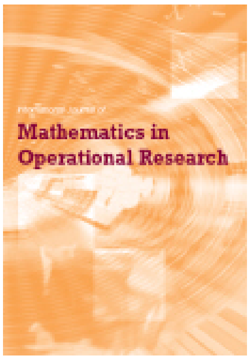


International Journal of Mathematics in Operational Research

ISSN online: 1757-5869 - ISSN print: 1757-5850
https://www.inderscience.com/ijmor

Simulation of batch service lateness queues with multiple vacations and two Bernoulli catastrophes using the developed 'goLHS' generator

Anfal Rezgui, Hafida Saggou, Megdouda Ourbih-Tari, Meriem Boubalou

DOI: 10.1504/IJMOR.2021.10044728

Article History:

Received:	25 June 2021
Accepted:	24 December 2021
Published online:	31 January 2023

Simulation of batch service lateness queues with multiple vacations and two Bernoulli catastrophes using the developed 'goLHS' generator

Anfal Rezgui and Hafida Saggou

Laboratory of RIIMA,
Faculty of Mathematics,
USTHB,
PB 32 El Alia 16111 Algiers, Algeria
Email: noor.anfal@gmail.com
Email: saggou-hafida@yahoo.fr

Megdouda Ourbih-Tari* and Meriem Boubalou

Institut des Sciences,
Centre Universitaire Morsli Abdellah de Tipaza,
Tipaza, 42020, Algeria
and
Laboratory of Applied Mathematics,
Faculty of Exact Sciences,
Bejaia University,
Bejaia, 06000, Algeria
Email: ourbihmeg@gmail.com
Email: boubaloumeriem@gmail.com
*Corresponding author

Abstract: This paper considers a new model of a single server queue subject to two different Bernoulli catastrophes with service lateness which represents the control duration and the service duration where both are in batches of fixed size K. We assume that catastrophes can occur when the server is in service or when it is on vacation. This proposed model is then solved using the appropriate generating functions, which gives steady state probabilities and some performance measures. Numerical results are sketched out to illustrate the effect of the system parameters on the main performance measures. To manage the proposed model with large batches, a Monte Carlo simulation is performed. For this purpose, a Latin hypercube sampling (LHS) generator called 'goLHS' is developed under MATLAB and it has been well tested. The 'rand' and 'goLHS' generators are used to compute the performance measures of the model when the batch size is large.

Keywords: batch service; Bernoulli catastrophe; multiple vacations; service lateness; Monte Carlo simulation; Latin hypercube sampling; LHS.

Reference to this paper should be made as follows: Rezgui, A., Saggou, H., Ourbih-Tari, M. and Boubalou, M. (2023) 'Simulation of batch service lateness queues with multiple vacations and two Bernoulli catastrophes using the developed 'goLHS' generator', *Int. J. Mathematics in Operational Research*, Vol. 24, No. 1, pp.128–153.

Biographical notes: Anfal Rezgui is a Doctoral student in Mathematics at the University Sciences and Technology Houari Boumediene (USTHB), Algiers, Algeria. She holds a license and Master's in Mathematics both from the Department of Probability and Statistics, USTHB. She is a member of research team at the RIIMA Laboratory in USTHB.

Hafida Saggou is currently a Full Professor at the University Sciences and Technology Houari Boumediene (USTHB), Algiers, Algeria. She is a member of a research team at the RIIMA Laboratory in USTHB. She received her DES in Mathematics at the USTHB. She received also her Doctorate in Mathematics at the USTHB. She is a member of PRFU Project. She is also numerous scientific journal papers and contributions. She is a supervisor for Doctoral studies at the USTHB.

Megdouda Ourbih-Tari is currently a Full Professor at the Tipaza University Center, Institute of Sciences. She is the Head of Research Team 'Statistics and Optimization by Simulation' at the Applied Mathematics Laboratory in Bejaia University and Research Projects. She was the Director of the Applied Mathematics Laboratory. She received her DES in Mathematics at the USTHB, Algiers, Post-Graduates Diploma and MPhil at the Lancaster University, UK both in Operational Research. She also received her Doctorate in Mathematics and Habilitation to Supervise Research (HDR) at the Bejaia University, Algeria. She is also numerous scientific journal papers, contributions. She is a supervisor for Doctoral studies.

Meriem Boubalou is currently a full teacher researcher at the Tipaza University Center in the Institute of Sciences. She is a member of the research team 'Statistics and Optimization by Simulation' at the Applied Mathematics Laboratory in Bejaia University. She holds a License of 'Statistics and Computer Processing of Data' and Master of Statistics and Decision Analysis at the Department of Mathematics in Bejaia University. She received also her Doctorate in Applied Mathematics at the Bejaia University, Algeri). She is a University Training Research Project (PRFU) member. She is also scientific journal paper.

1 Introduction

Catastrophes are natural or technological events which cause a serious breakdown in the functioning of a community or a society which has serious consequences for life in general and human beings in particular. In all catastrophe cases, it is necessary to seek to understand the processes which are at the origin of the causes and to put in place devices which avoid or minimise the consequences, then, if necessary, to manage them according to their seriousness.

The concept of catastrophes has various applications in a wide variety of areas of science and technology, particularly in computer-communications, biosciences, population studies and industries, etc. Queueing systems in which customers leave without getting service have received special attention in the literature. These types of queues are called queueing systems with catastrophes and are considered basic models of computers running in the presence of a virus (Chao, 1995).

The catastrophes are considered negative customers and can come from outside or inside the system, their characteristic is to remove from the system, some or all of the regular customers in the system. Such models were introduced by Gelenbe (1989) and Gelenbe et al. (1991). Chae et al. (2010) have studied a $GI/Geo/1$ queue with negative customers whose causes one positive customer to be removed if any is present. Zidani and Djellab (2018) have dealt with an $M/M/C/K$ queue with exponential abandonment at which the negative arrival has the effect of removing some number of customers from the service with a given probability. Sun and Wang (2019) have analysed an $M/M/1$ queue with negative customers, where the arrival of negative customer will break the server down and the positive customer being served (if any) is forced to leave the system.

Queueing systems with catastrophes, or so-called disasters, have been studied extensively, and this was discussed by several authors for instance: Kumar and Arivudainambi (2000) have studied an $M/M/1$ queue with possibility of catastrophes occurring at the service-facility as a Poisson process; whenever a catastrophe occurs, all customers in the system are destroyed immediately, then, the server is momentarily idle and is ready for service when a new arrival occurs; the asymptotic behaviour of the probability of the server being idle and the mean queue size were discussed. Di Crescenzo et al. (2003) has determined the heavy-traffic approximation of an $M/M/1$ queue with catastrophes where the transition probabilities, busy period and catastrophe waiting time densities were derived. Kumar and Madheswari (2002) have analysed an $M/M/2$ queueing system with the possibility of catastrophes; the service discipline is first come first served (FCFS), starting with an arbitrary number of customer, when a catastrophe occurs in the system, all customers are destroyed immediately, both servers get inactivated momentarily. A classical $M/M/1$ queue in the presence of disastrous breakdowns, system repair, and customer impatience was handled by Sudhesh (2010) where an explicit expression for the time-dependent queue size system distributions was derived and numerical illustrations were given. Priya and Sudhesh (2018) have studied a discrete-time infinite server queue subject to system disaster, the exact time-dependent probabilities of the number of customers present in the system are obtained. While Vijayashree and Anjuka (2019) dealt with the stationary analysis of an $M/M/1$ queue subject to a disaster and a subsequent repair. In addition, there have been many studies on queues with catastrophes, which have also been referred to as a ‘mass exodus’ for instance in Chen and Renshaw (1997) and Zhang and Li (2015). It can also be considered as a type of ‘stochastic clearing’ which removes all the workload in the system whenever it occurs (Stidham, 1974; Yang et al., 2002).

