ \lambda \boldsymbol{e}_{N-1}^{i-1} + \bar{\lambda}\boldsymbol{e}_{N-1}^i, & i = 2, \ldots, N-1 \end{cases}$ with $\boldsymbol{e}_j^i$ being a j-dimensional unit column vector whose ith component is 1 and the rest of the components are 0;

Finally, $\boldsymbol{C}$ is a column vector of order $N-1$ and it has the form given by $\boldsymbol{C} = \begin{pmatrix} \boldsymbol{0}_{(N-2)\times 1} \\ \lambda \end{pmatrix}$.

Since the above DTMC is a finite irreducible regular Markov chain, then for any choice of the system parameters there exists stationary probabilities of the system states which are defined as below:

$$\pi_{0,j} = \lim_{t\to\infty} \Pr\{X(t) = 0, Q(t) = j\}, \quad j = 0, 1,$$
$$\pi_{i,j} = \lim_{t\to\infty} \Pr\{X(t) = i, Q(t) = j\}, \quad i = 1, 2, \ldots, T-1, \quad j = 1, \ldots, \min(i+1, N-1),$$
$$\hat{\pi}_j = \lim_{t\to\infty} \Pr\{Q(t) = j\}, \quad j = 1, 2, \ldots, K.$$

Let us enumerate probabilities $\pi_{0,j}$ and $\pi_{i,j} (i = 1, 2, \ldots, T-1)$ in the lexicographic order and form row vectors $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_i$, namely $\boldsymbol{\pi}_0 = (\pi_{0,0}, \pi_{0,1})$ and $\boldsymbol{\pi}_i = (\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,\min(i+1,N-1)})$. It is well known that the vector $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_{T-1}, \hat{\pi}_1, \hat{\pi}_2, \ldots, \hat{\pi}_K)$ is the unique solution to the following system of linear algebraic equations:

$$\begin{cases} (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_{T-1}, \hat{\pi}_1, \hat{\pi}_2, \ldots, \hat{\pi}_K)(\boldsymbol{P} - \boldsymbol{I}_\Lambda) = \boldsymbol{0}_{1\times\Lambda}, \\ (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_{T-1}, \hat{\pi}_1, \hat{\pi}_2, \ldots, \hat{\pi}_K)\boldsymbol{e}_\Lambda = 1, \end{cases} \tag{1}$$

where $\Lambda = 2 + \frac{(N+1)(N-2)}{2} + (N-1)(T-N+1) + K$, $\boldsymbol{I}_m$ stands for an identity matrix of dimension m, and $\boldsymbol{e}_m$ is a column vector of dimension m consisting of 1's. In case the vector $\boldsymbol{\pi}$ has small dimension, the finite linear system (1) with the matrix $\boldsymbol{P} - \boldsymbol{I}_\Lambda$ can be solved numerically directly. However, the direct solving is time and resource consuming and hence can only be used for the low dimensional system. In

order to efficiently solve the above system of linear algebraic equations, we use the UL-type RG-factorization, given in Li (2010), to find the solution of Eq. (1). The major advantage of RG-factorization is that it can avoid the calculation of high dimensional matrices by decomposing them into small ones. Due to the special block structure of this DTMC, it is clear to see that employing the censoring technique, we can derive a simpler expression for the UL-type RG-factorization of matrix $\boldsymbol{P} - \boldsymbol{I}_\Lambda$. Such RG-factorization has not been reported in the existing literature and it is very useful to study the discrete-time *NT*-policy queue. Here, the UL-type RG-factorization of matrix $\boldsymbol{P} - \boldsymbol{I}_\Lambda$ is given by

$$\boldsymbol{P} - \boldsymbol{I}_\Lambda = (\boldsymbol{I}_\Lambda - \boldsymbol{R}_U)\mathrm{diag}\left(\boldsymbol{U}_0, \boldsymbol{U}_1, \ldots, \boldsymbol{U}_{T-1}, \hat{\boldsymbol{U}}_1, \hat{\boldsymbol{U}}_2, \ldots, \hat{\boldsymbol{U}}_K\right)(\boldsymbol{I}_\Lambda - \boldsymbol{G}_L), \qquad (2)$$

where

$$\boldsymbol{I}_\Lambda - \boldsymbol{R}_U =$$
$$\begin{pmatrix}
\boldsymbol{I}_2 & -\boldsymbol{R}_{0,1} \\
& \boldsymbol{I}_2 & -\boldsymbol{R}_{1,2} \\
& & \boldsymbol{I}_3 & \ddots \\
& & & \ddots & \ddots \\
& & & & \boldsymbol{I}_{N-2} & -\boldsymbol{R}_{N-3,N-2} \\
& & & & & \boldsymbol{I}_{N-1} & \ddots & & & -\widetilde{\boldsymbol{R}}_{N-2,1} & -\widetilde{\boldsymbol{R}}_{N-2,2} & \cdots & -\widetilde{\boldsymbol{R}}_{N-2,N-1} & -\widetilde{\boldsymbol{R}}_{N-2,N} \\
& & & & & & \ddots & \ddots & & \vdots & \vdots & \cdots & \vdots & \vdots \\
& & & & & & & \boldsymbol{I}_{N-1} & -\boldsymbol{R}_{T-2,T-1} & -\widetilde{\boldsymbol{R}}_{T-2,1} & -\widetilde{\boldsymbol{R}}_{T-2,2} & \cdots & -\widetilde{\boldsymbol{R}}_{T-2,N-1} & -\widetilde{\boldsymbol{R}}_{T-2,N} \\
& & & & & & & & \boldsymbol{I}_{N-1} & -\widetilde{\boldsymbol{R}}_{T-1,1} & -\widetilde{\boldsymbol{R}}_{T-1,2} & \cdots & -\widetilde{\boldsymbol{R}}_{T-1,N-1} & -\widetilde{\boldsymbol{R}}_{T-1,N} \\
& & & & & & & & & \boldsymbol{I}_1 & -\hat{\boldsymbol{R}}_{1,2} \\
& & & & & & & & & & \boldsymbol{I}_1 & \ddots \\
& & & & & & & & & & & \ddots & \ddots \\
& & & & & & & & & & & & \boldsymbol{I}_1 & -\hat{\boldsymbol{R}}_{N-1,N} \\
& & & & & & & & & & & & & \boldsymbol{I}_1 & -\hat{\boldsymbol{R}}_{N,N+1} \\
& & & & & & & & & & & & & & \boldsymbol{I}_1 & \ddots \\
& & & & & & & & & & & & & & & \ddots & \ddots \\
& & & & & & & & & & & & & & & & \boldsymbol{I}_1 & -\hat{\boldsymbol{R}}_{K-1,K} \\
& & & & & & & & & & & & & & & & & \boldsymbol{I}_1
\end{pmatrix},$$