Many researchers have been studying queueing systems with vacations, such as Doshi (1986) where the author investigates two queueing models with starting vacations when the server becomes idle; in the first one, the server keeps on taking vacations until when returning from a vacation it will find at least one customer in the queue; in the second, the server takes exactly one vacation at the end of each busy period. Lucantoni et al. (1990) have dealt with a single-server queueing system with multiple vacations; in such a queue, a server vacation begins whenever the system becomes empty. Tian and Zhang (2002) investigated a discrete-time $GI/Geo/1$ queue with server vacations where the authors considered an exhaustive service and a multiple vacation policy that requires the server to take vacations when the system has no customers waiting at a service completion time or vacation completion time. Yue et al. (2014) considered an $M/M/c$ queueing system with a synchronous vacation, where whenever the system becomes empty, all servers take a vacation; if the system is still empty when the vacation ends, all servers take another vacation; otherwise, they return to the service. Laxmi and Kassahun (2020) studied an infinite capacity multi-server Markovian queue with synchronous multiple working vacations, where the c servers go for a working vacation when the system is empty and they provide service according to exponential distributions with parameters μ_b and μ_v during regular and working vacation periods, respectively, such that $\mu_v < \mu_b$.

The study of queueing systems with batch-service dates back to Bailey (1954), who studied a single queue where customers arrive at random and are served in batches, with each batch size having a maximum fixed value; the author investigated the equilibrium distribution of queue length by the imbedded Markov chain method. Wu et al. (2010) have proposed an improved model for parallel batch systems which gives a better approximation by reducing the systematic error existing in the previous models which ignored the dependence between the queueing time and the wait-to-batch time.

On the other hand, we can observe the existence of the service lateness in several queueing models, in which the service can start late due to the verification or classification of the customers before being served, especially, if the service is performed in batches, so customers should undergo a control before service. Such a system often arises in many real life problems, for example, in the study of a population, health sectors, production lines, etc. Armony et al. (2009) have studied the performance impact of the delay before service, in customer contact centres, which is represented by making announcements on arrivals of customers who must wait before starting service. Saggou et al. (2019) have also discussed this type of queues, in which they expressed the lateness in service repair, where its repair is not supported immediately, there is a delay time verification before starting the repair.

So, this work will propose a new model of batch service queueing system with service lateness and two different Bernoulli catastrophes (during service and during vacation) under a multiple vacation policy. The catastrophe in service can be considered as a natural disaster or a virus, etc. The vacation can model the repair of the server or the server is called upon to perform another service task. The vacation catastrophes can be interpreted as a very lengthy repair resulting in the loss of customers (catastrophe). We present numerical applications to study the effect of various parameters on performance measures when the batch size K is small.

The simulation is one of the most used method for modelling and analysing various complex systems due to its contribution to improving performance in terms of productivity. In recent years, the simulation has been widely used to analyse the behaviour of queueing systems that arise from stochastic systems (Shen et al., 2018; Manurung, 2019) by several different methods and across many programs (Blanchet et al., 2018; Ourbih-Tari et al., 2019; Mishra et al., 2021).

Knowing that the Monte Carlo (MC) method is inaccurate compared to other sampling methods, given that Latin hypercube sampling (LHS) is an attractive alternative to the MC method in computer experiments (Iman and Conover, 1980;

McKay et al., 1979) and the latter is an efficient method for selecting the values of input variables by simultaneously stratifying them on all input dimensions (Stein, 1987; Helton and Davis, 2003), LHS was then selected to generate inputs for the simulation of our proposed model. Recently, LHS has been introduced in various software under Windows and Linux environments as the default method (Bell, 2019), for instance the MATLAB language, but it cannot be used in the simulation of queues because the sample size is not known at the beginning of the simulation. Boubalou et al. (2019) have proposed a getLHS modified generator in C language to generate input random variables in the $M/G/1$ queues, so, we will follow the method of LHS modified by Boubalou et al. (2019) to generate input samples in the simulation of queues. By the way, we have proposed a new LHS number generator in the MATLAB language, called 'goLHS' which is highly tested before used. Then, a software is developed in MATLAB for the simulation of our model to study the performance measures for any batch size K, especially when K is large since the theoretical computations are too complicated and even impossible. Finally, a simulation for a given time period is performed using the proposed goLHS generator and rand generator of MATLAB for the comparison purpose.

The main objective of this paper is to discuss the stability of the queue model exposed to two types of catastrophes, and to clarify the extent of the impact of the latter on the operation of the system under the batch service discipline. The service is implemented in two phases, starting with the control and followed by the service, and both are performed in batches. Thus, the main motivation of this research is to analyse and study the stability of batch service queue subject to catastrophes; including the study of the impact of each of the following three tasks, catastrophe, batch size and control on the average number of customers present in the system and also on the average rate of loss customers in the system. Additionally, in this paper, the simulation technique is used to model our system which is considered as a means of analysing the operation of the system by evaluating the impact of various changes that occur on global system performances in a practically complex case which is represented in our model by large batch sizes.

The paper is organised as follows. In Section 2, the model is described according to the given assumptions. Section 3 is devoted to the detailed analysis where we derive the system steady state equations and the probability generating functions. Moreover, the steady state probabilities for the number of customers in the queue are shown for completeness. The most important system performance measures are obtained in Section 4. Section 5 illustrates some numerical examples showing the impact of different parameters to performance measures of the system. A MC simulation of the proposed model is provided in Section 6 using either the MC method or the LHS method where a description of the proposed LHS number generator is provided together with the performed statistical tests. Section 7 presents the comparison of both simulation results and finally the last section concerns conclusion.

2 The model description

We consider a fixed-batch lateness service queueing system with the possibility of catastrophes under multiple vacations, at which customers arrive at a single-server facility, following a stationary Poisson process with rate λ.

The service times of customers are independently and identically distributed (iid) random variables following the same exponential distribution with parameter μ, the discipline is FCFS and there is an infinite number of places for waiting customers. The control times are iid exponentially distributed with parameter β.

Catastrophes occur in the system, when the server is in service with a probability p (does not occur with probability $1-p$) according to a Poisson process with a parameter ν, and when the server is on vacation with a probability q (does not occur with probability $1-q$) according to a Poisson process with a parameter ν'. While the length of a vacation time is exponentially distributed at rate α.

Without losing generality, we assume that initially at time $t=0$ there are K $(K \geq 1)$ customers in the system, so, the control starts immediately with a batch size K. After completion of the control, the service of such a batch starts immediately. Once the service is completed without catastrophes, if there are K or more customers in the queue, then the first K customers will be selected and their control will start, else, if there are less than K customers, the server goes for multiple vacations until at its return, it finds at least K customers waiting in the system to start their control.

Whenever a catastrophe occurs in the system when the server is in service or on vacation, all customers present in the system are lost immediately and the server goes for multiple vacations (the vacation may be considered as a repair or may be modelled as another service task performed outside the system).

We assume that the stochastic processes involved in our system are all independent of each other.

Let $Y(t)$ be the number of customers in the queue at time t. To define a Markov process, let us introduce the random variable $C(t)$ which represents the server states at time t, where

$$C(t) = \begin{cases} 1 & when\ the\ server\ is\ in\ control\ at\ time\ t. \\ 2 & when\ the\ server\ is\ busy\ (in\ service)\ at\ time\ t. \\ 3 & when\ the\ server\ is\ on\ vacation\ at\ time\ t. \end{cases}$$

then $\{Y(t), C(t), t \geq 0\}$ is a Markov process.

Since the model described above has a rather complex structure including batch control, batch service and Bernoulli catastrophes, a special attention is provided to its study due to its difficulty of analysis, in addition to its scientific usefulness, as it can be used to analyse and improve the system performances in various areas of real life, for instance, the recording of programs in computer science or the process of COVID vaccine in the medical field.

3 Analysis of the steady-state probabilities

In the case where the steady-state distribution exists, the limiting probabilities will be defined for $n \geq 0$ as follows

- $P_{n,1} = \lim_{t \longrightarrow \infty} P[Y(t) = n, C(t) = 1]$
- $P_{n,2} = \lim_{t \longrightarrow \infty} P[Y(t) = n, C(t) = 2]$
- $P_{n,3} = \lim_{t \longrightarrow \infty} P[Y(t) = n, C(t) = 3]$.

Then, the steady state equations of Chapmann-Kolmogorov of our model are given by

$$(\lambda+\beta)P_{0,1}=\mu P_{K,2}+\alpha P_{K,3}, \tag{1}$$

$$(\lambda+\beta)P_{n,1}=\lambda P_{n-1,1}+\mu P_{n+K,2}+\alpha P_{n+K,3}, n\geq 1 \tag{2}$$

$$(\lambda+\mu+\nu p)P_{0,2}=\beta P_{0,1}, \tag{3}$$

$$(\lambda+\mu+\nu p)P_{n,2}=\lambda P_{n-1,2}+\beta P_{n,1}, n\geq 1 \tag{4}$$

$$(\lambda+\nu' q)P_{0,3}=\mu P_{0,2}+\nu p\sum_{n\geq 0}P_{n,2}+\nu' q\sum_{n\geq 0}P_{n,3}, \tag{5}$$

$$(\lambda+\nu' q)P_{n,3}=\lambda P_{n-1,3}+\mu P_{n,2}, 1\leq n\leq K-1 \tag{6}$$

$$(\lambda+\alpha+\nu' q)P_{n,3}=\lambda P_{n-1,3}, n\geq K, \tag{7}$$

where the normalisation condition is

$$\sum_{n\geq 0}P_{n,1}+\sum_{n\geq 0}P_{n,2}+\sum_{n\geq 0}P_{n,3}=1. \tag{8}$$

Theorem 1: Under the stability condition, the steady state probabilities for number of customers in the queue when the server is in control, busy or on vacation are given respectively by

$$P_{n,1}=P_{0,1}s^n\sum_{i=0}^{n}\left(\frac{r}{s}\right)^i, n\geq 0 \tag{9}$$

where $r=\frac{1}{\tilde{z}}$ and $s=\frac{\lambda}{\alpha+\nu' q+\lambda}$.

$$P_{n,2}=\frac{\beta P_{0,1}}{\mu+\nu p+\lambda}\sum_{l=0}^{n}s^l\left(\sum_{i=0}^{l}\left(\frac{r}{s}\right)^i\right)\left(\frac{\lambda}{\mu+\nu p+\lambda}\right)^{n-l}, n\geq 0 \tag{10}$$

$$P_{n,3}=\begin{cases}\frac{P_{0,1}}{\lambda+\nu' q}\left\{\frac{\mu\beta}{\mu+\nu p+\lambda}\left[\sum_{m=0}^{n}\left(\sum_{l=0}^{m}s^l\left(\sum_{i=0}^{l}\left(\frac{r}{s}\right)^i\right)\right.\right.\right. \\ \left.\left.*\left(\frac{\lambda}{\mu+\nu p+\lambda}\right)^{m-l}\right)*\left(\frac{\lambda}{\lambda+\nu' q}\right)^{n-m}\right] \\ +\left(\frac{\lambda}{\lambda+\nu' q}\right)^n*\left(\frac{\beta}{\mu+\nu p}\frac{\nu p-\nu' q}{(1-r)(1-s)}\right. \\ \left.\left.-\frac{\nu' q}{(1-r)(1-s)}\right)\right\}+\left(\frac{\lambda}{\lambda+\nu' q}\right)^n\frac{\nu' q}{\lambda+\nu' q}, & \text{for } 0\leq n<K \\ \left(\frac{\lambda}{\alpha+\nu' q+\lambda}\right)^{n-K+1}P_{K-1,3}, & \text{for } n\geq K\end{cases} \tag{11}$$

where

$$P_{0,1} = \frac{Nr}{Dr}, \tag{12}$$

such as

$$Nr = 1 - \frac{\nu' q}{\lambda + \nu' q} \sum_{n=0}^{K-1} \left(\frac{\lambda}{\lambda + \nu' q} \right)^n - \frac{\nu' q}{\alpha + \nu' q} \left(\frac{\lambda}{\lambda + \nu' q} \right)^K,$$

and

$$\begin{aligned} Dr =& \frac{\beta + \mu + \nu p}{(\mu + \nu p)(1-r)(1-s)} \\ &+ \left(\frac{1}{\lambda + \nu' q} \right) \left\{ \frac{\mu \beta}{\mu + \nu p + \lambda} \sum_{n=0}^{K-1} \left[\sum_{m=0}^{n} \left(\sum_{l=0}^{m} s^l \left(\sum_{i=0}^{l} \left(\frac{r}{s} \right)^i \right) \right. \right. \right. \\ &* \left. \left. \left(\frac{\lambda}{\mu + \nu p + \lambda} \right)^{m-l} \right] * \left(\frac{\lambda}{\nu' q + \lambda} \right)^{n-m} \right] + \left[\frac{\beta}{\mu + \nu p} \frac{\nu p - \nu' q}{(1-r)(1-s)} \right. \\ &- \left. - \frac{\nu' q}{(1-r)(1-s)} \right] * \sum_{n=0}^{K-1} \left(\frac{\lambda}{\lambda + \nu' q} \right)^n \Bigg\} + \left(\frac{1}{\lambda + \nu' q} \right) \left\{ \frac{\mu \beta}{\mu + \nu p + \lambda} \right. \\ &* \left[\sum_{m=0}^{K-1} \left(\sum_{l=0}^{m} s^l \left(\sum_{i=0}^{l} \left(\frac{r}{s} \right)^i \right) * \left(\frac{\lambda}{\mu + \nu p + \lambda} \right)^{m-l} \right) * \left(\frac{\lambda}{\nu' q + \lambda} \right)^{K-1-m} \right] \\ &+ \left(\frac{\lambda}{\lambda + \nu' q} \right)^{K-1} * \left[\frac{\beta}{\mu + \nu p} \frac{\nu p - \nu' q}{(1-r)(1-s)} \right. \\ &- \left. \left. \frac{\nu' q}{(1-r)(1-s)} \right] \right\} \left(\frac{\lambda}{\alpha + \nu' q} \right). \end{aligned}$$

Proof: To solve the system equations (1)–(7), we use the method of generating functions. Let us define for $|z| \leq 1$, the following probabilities generating functions (PGF's) for the number of customers in the queue

- when the server is in control

  $$B(z) = \sum_{n \geq 0} z^n P_{n,1},$$

- when the server is busy

  $$G(z) = \sum_{n \geq 0} z^n P_{n,2},$$

- when the server is on vacation

  $$H(z) = \sum_{n \geq 0} z^n P_{n,3}.$$

a Multiplying both sides of equation (2) by z^n, then taking summation over all possible values of n and adding the obtained equation to equation (1), we get

$$\left[\lambda z^{K+1}-(\lambda+\beta)z^{K}\right]B(z)+\mu G(z)+\alpha H(z)$$
$$=\mu\sum_{n=0}^{K-1}z^{n}P_{n,2}+\alpha\sum_{n=0}^{K-1}z^{n}P_{n,3}. \quad (13)$$

b In the same way as a, but for equations (4) and (3) instead of equations (2) and (1) respectively, we have

$$[\mu+\nu p+\lambda(1-z)]G(z)=\beta B(z). \quad (14)$$

c Using also the same principe as in a for equations (6) and (7) instead of equation (2) and (5) instead of equation (1), we obtain

$$[\alpha+\nu' q+\lambda(1-z)]H(z)=\mu\sum_{n=0}^{K-1}z^{n}P_{n,2}+\alpha\sum_{n=0}^{K-1}z^{n}P_{n,3}$$
$$+\nu p\sum_{n\geq 0}P_{n,2}+\nu' q\sum_{n\geq 0}P_{n,3}. \quad (15)$$

Now from equations (13), (14) and (15), we get

$$B(z)=\frac{A-B}{L}, \quad (16)$$

where

- $A=(\nu' q+\lambda(1-z))\left(\mu\sum_{n=0}^{K-1}z^{n}P_{n,2}+\alpha\sum_{n=0}^{K-1}z^{n}P_{n,3}\right)(\mu+\nu p+\lambda(1-z))$.
- $B=\alpha\left[\nu p\sum_{n\geq 0}P_{n,2}+\nu' q\sum_{n\geq 0}P_{n,3}\right](\mu+\nu p+\lambda(1-z))$.
- $L=(\alpha+\nu' q+\lambda(1-z))\left[\left(\lambda z^{K+1}-(\lambda+\beta)z^{K}\right)(\mu+\nu p+\lambda(1-z))+\mu\beta\right]$.