and

$$\boldsymbol{I}_\Lambda - \boldsymbol{G}_L = \begin{pmatrix}
\boldsymbol{I}_2 \\
-\boldsymbol{G}_{1,0} & \boldsymbol{I}_2 \\
-\boldsymbol{G}_{2,0} & & \boldsymbol{I}_3 \\
\vdots & & & \ddots \\
-\boldsymbol{G}_{N-3,0} & & & & \boldsymbol{I}_{N-2} \\
-\boldsymbol{G}_{N-2,0} & & & & & \boldsymbol{I}_{N-1} \\
\vdots & & & & & & \ddots \\
-\boldsymbol{G}_{T-1,0} & & & & & & & \boldsymbol{I}_{N-1} \\
-\bar{\boldsymbol{G}}_{1,0} & & & & & & & & \boldsymbol{I}_1 \\
& & & & & & & & \hat{\boldsymbol{G}}_{2,1} & \boldsymbol{I}_1 \\
& & & & & & & & & \ddots & \ddots \\
& & & & & & & & & & \ddots & \boldsymbol{I}_1 \\
& & & & & & & & & & & \hat{\boldsymbol{G}}_{K-1,K-2} & \boldsymbol{I}_1 \\
& & & & & & & & & & & & \hat{\boldsymbol{G}}_{K,K-1} & \boldsymbol{I}_1
\end{pmatrix}.$$

Compare the right-hand side of Eq. (2) and matrix $\boldsymbol{P}-\boldsymbol{I}_{\Lambda}$, we may find that the matrices $\boldsymbol{U}_i, \hat{\boldsymbol{U}}_i, \hat{\boldsymbol{G}}_{i,i-1}, \bar{\boldsymbol{G}}_{1,0}, \boldsymbol{G}_{i,0}, \hat{\boldsymbol{R}}_{i,i+1}, \widetilde{\boldsymbol{R}}_{i,N}, \widetilde{\boldsymbol{R}}_{T-1,j}, \widetilde{\boldsymbol{R}}_{i,j}$ and $\boldsymbol{R}_{i,i+1}$ can be calculated through the following backward recursion:

$$
\begin{aligned}
&\hat{\boldsymbol{U}}_K = (\mu\lambda + \bar{\mu}) - \boldsymbol{I}_1, \quad \hat{\boldsymbol{G}}_{K,K-1} = -\hat{\boldsymbol{U}}_K^{-1}\big(\mu\bar{\lambda}\big), \quad \hat{\boldsymbol{R}}_{K,K-1} = -(\bar{\mu}\lambda)\hat{\boldsymbol{U}}_K^{-1},\\
&\hat{\boldsymbol{U}}_i = \big(\mu\lambda + \bar{\mu}\bar{\lambda}\big) - \boldsymbol{I}_1 - \hat{\boldsymbol{R}}_{i,i+1}\hat{\boldsymbol{U}}_{i+1}\hat{\boldsymbol{G}}_{i+1,i}, \quad i = 1,2,\ldots,K-1,\\
&\boldsymbol{U}_i = -\boldsymbol{I}_{\min(i+1,N-1)}, \quad i = 1,2,\ldots,T-1,\\
&\boldsymbol{U}_0 = \begin{pmatrix} \boldsymbol{H}_1 \\ \boldsymbol{0}_{1\times 2} \end{pmatrix} - \boldsymbol{I}_2 - \boldsymbol{R}_{0,1}\boldsymbol{U}_1\boldsymbol{G}_{1,0},\\
&\hat{\boldsymbol{G}}_{i,i-1} = -\hat{\boldsymbol{U}}_i^{-1}\big(\mu\bar{\lambda}\big), \quad i = 2,3,\ldots,K-1, \quad \bar{\boldsymbol{G}}_{1,0} = -\hat{\boldsymbol{U}}_1^{-1}\begin{pmatrix} \mu\bar{\lambda} & 0 \end{pmatrix},\\
&\boldsymbol{G}_{T-1,0} = \boldsymbol{U}_{T-1}^{-1}\big(\tilde{\boldsymbol{R}}_{T-1,1}\hat{\boldsymbol{U}}_1\bar{\boldsymbol{G}}_{1,0}\big),\\
&\boldsymbol{G}_{i,0} = \boldsymbol{U}_i^{-1}\big(\boldsymbol{R}_{i,i+1}\boldsymbol{U}_{i+1}\boldsymbol{G}_{i+1,0} + \tilde{\boldsymbol{R}}_{i,1}\hat{\boldsymbol{U}}_1\bar{\boldsymbol{G}}_{1,0}\big), \quad i = 1,2,\ldots,T-2,\\
&\hat{\boldsymbol{R}}_{i,i+1} = -(\bar{\mu}\lambda)\hat{\boldsymbol{U}}_{i+1}^{-1}, \quad i = 1,2,\ldots,K-2,\\
&\widetilde{\boldsymbol{R}}_{i,N} = -\boldsymbol{C}\hat{\boldsymbol{U}}_N^{-1}, \quad i = N-2,N-1,\ldots,T-1,\\
&\widetilde{\boldsymbol{R}}_{T-1,j} = \big(\tilde{\boldsymbol{R}}_{T-1,j+1}\hat{\boldsymbol{U}}_{j+1}\hat{\boldsymbol{G}}_{j+1,j} - \boldsymbol{D}_j\big)\hat{\boldsymbol{U}}_j^{-1}, \quad j = 1,2,\ldots,N-1,\\
&\widetilde{\boldsymbol{R}}_{i,j} = \big(\tilde{\boldsymbol{R}}_{i,j+1}\hat{\boldsymbol{U}}_{j+1}\hat{\boldsymbol{G}}_{j+1,j}\big)\hat{\boldsymbol{U}}_j^{-1}, \quad i = N-2,N-1,\ldots,T-2, \quad j = 1,2,\ldots,N-1,\\
&\boldsymbol{R}_{i,i+1} = -\boldsymbol{H}_{i+1}\boldsymbol{U}_{i+1}^{-1}, \quad i = 1,2,\ldots,T-2, \quad \boldsymbol{R}_{0,1} = -\begin{pmatrix} \boldsymbol{0}_{1\times 2} \\ \boldsymbol{H}_1 \end{pmatrix}\boldsymbol{U}_1^{-1}.
\end{aligned}
$$