The generating function $B(z)$ has the property that it must converge inside the unit circle $|z|\leq 1$ and noticing that, the expression in the $B(z)'$ denominator has $K+1$ zeros, we can use the *Rouche's theorem* to say that K zeros of this expression lie inside the circle $|z|\leq 1$ and one zero lies outside the circle $|z|\leq 1$. According to the holomorphy of the generating function, the K zeros of the $B(z)'$ denominator must coincide with K zeros of $B(z)'$ numerator.

Let $\tilde{z}$ be a zero which lies outside the circle $|z|\leq 1$ and as $B(z)$ is convergent, the K zeros will be then simplified to allow us to write

$$B(z)=\frac{C}{(\alpha+\nu' q+\lambda(1-z))\lambda(z-\tilde{z})}, \quad (17)$$

by replacing $z=0$ in equation (17), we obtain $C=\lambda(\alpha+\nu' q+\lambda)\tilde{z}P_{0,1}$ then $B(z)$ becomes

$$B(z)=\frac{(\alpha+\nu' q+\lambda)\tilde{z}P_{0,1}}{(\alpha+\nu' q+\lambda(1-z))(\tilde{z}-z)}, \quad (18)$$

after some simplifications, equation (18) is as follows

$$B(z) = P_{0,1} \sum_{n\geq 0} \left(\frac{\lambda z}{\alpha + \nu' q + \lambda} \right)^n \sum_{n\geq 0} \left(\frac{z}{\tilde{z}} \right)^n . \tag{19}$$

From the formula (19), we deduce equation (9) and replacing equation (9) in equation (4) recursively for $n = 0, 2, 3, ...$ we get equation (10), then, substituting equation (10) in equation (6) recursively for $n = 0, 2, 3, ...$ we have equation (11) and this ends the proof of the theorem.

Remark 1: For the system to be stable, it is necessary and sufficient that $\frac{\lambda}{\mu + \beta} < K$.

4 The performance measures of the system

The performance measures of the system are given in the following corollaries.

Corollary 1:

1 The probability that the server is in control is given by

$$P_C = \sum_{n\geq 0} P_{n,1} = \frac{P_{0,1}}{(1-r)(1-s)}.$$

2 The probability that the server is in service (busy) is given by

$$P_S = \sum_{n\geq 0} P_{n,2} = \frac{\beta P_{0,1}}{(1-r)(1-s)(\mu + \nu p)}.$$

3 The probability that the server is waiting for service is given by

$$P_W = P_C = \frac{P_{0,1}}{(1-r)(1-s)}.$$

4 The probability that the server is on vacation is given by

$$P_V = \sum_{n\geq 0}^{K-1} P_{n,3} + (\frac{\lambda}{\alpha + \nu' q}) P_{K-1,3}.$$

Corollary 2: Let N and Y be respectively the number of customers in the system and the number of customers in the queue; their expectations under the steady state conditions are given respectively by

$$E(N) = \sum_{n\geq 0} \left[(n+K) \left(P_{n,1} + P_{n,2} \right) + n P_{n,3} \right],$$

$$E(Y) = \sum_{n\geq 0} n \ \left[P_{n,1} + P_{n,2} + P_{n,3} \right].$$

Corollary 3:

1 The average rate of loss customers during the service period is

$$\xi_s = \nu p \sum_{n \geq 0} (n + K) P_{n,2}.$$

2 The average rate of loss customers during the vacation period is

$$\xi_v = \nu' q \sum_{n \geq 0} n P_{n,3}.$$

3 The average rate of loss customers in the system is

$$\xi = \xi_s + \xi_v.$$

5 Numerical study

In this section, we present some numerical examples to study the effect of various parameters such as the batch size, the catastrophe rate during service, the catastrophe rate during vacation and the control rate on the main performance measures for small batch size K = 1, 2, 3, 4.

5.1 Effect of K on $E(N)$ and $E(Y)$

Take λ = 3, μ = 9, β = 12, α = 6, ν = 5, ν' = 4, p = 0.5 and q = 0.3.

From Table 1, we deduce that the number of customers in the system increases with the increasing values of K. Indeed, when the value of K increases, the number of customers present in the service and in the control increases, which results in the increasing of the number of customers in the system.

Table 1 The effect of K on $E(N)$ and $E(Y)$

	K = 1	*K = 2*	*K = 3*	*K = 4*
$E(N)$	0.8953	1.1243	1.3557	1.5620
$E(Y)$	0.5493	0.7854	1.0674	1.3241

The same pattern is observed for the number of customers in the queue. Indeed, as the K value increases, the number of customers waiting to enter the control increases, which leads to the increase of the number of customers in the queue.

5.2 Effect of K on P_S, P_V, P_C, ξ_s, ξ_v and ξ

Now we take λ = 8, μ = 5, α = 6, β = 9, ν = 2, ν' = 3, p = 0.5 and q = 0.4 (P_C is calculated to verify the equality $P_C + P_S + P_V = 1$).

It is clearly observed from Table 2 that when the batch K increases, the probability of the busy period P_S decreases, which leads to a decrease of the average rate ξ_s of customers loss during the service. On the other hand, the probability P_V of the vacation period and the average rate ξ_v of customers loss during the vacation increase significantly as the batch size K increases, which is in line with the expectation, indeed, the higher the vacation period, the more the server becomes prone to catastrophes during this period, and consequently more customers are lost.

Table 2 The effect of K on P_S, P_V, P_C, ξ_s, ξ_v and ξ

	$K = 1$	$K = 2$	$K = 3$	$K = 4$
P_S	0.4644	0.3669	0.2803	0.2135
P_V	0.2261	0.3885	0.5328	0.6441
P_C	0.3096	0.2446	0.1869	0.1424
ξ_s	2.0122	1.9567	1.7753	1.5659
ξ_v	0.3014	0.7005	1.2335	1.8013
ξ	2.3136	2.6573	3.0088	3.3671

5.3 Effect of ν and ν' on $E(N)$ varying p and q

We show the effect of the catastrophe parameters ν (in service) and ν' (on vacation) on $E(N)$ for different values of p when $q = 0.5$ and for different values of q when $p = 0.5$ which can be seen respectively in Figures 1 and 2. The other parameters are chosen arbitrary such as $K = 1$, $\lambda = 3$, $\mu = 5$, $\alpha = 3$, $\nu = 2$ and $\nu' = 3$.

Figure 2 indicates that $E(N)$ decreases rapidly as ν' increases with increasing values of q, while Figure 1 indicates that $E(N)$ decreases slowly as ν increases with increasing values of p. We can deduce from both figures that $E(N)$ changes outstandingly when changing the q and ν' values compared to the changing of p and ν values, which indicates that the occurrence of catastrophes on vacation has a high influence on the system than the occurrence of catastrophe in service. Thus, the occurrence of catastrophes during the vacation period leads to the loss of a very large number of customers in the system, which in turn minimises the operation of the system and significantly reduces its productivity, compared to its occurrence during the service period.

5.4 Effect of β on $E(N)$

Now, we illustrate the effect of the control parameter β on $E(N)$ for different values of p when $q = 0.3$ and for different values of q when $p = 0.3$. The other parameters are chosen arbitrary such as $K = 3$, $\lambda = 5$, $\mu = 6$ and $\alpha = 2$.

5.4.1 Varying p

- Figure 3: For each value of p, we study the influence of the control rate β on $E(N)$. Here, ν and ν' are taken to be equal to 3.
- Figure 4: For each value of p, we study the influence of the control rate β on $E(N)$, but here, ν and ν' are taken to be $\nu = 15$ and $\nu' = 3$ ($\nu > \nu'$).
- Figure 5: For each value of p, we study the influence of the control rate β on $E(N)$, but here, ν and ν' are taken to be $\nu = 3$ and $\nu' = 15$ ($\nu < \nu'$).

Almost the same pattern is seen in Figures 3, 4 and 5 where $E(N)$ decreases with increasing values of β, respectively p. This means that the higher the control rate, the lower the average number of customers in the system, resulting in faster system performance and more customers served.

Figure 1 The effect of ν on $E(N)$ for different values of p (see online version for colours)

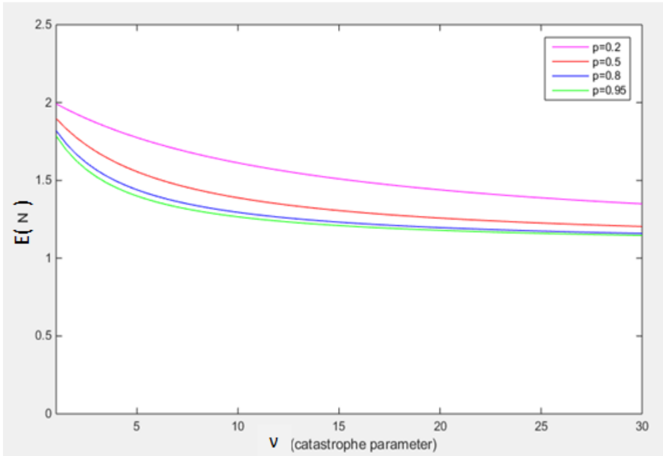
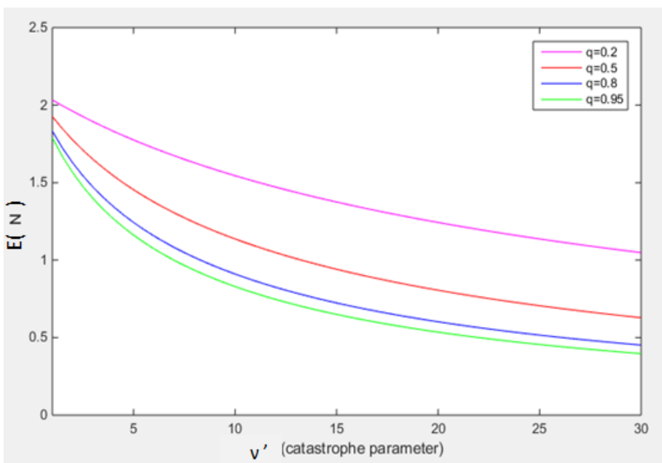


Figure 2 The effect of ν' on $E(N)$ for different values of q (see online version for colours)



We also notice that the decrease of $E(N)$ is significant when the value of ν' is greater than ν (Figure 5) compared when both parameters are equal (Figure 3), but it is not, when the value of ν' is less than ν (Figure 4). This is due to the large loss of customers during the vacation period resulting from the high rate of catastrophes in this period.

5.4.2 Varying q

- Figure 6: For each value of q, we study the influence of the control rate β on $E(N)$. Here, ν and ν' are taken to be equal to 3.
- Figure 7: For each value of q, we study the influence of the control rate β on $E(N)$, but here, ν and ν' are taken to be ν = 15 and ν' = 3 ($\nu > \nu'$).
- Figure 8: For each value of q, we study the influence of the control rate β on $E(N)$, but here, ν and ν' are taken to be ν = 3 and ν' = 15 ($\nu < \nu'$).

Figure 3 The effect of β on $E(N)$ for $\nu = \nu'$ (see online version for colours)

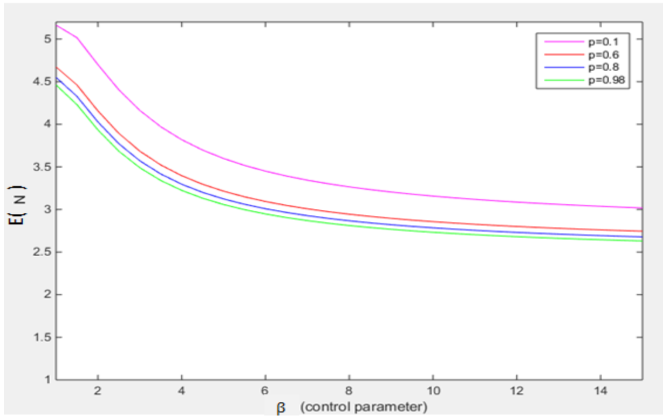


Figure 4 The effect of β on $E(N)$ for $\nu > \nu'$ (see online version for colours)

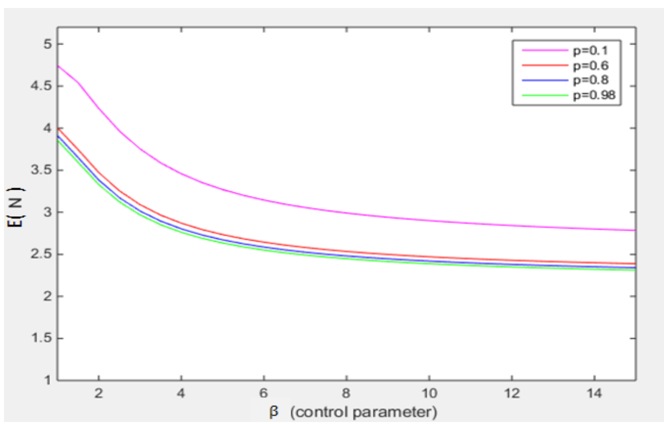


Figure 5 The effect of β on $E(N)$ for $\nu < \nu'$ (see online version for colours)

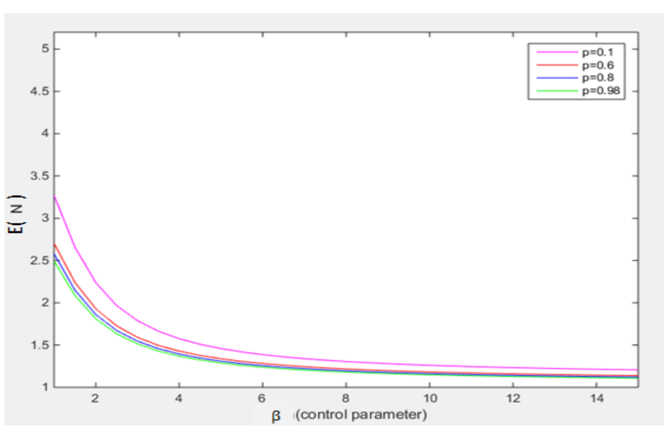


Figure 6 The effect of β on $E(N)$ for $\nu = \nu'$ (see online version for colours)

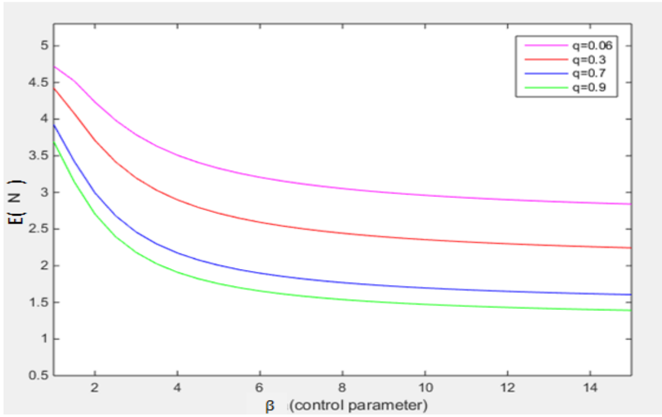
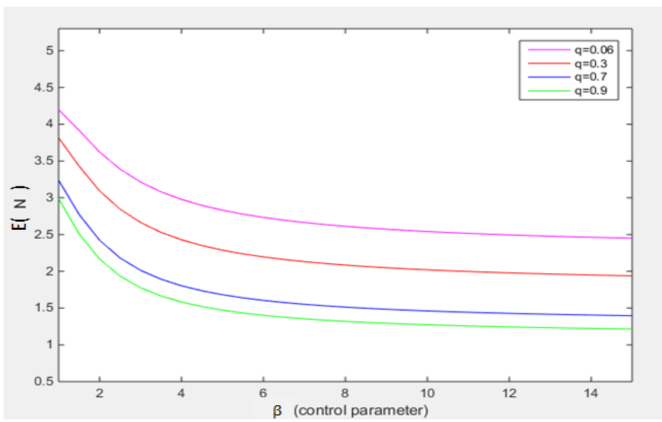


Figure 7 The effect of β on $E(N)$ for $\nu > \nu'$ (see online version for colours)



We can see from Figures 6 to 8 that $E(N)$ decreases with increasing values of β, respectively q. By the way, the decrease of $E(N)$ is significant, not only when the value of ν' is greater than ν (Figure 8), but also with the varying of q values, this is due to the high probability of catastrophes occurring during the vacation period in all three cases, which also results in a greater possibility of losing customers and therefore a considerable decrease in the average number of customers in the system.