Refer to the Subsection 2.6.3 of Li (2010), the stationary probability vectors of the DTMC are given by

$$
\begin{cases}
\boldsymbol{\pi}_0 = \tau\boldsymbol{x}_0, & \\
\boldsymbol{\pi}_i = \boldsymbol{\pi}_{i-1}\boldsymbol{R}_{i-1,i}, & i = 1,2,\ldots,T-1,\\
\hat{\pi}_1 = \sum\limits_{j=N-2}^{T-1}\boldsymbol{\pi}_j\tilde{\boldsymbol{R}}_{j,1}, & \\
\hat{\pi}_i = \sum\limits_{j=N-2}^{T-1}\boldsymbol{\pi}_j\tilde{\boldsymbol{R}}_{j,i} + \hat{\pi}_{i-1}\hat{\boldsymbol{R}}_{i-1,i}, & i = 2,3,\ldots,N,\\
\hat{\pi}_i = \hat{\pi}_{i-1}\hat{\boldsymbol{R}}_{i-1,i}, & i = N+1,\ldots,K-1,K,
\end{cases}
$$

where $\boldsymbol{x}_0$ is the stationary probability vector of the censored Markov chain $\boldsymbol{U}_0$ to level 0 and the scalar τ is uniquely determined by $\boldsymbol{\pi}_0\boldsymbol{e}_2 + \sum_{i=1}^{T-1}\boldsymbol{\pi}_i\boldsymbol{e}_{\min(i+1,N-1)} + \sum_{i=1}^{K}\hat{\pi}_i = 1$. Once the stationary probability vectors are calculated, some important performance measures will be given in terms of $\boldsymbol{\pi}_i$ and $\hat{\pi}_i$. Here, we demonstrate several main performance characteristics of the system to bring out the qualitative aspects of the model under study. These are listed below along with their formulas for computation.

- The mean number of customers in the system:

$$\mathrm{E}[L_{NT}] = \pi_{0,1} + \sum_{j=1}^{N-1} \sum_{i=\max(j-1,1)}^{T-1} j\boldsymbol{\pi}_i \boldsymbol{e}^j_{\min(N-1,i+1)} + \sum_{j=1}^{K} j\hat{\pi}_j.$$

- The blocking probability of an arbitrary customer: $P_{\text{block}} = \hat{\pi}_K$.
- The mean sojourn time in the system: $W_s = \frac{\mathrm{E}[L_{NT}]}{\lambda(1-P_{\text{block}})}$.
- The probability that the server is idle: $P_{\text{idle}} = \boldsymbol{\pi}_0 \boldsymbol{e}_2 + \sum_{i=1}^{T-1} \boldsymbol{\pi}_i \boldsymbol{e}_{\min(N-1,i+1)}$.
- The probability that the server is busy: $P_{\text{busy}} = \sum_{i=1}^{K} \hat{\pi}_i$.

4 Expected length of a regeneration cycle

For evaluating the long-run average cost per unit time, we also need to derive the expected interval of a regeneration cycle. In the present model, a regeneration cycle consists of an idle period and a busy period. When all the customers in the system are served and there are no customers waiting in the queue, an idle period starts. It stops when the server becomes reactivated either because of situation (i) or (ii) as described in Sect. 2. Let I_{NT} be the length of the idle period. Denote $I_{NT}(z)$ by the probability generating function (PGF) of I_{NT}. According to the model assumptions, we can show that