5.5 Effect of pairs (ν, ν'), (ν, β) and (ν', β) on the mean number of customers in the system

In this subsection, we illustrate respectively in Figures 9 to 11 the effect of catastrophe parameters in service ν, and on vacation ν', together with the control rate parameter β on $E(N)$.

Figure 8 The effect of β on $E(N)$ for $\nu < \nu'$ (see online version for colours)

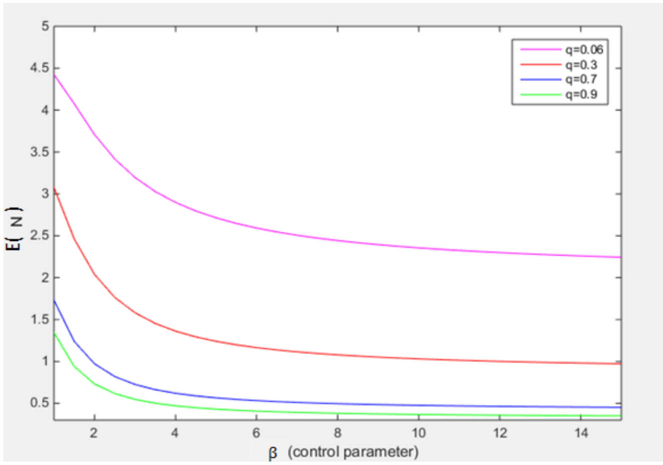
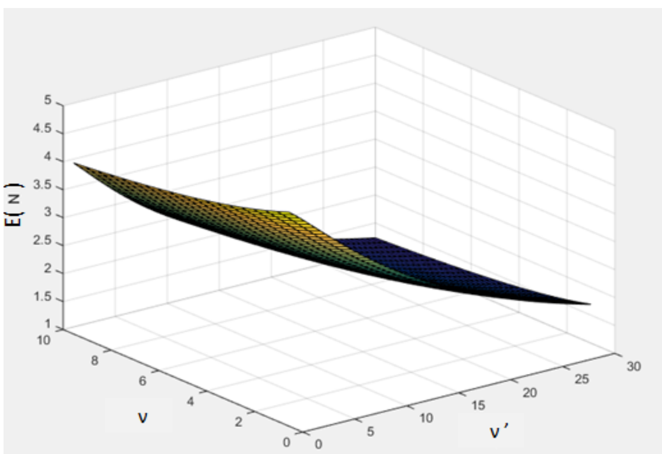


Figure 9 The effect of (ν, ν') on $E(N)$ (see online version for colours)



From Figures 9 to 11, we can say that the mean number of customers in the system behaves well with the chosen parameters and all plotted surfaces are smooth.

6 MC simulation

This section is concerned by the MC simulation of the proposed model of queueing system by generating random samples of different input variables using two different sampling methods in order to compute and analyse performance measures for large batch sizes. The considered output variables of this queue are the number of customers in the system N and the number of customers in the queue Y. Both response variables have the mean parameter to be estimated by $\bar{N}$ and $\bar{Y}$. For this purpose, first, a software is developed in MATLAB for the simulation of our model using the 'rand' generator incorporated in MATLAB language to generate inputs. Since the LHS method has important advantages over the MC method for some common purposes and knowing that the LHS software exists in MATLAB, but in simulation queues, the sample size is unknown in advance, so, we will follow the method of LHS modified by Boubalou et al. (2019) to generate input samples to be used in the simulation of our model. To this end, an LHS generator is developed in MATLAB that we have called 'goLHS' and it was well tested to be used safely. The simulation results are given in Section 7 for the comparison.

Figure 10 The effect of (β, ν) on $E(N)$ (see online version for colours)

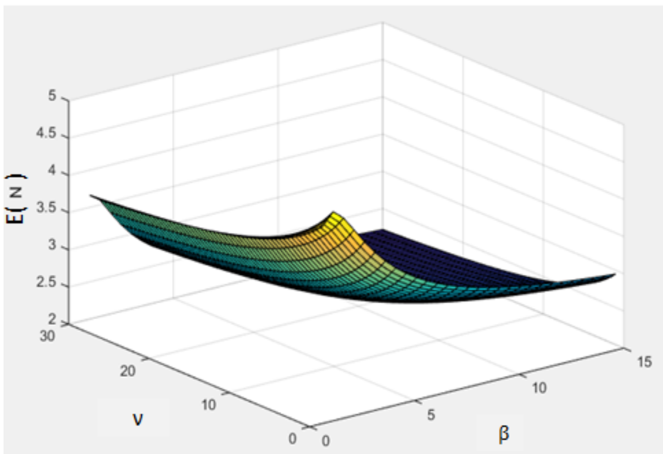
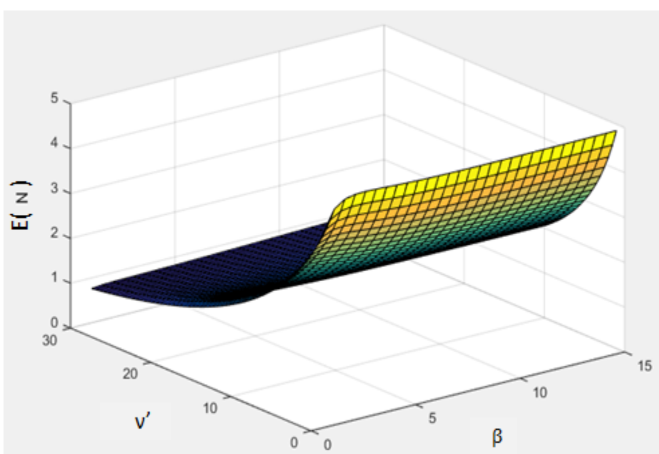


Figure 11 The effect of (β, ν') on $E(N)$ (see online version for colours)



6.1 *Standard LHS*

Let us denote the cumulative distribution function (*cdf*) of the vector of random variables $X = (X_1, ..., X_K)$ by $F(x)$, and the required number of realisations by n. In contrast to the MC method, LHS operates by dividing the range of each $X_k, k = 1, ..., K$

into n strata of an equal marginal probability $1/n$, and sampling once from each stratum. The samples of $X_{ik}, i = 1, ..., n, k = 1, ..., K$ of each component are drawn from the respective strata according to

$$x_{ik} = F_k^{-1}\left(\frac{p_{ik} - r_{ik}}{n}\right),$$

where F_k^{-1} represents the inverse of the target *cdf* for the variable X_k, the p_{ik} are independent uniform random permutations of the integers $1, 2, ..., n$, and the r_{ik} are independent $U_{[0,1]}$ random numbers and independent of the p_{ik}.

6.2 The use of LHS in queueing systems

Without losing generality, we consider a random variable X_k that we denote by X and therefore a *cdf* F_k that we denote by F in a given run generated by m sub-runs. So, in the modified LHS run (Boubalou et al., 2019), we generate input samples by distributing a block of m sub-samples of size, an integer n_j, randomly generated as required by the simulation for $j = 1, ..., m$. We stop the process of the generation when the simulation run terminates. Then, the sub-samples values of X are obtained by the following formulas:

$$x_{ij} = F^{-1}(S_{ij}), \qquad \text{for } i = 1, ..., n_j \text{ and } j = 1, ..., m$$

where S_{ij} is randomly selected from the following LHS numbers:

$$S_{ij} = \frac{i_j - r_{ij}}{n_j}, \qquad \text{for } i = 1, ..., n_j \text{ and } j = 1, ..., m \tag{20}$$

6.3 The proposed algorithm of the LHS modified to be used in MATLAB

At the beginning of the first run, let $j = 0$.

a
- a_1 Generate an integer number 'n' from the interval [min, max].
- a_2 Generate the LHS modified numbers S_i, $i = 1, 2, ..., n$ and store them in the vector *Sdata*.
- a_3 Permute integers from 1 to n randomly and store them in the vector 'index' such as *index = randperm(n)*.

b Generate the n observations of LHS $x_{j+1}, ..., x_{j+n}$, and store them in the vector 'x' such as *x = [Sdata(index)]*.

c After each sub-run, check that:

- c_1 If the sample of Latin hypercube values is not enough and more values are needed, let $j = j + n$:
  - Do A.
  - Generate the n observations of LHS $x_{j+1}, ..., x_{j+n}$, and store them in a vector *x = [x, Sdata(index)]*.

- Go to c.

c_2 Otherwise, collect the results after each sub-run and go to d.

d Collect the results after each run.

6.4 Description of the proposed LHS generator

The 'goLHS' number generator is designed in MATLAB language and it ensures pseudo-random numbers as required by the simulation according to the proposed LHS algorithm given in Subsection 6.3. The built-in LHS library consists on three files permutation MATLAB, goLHS1 and goLHS2.

The first file of the developed library contains the following available functions in MATLAB:

- randi([min, max]): Generate a random integer number uniformly distributed from the interval [min, max].
- randperm(): Gives a row vector containing a random permutation of the integers from 1 to the inclusive given value.

The second file 'goLHS1' is the function that generates the starter bloc of uniform numbers between 0 and 1 using the proposed LHS modified algorithm.

The main file 'goLHS2' generates the desired LHS numbers according to the proposed LHS modified algorithm.

To generate a sequence of pseudo-random numbers, we firstly call the goLHS1 function for generating the starter block, this function will in its turn call the permutation MATLAB function which asks the user to enter the 'min' and 'max' values. Then, we call the goLHS2(n) function to compute the 'n' desired LHS numbers $S_i, i = 1, ..., n$ using formula (20) in Subsection 6.2 for a given j.

6.5 Characteristics of the uniform [0, 1[distribution

We give in the following table the characteristics of the theoretical Uniform distribution [0, 1[and the empirical uniform distribution given by the sample of 5,000 numbers generated by goLHS.

Table 3 Characteristics of $U_{(0,1)}$ distribution

—	*Theoretical*	*Empirical 'LHS'*
Mean value	0.5	0.4999765
Variance value	0.0833333	0.08340814

Table 3 shows that the characteristics of the generator goLHS is closer to the theoretical parameters of U[0, 1[. The estimates of the empirical uniform distribution are simple, but since the used sample size is quite large, so no variability of the estimates is left as a guess for the true parameters.

6.6 Testing the uniformity

Let us test the uniformity of our LHS library, by:

- The Kolmogorov-Smirnov (KS) test.
- The representation on R^2.
- The representation of the empirical *cdf* (*ecdf*) and the uniform *cdf*.

For all performed tests, we use a significant level of $\alpha = 0.05$ and all obtained results show that the LHS library is a good generator.

Figure 12 Scatterplots of 50 pairs of goLHS numbers over $[0, 1[^2$ (see online version for colours)

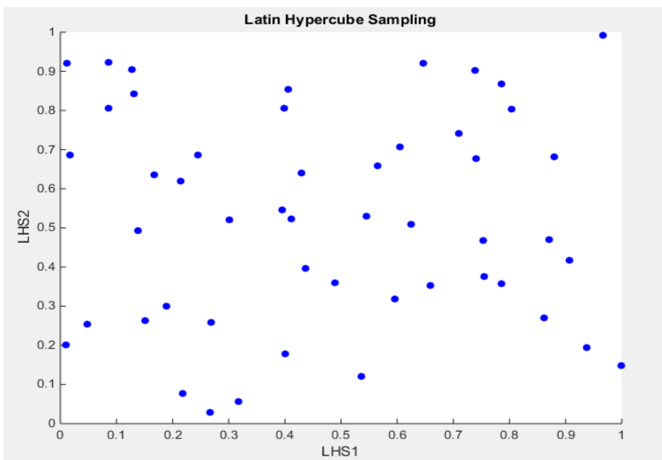
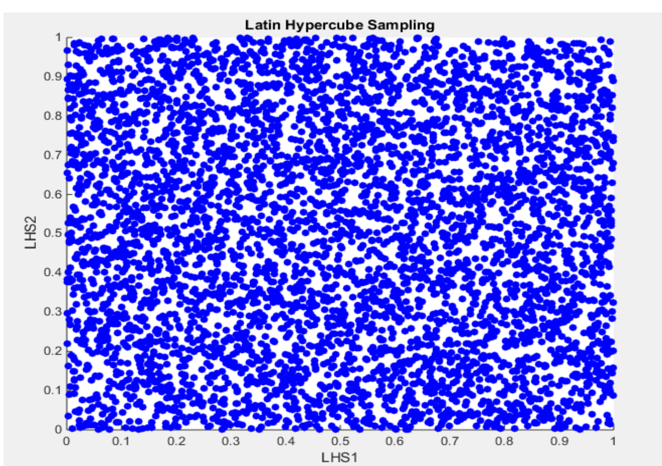


Figure 13 Scatterplots of 5,000 pairs of goLHS numbers over $[0, 1[^2$ (see online version for colours)



6.6.1 KS test

The nonparametric KS test is used to check whether a sample from our LHS library can actually be fitted to the uniform distribution. Let us define the null hypothesis H_0 by the stream of goLHS following a uniform distribution. For this test carried out using the statistical package 'R', a stream of n = 5,000 numbers is generated by the LHS library and the results are given in Table 4.

Table 4 The results of KS test

—	*Statistic*	*p-value*
LHS	0.0023	1

It can be seen clearly from Table 4 that the p-value obtained is greater than 0.05, so there is certainly no presumption against the null hypothesis.

6.6.2 Graphical representation of the double series on R^2

Both small and large samples are generated to study the uniform distribution over the square $[0, 1[^2$. Two samples of size 50 are generated by goLHS for small samples, also two samples of size 5,000 are generated for large samples.

We can clearly see from Figures 12 and 13 that, the LHS results are spread out and show no group effects, which means that the uniformity of goLHS streams is demonstrated, mainly in the illustration of Figure 13.

6.6.3 The representation of *ecdf* of LHS and *cdf* of $U_{[0,1[}$

To confirm the results obtained in both Subsections 6.6.1 and 6.6.2, we plot the *ecdf* of the sample of size 100 generated from goLHS and the *cdf* of $U_{[0,1[}$.

Figure 14 The representation of the 'ecdf' of LHS and the 'cdf' of $U_{[0,1[}$ (see online version for colours)

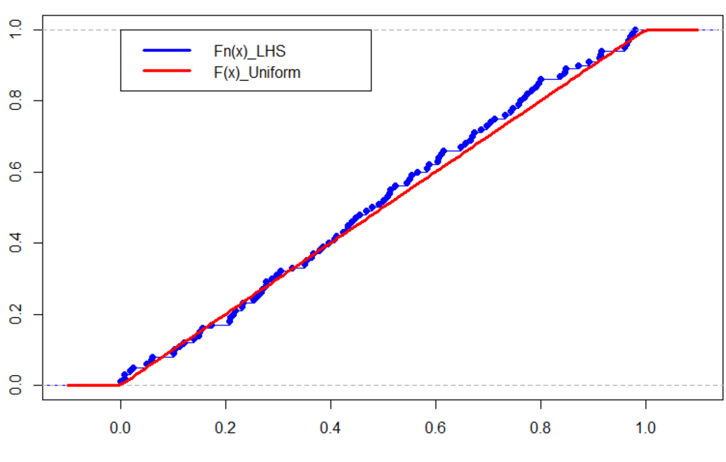


Figure 14 shows that there is a clear match between the *ecdf* of goLHS and the *cdf* of $U_{[0,1[}$.