$$\begin{aligned}
I_{NT}(z) &= \mathrm{E}\left[z^{I_{NT}}\right] = \mathrm{E}\left[z^{A_1+Y_{NT}}\right] = \mathrm{E}\left[z^{A_1}\right]\left(\mathrm{E}\left[z^{Y_{NT}}\mathbf{1}_{\left\{\sum_{i=2}^{N} A_i \le T\right\}}\right] + \mathrm{E}\left[z^{Y_{NT}}\mathbf{1}_{\left\{\sum_{i=2}^{N} A_i > T\right\}}\right]\right) \\
&= \mathrm{E}\left[z^{A_1}\right]\left(\mathrm{E}\left[z^{\sum_{i=2}^{N} A_i}\right]\Pr\left\{\sum_{i=2}^{N} A_i \le T\right\} + \mathrm{E}\left[z^{T}\right]\Pr\left\{\sum_{i=2}^{N} A_i > T\right\}\right) \\
&= \mathrm{E}\left[z^{A_1}\right]\left[\sum_{n=N-1}^{T} \mathrm{E}[z^n]\Pr\left\{\sum_{i=2}^{N} A_i = n\right\} + E\left[z^{T}\right] \quad 1 - \Pr\left\{\sum_{i=2}^{N} A_i \le T\right\}\right)\right] \\
&= \frac{\lambda z}{1-\bar{\lambda} z} \quad \sum_{n=N-1}^{T} z^n \boldsymbol{\alpha}\boldsymbol{H}^{n-1}\boldsymbol{H}_0 + z^T\left[1 - \boldsymbol{\alpha}\boldsymbol{H}^{N-2}(\boldsymbol{I}_{N-1} - \boldsymbol{H})^{-1}(\boldsymbol{I}_{N-1} - \boldsymbol{H}^{T-N+2})\boldsymbol{H}_0\right]\Bigg),
\end{aligned}$$

where $\mathbf{1}_{\{C\}}$ is an indicator function such that $\mathbf{1}_{\{C\}} = \begin{cases} 1, & \text{if event C occurs} \\ 0, & \text{if event C does not occur} \end{cases}$, $\boldsymbol{\alpha} = (1, \underbrace{0, \ldots, 0}_{N-2})$, $\boldsymbol{H}_0 = (\underbrace{0, \ldots, 0}_{N-2}, \lambda)^{\top}$ and the superscript '$\top$' denotes the transpose of a matrix or a vector. From the PGF of I_{NT} we obtain

$$\mathrm{E}[I_{NT}] = \frac{1}{\lambda} + \sum_{n=N-1}^{T} n\boldsymbol{\alpha}\boldsymbol{H}^{n-1}\boldsymbol{H}_0 + T\Big[1 - \boldsymbol{\alpha}\boldsymbol{H}^{N-2}(\boldsymbol{I}_{N-1} - \boldsymbol{H})^{-1}\big(\boldsymbol{I}_{N-1} - \boldsymbol{H}^{T-N+2}\big)\boldsymbol{H}_0\Big]. \tag{3}$$

As for the *NT*-policy queue, the busy period B_{NT} is defined to be the time elapsed from the server's return to the system until the queue becomes empty again and the next vacation begins. Because the server's busy period might be initiated by one or more customers in such system, deriving the expected length of busy period should be implemented by means of the analysis of the busy period that starts with i customers, namely the so-called i-busy period B_i, where the subscript i represents the number of arrivals during the vacation time. Actually, the expected length of i-busy period is only related to the following three factors: (1) customer arrival rate and service rate, (2) the number of customers initially present at the queue, (3) buffer capacity, and it has nothing to do with the service control policy. Hence, for solving $\mathrm{E}[B_i]$, we first consider the *Geo*/*Geo*/1/K queue without *NT*-policy. Let $B_i(z), i = 1, 2, \ldots, K$ denote the PGF of the i-busy period. By employing first-step analysis and the memoryless property of geometric distribution we can get the equations governing the dynamic of the generating functions $B_i(z)$:

$$\begin{aligned} B_i(z) &= \mathrm{E}\big[z^{B_i}\big] = \mathrm{E}\big[z^{B_i}\mathbf{1}_{\{A_{i+1}<S\}}\big] + \mathrm{E}\big[z^{B_i}\mathbf{1}_{\{A_{i+1}=S\}}\big] + \mathrm{E}\big[z^{B_i}\mathbf{1}_{\{A_{i+1}>S\}}\big] \\ &= \mathrm{E}\Big[z^{\min(A_{i+1},S)}\Big]\left(\mathrm{E}\big[z^{B_{i+1}}\big]\frac{\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} + \mathrm{E}\big[z^{B_i}\big]\frac{\lambda\mu}{1-\bar{\lambda}\bar{\mu}} + \mathrm{E}\big[z^{B_{i-1}}\big]\frac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}\right) \\ &= \frac{(1-\bar{\lambda}\bar{\mu})z}{1-\bar{\lambda}\bar{\mu}z}\left(B_{i+1}(z)\frac{\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} + B_i(z)\frac{\lambda\mu}{1-\bar{\lambda}\bar{\mu}} + B_{i-1}(z)\frac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}\right) \\ &= \frac{z}{1-\bar{\lambda}\bar{\mu}z}\big(B_{i+1}(z)\lambda\bar{\mu} + B_i(z)\lambda\mu + B_{i-1}(z)\bar{\lambda}\mu\big), \quad i = 1, 2, \ldots, K-1, \end{aligned} \tag{4}$$

$$\begin{aligned} B_K(z) &= \mathrm{E}\big[z^{B_K}\big] = \mathrm{E}\big[z^{B_K}\mathbf{1}_{\{A_{K+1}\le S\}}\big] + \mathrm{E}\big[z^{B_{K-1}}\mathbf{1}_{\{A_{K+1}>S\}}\big] \\ &= \mathrm{E}\Big[z^{\min(A_{K+1},S)}\Big]\left(\mathrm{E}\big[z^{B_K}\big]\frac{\lambda}{1-\bar{\lambda}\bar{\mu}} + \mathrm{E}\big[z^{B_{K-1}}\big]\frac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}\right) \\ &= \frac{(1-\bar{\lambda}\bar{\mu})z}{1-\bar{\lambda}\bar{\mu}z}\left(B_K(z)\frac{\lambda}{1-\bar{\lambda}\bar{\mu}} + B_{K-1}(z)\frac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}\right) \\ &= \frac{z}{1-\bar{\lambda}\bar{\mu}z}\big(B_K(z)\lambda + B_{K-1}(z)\bar{\lambda}\mu\big). \end{aligned} \tag{5}$$

Here, for mathematical convenience, the busy period initiated by 0 customer is defined to be of zero length. Thus, in Eq. (4), when $i = 1$, $B_0(z) = \mathrm{E}[z^{B_0}] = 1$. Indeed, differentiating both sides of Eqs. (4) and (5) with respect to z and setting $z = 1$ yields the following Eq. (6).

$$\begin{cases} \mathrm{E}[B_1]\dfrac{\bar{\lambda}\mu+\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}}+\mathrm{E}[B_2]\dfrac{-\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}}=\dfrac{1}{1-\bar{\lambda}\bar{\mu}}, \\ \mathrm{E}[B_{i-1}]\dfrac{-\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}+\mathrm{E}[B_i]\dfrac{\bar{\lambda}\mu+\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}}+\mathrm{E}[B_{i+1}]\dfrac{-\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}}=\dfrac{1}{1-\bar{\lambda}\bar{\mu}}, \quad i=2,3,\ldots,K-1, \\ \mathrm{E}[B_{K-1}]\dfrac{-\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}+\mathrm{E}[B_K]\dfrac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}=\dfrac{1}{1-\bar{\lambda}\bar{\mu}}. \end{cases} \tag{6}$$

It can be verified that Eq. (6) can be expressed in matrix form as

$$\boldsymbol{M}(\mathrm{E}[B_1],\mathrm{E}[B_2],\ldots,\mathrm{E}[B_K])^{\top}=\frac{1}{1-\bar{\lambda}\bar{\mu}}\boldsymbol{e}_K, \tag{7}$$

in which

$$\boldsymbol{M}=\begin{pmatrix} \dfrac{\bar{\lambda}\mu+\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} & \dfrac{-\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} & 0 & \cdots & 0 & 0 \\ \dfrac{-\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}} & \dfrac{\bar{\lambda}\mu+\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} & \dfrac{-\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} & \cdots & 0 & 0 \\ 0 & \dfrac{-\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}} & \dfrac{\bar{\lambda}\mu+\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \dfrac{\bar{\lambda}\mu+\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} & \dfrac{-\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}} \\ 0 & 0 & 0 & \cdots & \dfrac{-\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}} & \dfrac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}} \end{pmatrix}_{K\times K}.$$

Using Cramer's rule and performing some appropriate elementary column(row) transformations on matrices $\boldsymbol{M}$ and $\boldsymbol{M}_1$, the expected length of the busy period that starts with one customer is explicitly given as

$$\mathrm{E}[B_1]=\frac{\det(\boldsymbol{M}_1)}{\det(\boldsymbol{M})}=\frac{1}{\bar{\lambda}\mu}\sum_{r=1}^{K}\rho^{r-1}=\frac{1-\rho^K}{\mu-\lambda},$$

where $\boldsymbol{M}_1$ is derived from the matrix $\boldsymbol{M}$ by replacing the first column with the vector $\frac{1}{1-\bar{\lambda}\bar{\mu}}\boldsymbol{e}_K$, and $\rho=\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}$. Once $\mathrm{E}[B_1]$ is obtained, the explicit expression for the expected length of i-busy period ($i=2,3,\ldots,K$) can be determined by the following way. Since $\mathrm{E}[B_0]=0$, rearranging the terms in Eq. (6), we can arrive at

$$\frac{\lambda\bar{\mu}}{1-\bar{\lambda}\bar{\mu}}(\mathrm{E}[B_{i+1}]-\mathrm{E}[B_i])=-\frac{1}{1-\bar{\lambda}\bar{\mu}}+\frac{\bar{\lambda}\mu}{1-\bar{\lambda}\bar{\mu}}(\mathrm{E}[B_i]-\mathrm{E}[B_{i-1}]), \quad i=1,\ldots,K-1. \tag{8}$$

Let $\Phi_i=\mathrm{E}[B_{i+1}]-\mathrm{E}[B_i], i=0,1,\ldots,K-1$, the above Eq. (8) can be rewritten as

$$\Phi_i=-\frac{1}{\lambda\bar{\mu}}+\frac{1}{\rho}\Phi_{i-1}, \quad i=1,\ldots,K-1. \tag{9}$$

Because $\Phi_0 = \mathrm{E}[B_1]$, we can iteratively obtain the following expression for Φ_i from Eq. (9),

$$\Phi_i = \frac{-1}{\lambda\bar{\mu}}\sum_{r=1}^{i}\frac{1}{\rho^{r-1}} + \frac{\Phi_0}{\rho^i} = \frac{-1}{\lambda\bar{\mu}}\sum_{r=1}^{i}\frac{1}{\rho^{r-1}} + \frac{\mathrm{E}[B_1]}{\rho^i}, \quad i = 1, 2, \ldots, K-1. \tag{10}$$

Substituting $\mathrm{E}[B_1] = \frac{1}{\bar{\lambda}\mu}\sum_{r=1}^{K}\rho^{r-1}$ into Eq. (10) leads to

$$\Phi_i = \frac{1}{\bar{\lambda}\mu}\frac{1-\rho^{K-i}}{1-\rho}, \quad i = 1, 2, \ldots, K-1.$$

Thus, a simple iteration formula based on initial value $\mathrm{E}[\mathrm{B}_1]$ can be expressed as

$$\mathrm{E}[B_{i+1}] = \mathrm{E}[B_i] + \frac{1}{\bar{\lambda}\mu}\frac{1-\rho^{K-i}}{1-\rho}, \quad i = 1, 2, \ldots, K-1. \tag{11}$$

From Eq. (11), the explicit expression of $\mathrm{E}[\mathrm{B_i}](\mathrm{i} = 2, 3, \ldots, \mathrm{K})$ is given by

$$\mathrm{E}[B_i] = \frac{1}{\mu-\lambda}\left[i - \frac{\rho^{K-i+1}(1-\rho^i)}{1-\rho}\right]. \tag{12}$$

Then, let us consider the number of waiting customers when the server is activated, which is denoted by J. From the NT-policy characteristics, the probability mass function of J is given as follows

$$\Pr\{J=l\} = \begin{cases} \binom{T}{l-1}\lambda^{l-1}\bar{\lambda}^{T-l+1}, & l = 1, 2, \ldots, N-1, \\ \lambda^{N-1}\sum\limits_{r=N-2}^{T-1}\binom{r}{N-2}\bar{\lambda}^{r-N+2} = \boldsymbol{\alpha}\boldsymbol{H}^{N-2}(\boldsymbol{I}_{N-1}-\boldsymbol{H})^{-1}(\boldsymbol{I}_{N-1}-\boldsymbol{H}^{T-N+2})\boldsymbol{H}_0, & l = N. \end{cases}$$

Using the law of total expectation, we have

$$\begin{aligned} \mathrm{E}[B_{\mathrm{NT}}] &= \sum_{i=1}^{N}\mathrm{E}[B_i]\Pr\{J=i\} \\ &= \sum_{i=1}^{N-1}\binom{T}{i-1}\lambda^{i-1}\bar{\lambda}^{T-i+1}\left\{\frac{1}{\mu-\lambda}\left[i - \frac{\rho^{K-i+1}(1-\rho^i)}{1-\rho}\right]\right\} \\ &\quad + \boldsymbol{\alpha}\boldsymbol{H}^{N-2}(\boldsymbol{I}_{N-1}-\boldsymbol{H})^{-1}(\boldsymbol{I}_{N-1}-\boldsymbol{H}^{T-N+2})\boldsymbol{H}_0\left\{\frac{1}{\mu-\lambda}\left[N - \frac{\rho^{K-N+1}(1-\rho^N)}{1-\rho}\right]\right\} \\ &= \frac{1-\rho^K}{\mu-\lambda} + \frac{1}{\mu-\lambda}\left[T\lambda - \sum_{i=N-1}^{T} i\binom{T}{i}\lambda^i\bar{\lambda}^{T-i}\right] \\ &\quad + \frac{\boldsymbol{\alpha}\boldsymbol{H}^{N-2}(\boldsymbol{I}_{N-1}-\boldsymbol{H})^{-1}(\boldsymbol{I}_{N-1}-\boldsymbol{H}^{T-N+2})\boldsymbol{H}_0}{\mu-\lambda}\left[N-1-\frac{\rho^K(1-\rho^{N-1})}{\rho^{N-1}(1-\rho)}\right] \\ &\quad - \frac{1}{\mu-\lambda}\sum_{i=1}^{N-1}\binom{T}{i-1}\lambda^{i-1}\bar{\lambda}^{T-i+1}\frac{\rho^K(1-\rho^{i-1})}{\rho^{i-1}(1-\rho)}. \end{aligned} \tag{13}$$

It is worth noting that $\frac{1-\rho^K}{\mu-\lambda}$ is the expected length of the busy period of the standard *Geo*/*Geo*/1/*K* system. Therefore, Eq. (13) has demonstrated the stochastic decomposition structure of the busy period in *Geo*/*Geo*/1/*K* queue with *NT*-policy.

Define a regeneration cycle as the time elapsed between two consecutive epochs at which the system becomes empty. Let $\mathrm{E}[B_C]$ be the expected regeneration cycle. Then, from Eqs. (3) and (13) we have that $\mathrm{E}[B_C] = \mathrm{E}[B_{NT}] + \mathrm{E}[I_{NT}]$. Just because the explicit expressions of $\mathrm{E}[B_{NT}]$ and $\mathrm{E}[I_{NT}]$ are slightly cumbersome to write, we do not indent to substitute $\mathrm{E}[B_{NT}]$ and $\mathrm{E}[I_{NT}]$ into the above formula. In addition, it is particularly worth mentioning that there is an effective way to validate the correctness of the theoretical results presented above. If the analysis about the regeneration cycle is correct, then the following equality must hold:

$$P_{\text{idle}} = \boldsymbol{\pi}_0 \boldsymbol{e}_2 + \sum_{i=1}^{T-1} \boldsymbol{\pi}_i \boldsymbol{e}_{\min(N-1,i+1)} = \frac{\mathrm{E}[B_{NT}]}{\mathrm{E}[B_C]}. \tag{14}$$

For the case with $N = 5, T = 9, K = 12, \mu = 0.24$ and $\lambda = 0.21$, by numerical computation, we find that the above equality is indeed true, namely

$$\boldsymbol{\pi}_0 \boldsymbol{e}_2 + \sum_{i=1}^{8} \boldsymbol{\pi}_i \boldsymbol{e}_{\min(4,i+1)} = \frac{\mathrm{E}[B_{NT}]}{\mathrm{E}[B_C]} = 0.145006531946 23.$$

So, Eq. (14) often acts as an internal check on our results, which is very useful in debugging numerical programs and checking accuracy for computations.

Remark 1 We note that several interesting queueing systems can be viewed as special cases of this model.

Case 1. As $T \to \infty$, the current model can be reduced to the ordinary *Geo*/*Geo*/1/*K* queue with *N* policy. Thus, the regeneration cycle $\mathrm{E}\left[B_C^{N-\text{policy}}\right]$ can be easily obtained from Eq. (12)

$$\mathrm{E}\left[B_C^{N-\text{policy}}\right] = \frac{N}{\lambda} + \frac{1}{\mu-\lambda}\left[N - \frac{\rho^{K-N+1}(1-\rho^N)}{1-\rho}\right].$$

Case 2. If we put $N = \infty$, hence the *Geo*/*Geo*/1/*K* queue with *NT*-policy reduces to the *Geo*/*Geo*/1/*K* queue with simple threshold *T*-policy. Thus, utilizing Eq. (12), the regeneration cycle $\mathrm{E}\left[B_C^{T-\text{policy}}\right]$ can be derived as follows

$$\mathrm{E}\left[B_C^{T-\text{policy}}\right] = \left(\frac{1}{\lambda} + T\right) + \frac{1-\rho^K}{\mu-\lambda} + \frac{T\lambda}{\mu-\lambda} - \frac{1}{\mu-\lambda}\sum_{i=1}^{T}\binom{T}{i}\lambda^i \bar{\lambda}^{T-i}\frac{1-\rho^i}{\rho^i(1-\rho)}.$$

5 The jointly optimal threshold (N^*, T^*) for the average cost criterion

The research motivation for the *NT*-policy queue comes from the operating cost consideration. To maintain the normal operation of the system, human resources salary, power consumption costs, etc. are paid when the server is available and are not charged when the server is unavailable. These costs may be interpreted as the additional cost incurred by reopening the service facility. It provides an incentive for turning off the server when no customer presents in the queue. On the other hand, when the customers are viewed as machines or taxis waiting for repair, each time unit they spend in the queue or in service represents lost operating time. One way to model this effect is to create for each customer a holding cost that is an increasing function of the mean queue length. This holding cost provides an incentive for turning on the server before there are too many customers waiting in line. In this section, we will find the values of N and T which produce the optimal stationary *NT*-policy such that the expected operating cost of the system is minimized. We assume that a start-up cost of c_s is incurred each time the server is turned on and that a linear holding cost of c_h per unit time is incurred for each customer present in the system. Furthermore, since the buffer capacity of this system is finite, the overflowing customers will be rejected from the system, which will directly result in revenue loss. Hence, a fixed cost c_l for every lost customer when the system is blocked should be taken into account in the cost structure. Thus, the long-run average operating cost per unit time under *NT*-policy with buffer capacity K can be written as

$$TC(N,T) = c_h \mathrm{E}[L_{NT}] + c_s \frac{1}{\mathrm{E}[B_c]} + c_l P_{\mathrm{block}} \lambda. \tag{15}$$

Substitution of $\mathrm{E}[L_{NT}]$, P_{block} and $\mathrm{E}[B_c]$ into Eq. (15), the cost function $TC(N, T)$ is too complicated to be shown here. However, the good feature about finding the optimal policy is that both decision parameters N and T are in a finite value range with the relation $1 < N \leq T < K$. Based on the above facts, the direct-search algorithm can be used to find the optimal joint solution (N^*, T^*) so as to minimize the cost function. If we write the computer program in Matlab software, the computational time is acceptable for reasonable K values ($\leq$50). For example, when $K = 40$, it takes reasonable amount of time on a personal computer having Intel Core i5 processor at 2.6 GHz with 4 GB DDR3 RAM, and this simple direct-search algorithm over finite space is terminated after achieving a global optimum.

To demonstrate the tractability of the suggested method, we perform the numerical experiment by considering the following system and cost parameters: $K = 21, \mu = 0.7, \lambda = 0.35, c_h = 10, c_s = 1000, c_l = 10{,}000$, and vary the threshold value N from 3 to $T - 1$, and T ranges from 6 to 21. A 3D trajectory of the unit time operating cost is shown in Fig. 3. As can be seen in Fig. 3, it convinces us that the unit time cost function is convex, and minimum expected cost per unit time of 61.0119 can be obtained at the optimal solution $(N^*, T^*) = (6, 18)$.

Furthermore, to examine the effects of different parameters on the optimum joint threshold N^* and T^*, a numerical illustration of sensitivity analysis based on

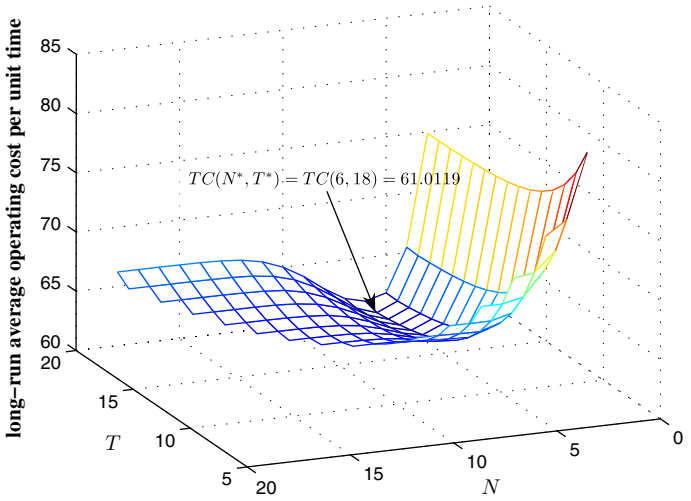

Fig. 3 A 3D trajectory of the long-run average operating cost per unit time under (N, T)-policy

Table 1 The joint optimum threshold values (N^*, T^*) and its long-run average operating cost $TC(N^*, T^*)$ for different values of λ and μ

(λ, μ)	(0.4, 0.7)	(0.5, 0.8)	(0.6, 0.9)
Case 1:			
(N^*, T^*)	(6, 42)	(7, 32)	(7, 26)
$TC(N^*, T^*)$	67.2857	70.4762	72.2857
Case 2:			
(N^*, T^*)	(4, 38)	(4, 25)	(5, 20)
$TC(N^*, T^*)$	88.8571	93.5417	96.0000
Case 3:			
(N^*, T^*)	(6, 41)	(6, 32)	(6, 25)
$TC(N^*, T^*)$	61.5714	64.5833	66.3333

changes in specific values of system parameters is also conducted in the following different cost cases:

Case 1: $c_s = 1200, c_h = 10$ and $c_l = 1000$;

Case 2: $c_s = 1000, c_h = 20$ and $c_l = 1000$;

Case 3: $c_s = 1000, c_h = 10$ and $c_l = 10{,}000$,

where we fix the system capacity as $K = 60$, vary T from 6 to 60, and vary N from 3 to $T - 1$. The joint optimum threshold values and the minimum long-run average operating cost for the above three cost cases are summarized in Table 1 for various values of (λ, μ). All the calculations have been done on the Matlab software package and all the dates are reported here in four decimal places. From Table 1, we observe that (i) N^* and T^* seem very sensitive to the changes of c_h, but they are not significantly affected by the changes of c_l; (ii) Table 1 further indicates that $TC(N^*, T^*)$ increases evidently with the increase of c_h. Moreover, the effect of

changes of c_s on the long-run average operating cost are less significant than c_h; (iii) The higher the ratio of λ and μ is, the larger the value of $TC(N^*, T^*)$ is. Since the main purpose of this paper is to describe how to solve the performance measures related to the long-run average operating cost, numerical tests are not very wide enough to draw useful information regarding the characteristics of $TC(N, T)$. But we believe that the system and cost parameters may play important roles in deciding the shape of $TC(N, T)$. That is to say, $TC(N, T)$ is not always a convex function for any choice of system and cost parameters.

6 Conclusions

In this study, steady-state analysis results have been presented for a controllable discrete-time finite-buffer queue in which the server applies a bicriterion *NT*-policy during his vacation time. We take some more efficient approaches to get these stationary results. On the one hand, based on the algorithmic method proposed by Li (2010), a simpler UL-type RG-factorization has been employed to find the distribution of the queue length at an arbitrary epoch. On the other hand, in order to develop an optimal management policy for the system, the expected length of the regeneration cycle is investigated by using an extended definition of the busy period that might be initiated with multiple customers. Furthermore, through some nonroutine algebraic manipulations, the explicit closed form of the busy period for finite-buffer *NT*-policy queue is firstly reported, and the existence of the stochastic decomposition property of the busy period is also demonstrated. It may be remarked here that the techniques for busy period analysis adopted in our paper can be used to analyze other complex queueing models such as the finite-buffer $\min(N, V)$- or (N, p)-policy queue with single and multiple vacations (see Wu et al. (2014) and Feinberg and Kim (1996)). These are possible extensions and suggested directions for our future research.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. This research was partially supported by grant from NSERC DAS programs, National Natural Science Foundation of China (Nos. 71301111, 71171138, 71402072) and the FSUSE (No.2012RC23).

References

Alfa AS, Frigui I (1996) Discrete NT-policy single server queue with Markovian arrival process and phase type service. Eur J Oper Res 88:599–613

Alfa AS, Li W (2000) Optimal (N, T)-policy for M/G/1 system with cost structures. Perform Eval 42:265–277

Balachandran KR (1973) Control policies for a single server system. Manag Sci 19:1013–1018

Doganata YN (1990) NT-vacation policy for M/G/1 queue with starter. In: Arikan E (ed) Communication, control, and signal processing. Elsevier Science Publishers, Amsterdam, pp 1663–1669

Doshi BT (1986) Queueing systems with vacations-A survey. Queueing Syst 1:29–66

Feyaerts B, Vuyst SD, Wittevrongel S, Bruneel H (2010) Analysis of a discrete-time queueing system with an NT-policy. Lect Notes Comput 6148:29–43

Feyaerts B, Vuyst SD, Bruneel H, Wittevrongel S (2014) The impact of the NT-policy on the behaviour of a discrete-time queue with general service times. J Ind Manag Optim 10:131–149
Feinberg EA, Kim DJ (1996) Bicriterion optimization of an M/G/1 queue with a removable server. Probab Eng Inf Sci 10:57–73
Gakis KG, Rhee HK, Sivazlian BD (1995) Distributions and first moments of the busy and idle periods in controllable M/G/1 queueing models with simple and dyadic policies. Stoch Anal Appl 13:47–81
Gupta SM (1995) Interrelationship between controlling arrival and service in queueing systems. Comput Oper Res 22:1005–1014
Heyman DP (1977) T-policy for the M/G/1 queue. Manag Sci 23:775–778
Hur S, Kim J, Kang C (2003) An analysis of the M/G/1 system with N and T policy. Appl Math Model 27:665–675
Jiang FC, Huang DC, Yang CT, Lin CH, Wang KH (2010) Design strategy for optimizing power consumption of sensor node with Min(N, T) policy M/G/1 queuing models. Int J Commun Syst 25:652–671
Ke JC (2005) Modified T vacation policy for an M/G/1 queueing system with an un-reliable server and startup. Math Comput Model 41:1267–1277
Ke JC (2006a) On M/G/1 system under NT policies with breakdowns, startup and closedown. Appl Math Model 30:49–66
Ke JC (2006b) Optimal NT policies for M/G/1 system with a startup and unreliable server. Comput Ind Eng 50:248–262
Li QL (2010) Constructive computation in stochastic models with applications: the RG-factorizations. Springer, New York
Li W, Alfa AS (2000) Optimal policies for M/M/m queue with two different kinds of (N, T)-policies. Naval Res Logist 47:240–258
Pacheco A, Ribeiro H (2008) Moments of the duration of busy periods of M^X/G/1/n systems. Probab Eng Inf Sci 22:347–354
Ross SM (1996) Stochastic processes, 2nd edn. Wiley, New York
Tadj L (2003) On an M/G/1 quorum queueing system under T-policy. J Oper Res Soc 54:466–471
Wu WQ, Tang YH, Yu MM (2014) Analysis of an M/G/1 queue with multiple vacations, N-policy, unreliable service station and repair facility failures. Int J Supply Oper Manag 1:1–19
Yadin M, Naor P (1963) Queueing systems with a removable service station. Oper Res Q 14:393–405
Zhang ZG, Tadj L, Bounkhel M (2011) Cost evaluation in M/G/1 queue with T-policy revisited, technical note. Eur J Oper Res 214:814–817