6.7 Testing the independence

The independence is tested through the runs test which is available on R software with the Randtests[1] package. The runs test is used to *see* if the elements of a sequence are mutually independent, so, the hypothesis H_0 to be tested is the independence of the goLHS stream.

Table 5 The results of the runs test provided by a goLHS stream of 4,600 numbers

—	*Runs statistic*	*p-value*
LHS	0.11797	0.9061

The results of Table 5 show that the independence for LHS sequence is statistically no significant according to the p-value obtained in the test which is too greater than 0.05. Thus, the generated sample is independent.

7 Comparing simulation results of the proposed queueing system using the MC and LHS methods

In this section, we present in Table 6 the simulation results of the proposed model, obtained by using both the 'rand' and 'goLHS' generators. When the simulation is carried out, the size of sub-samples n_j is generated between min = 40 and max = 90, the simulation runs are repeated 1,000 times, the simulation period is set to 150 unit of times for each replicated run. The other parameters are chosen such as λ = 3, μ = 9, β = 12, α = 6, ν = 5, ν' = 4, p = 0.5 and q = 0.3.

Table 6 The simulation results of our queue with the two generators goLHS and rand

—	*K = 1*		*K = 2*		*K = 3*		*K = 4*	
	LHS	*MC*	*LHS*	*MC*	*LHS*	*MC*	*LHS*	*MC*
$mean(\bar{N})$	0.9783	1.0767	1.1472	0.1516	1.3538	1.3487	1.5769	1.5879
$var(\bar{N})$	0.0274	0.0355	0.0077	0.0086	0.0091	0.0116	0.0127	0.0142
$mean(\bar{Y})$	0.6160	0.6737	0.7405	0.7487	0.9907	0.9856	1.2612	1.2618
$var(\bar{Y})$	0.0264	0.0350	0.0063	0.0069	0.0070	0.0094	0.0101	0.0117

Table 7 Some examples of the simulation of our queue with goLHS generator

—	*K = 5*	*K = 10*	*K = 15*	*K = 20*
$mean(\bar{N})$	1.7853	2.4588	2.8313	3.0138
$var(\bar{N})$	0.0182	0.0581	0.1206	0.1698
$mean(\bar{Y})$	1.5075	2.3258	2.7681	2.9791
$var(\bar{Y})$	0.0150	0.0509	0.1114	0.1620

It is deduced from Table 6 that the simulation estimates are all closer to the theoretical values (see Table 1) whatever the used generator. On the other hand, LHS reduces the variance of all simulation estimates compared to the MC method. Consequently, we deduce that goLHS generator is better than the 'rand' generator integrated in MATLAB especially on queueing systems.

However, we notice from Table 6 that the greater the batch size K, the more the simulation results are closer to the theoretical values. The developed software is therefore efficient and we can use it in computing all performance measures of the system regardless of the batch size K. The theoretical formulas of our queueing system are complicated and hard to implement whenever the batch size K increases, so, we recommend to use the developed simulation software to carry out the performance measures of these types of queues.

Some instances of simulation are then given in Table 7 for K = 5, 10, 15 and 20 using the best goLHS generator to sample inputs.

8 Conclusions

A new model of batch service lateness queueing system with two different Bernoulli catastrophes under multiple vacations policy was proposed, where its steady state probabilities were derived together with some performance measures by using the appropriate generating functions.

Numerical examples showing the effect of batch size on the mean numbers of customers in the system and in the queue are provided, both of which increase with increasing batch size. An increase in the batch size reduces the probability of the busy period resulting in a decrease in the average rate of loss of customers during service, while, the probability of the period of vacation and the average rate of loss of customers during the vacation increase significantly, and this is due to the fact that the server is more prone to catastrophes during this period and as a result more customers are lost.

Illustrations of the mean number of customers in the system against the catastrophe and control parameters indicate that the former decreases whenever the latter increase. In addition, we conclude through all the illustrations, that in this model, the occurrence of catastrophes on vacation has a greater influence on the system than the occurrence of catastrophes in the service.

In addition, the theoretical formulas of our model seem too complicated and hard to implement which proves the limitations of the research but this is taken into account by the simulation. Indeed, the simulation experiments performed on the proposed model using the software developed under MATLAB by the MC and LHS methods have shown good results compared to the theoretical values. However, it is also demonstrated that the results of the proposed goLHS generator are more accurate and efficient than those of the MC method.

Therefore, the obtained results could be applied to estimate the losses that may occur due to a catastrophic event in this type of systems. Moreover, the developed goLHS is proved to be a good generator, so, the developed software using goLHS can be safely used to compute performance measures of systems in which their theoretical formulas are complicated and hard to implement, including those with batch queues, especially for large batch size in order to accelerate and facilitate the study of queues.

The future extension of this research work is that the proposed model can be extended to a queueing model in which both arrivals and service are in batches instead of just batch lateness service, which is rarely studied due to its very complex equations. The simulation can also be used to help the user of such system to develop performances. Another future research direction is the possibility to apply the obtained results for queues with catastrophes and priorities.

References

Armony, M., Shimkin, N. and Whitt, W. (2009) 'The impact of delay announcements in many-server queues with abandonment', *Operations Research*, Vol. 57, No. 1, pp.66–81.

Bailey, N.T. (1954) 'On queueing processes with bulk service', *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 16, No. 1, pp.80–87.

Bell, I.H. (2019) 'Cego: C++ 11 evolutionary global optimization', *Journal of Open Source Software*, Vol. 4, No. 36, p.1147.

Blanchet, J., Dong, J. and Pei, Y. (2018) 'Perfect sampling of $GI/GI/c$ queues', *Queueing Systems*, Vol. 90, No. 1, pp.1–33.

Boubalou, M., Ourbih-Tari, M., Aloui, A. and Zioui, A. (2019) 'Comparing $M/G/1$ queue estimators in Monte Carlo simulation through the tested generator getRDS and the proposed getLHS using variance reduction', *Monte Carlo Methods and Applications*, Vol. 25, No. 2, pp.177–186.

Chae, K.C., Park, H.M. and Yang, W.S. (2010) 'A $GI/Geo/1$ queue with negative and positive customers', *Applied Mathematical Modelling*, Vol. 34, No. 6, pp.1662–1671.

Chao, X. (1995) 'A queueing network model with catastrophes and product form solution', *Operations Research Letters*, Vol. 18, No. 2, pp.75–79.

Chen, A. and Renshaw, E. (1997) 'The $M/M/1$ queue with mass exodus and mass arrivals when empty', *Journal of Applied Probability*, Vol. 34, No. 1, pp.192–207.

Di Crescenzo, A., Giorno, V., Nobile, A.G. and Ricciardi, L.M. (2003) 'On the $M/M/1$ queue with catastrophes and its continuous approximation', *Queueing Systems*, Vol. 43, No. 4, pp.329–347.

Doshi, B.T. (1986) 'Queueing systems with vacations-a survey', *Queueing Systems*, Vol. 1, No. 1, pp.29–66.

Gelenbe, E., Glynn, P. and Sigman, K. (1991) 'Queues with negative arrivals', *Journal of Applied Probability*, Vol. 28, No. 1, pp.245–250.

Gelenbe, E. (1989) 'Random neural networks with negative and positive signals and product form solution', *Neural Computation*, Vol. 1, No. 4, pp.502–510.

Helton, J.C. and Davis, F.J. (2003) 'Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems', *Reliability Engineering & System Safety*, Vol. 81, No. 1, pp.23–69.

Iman, R.L. and Conover, W. (1980) 'Small sample sensitivity analysis techniques for computer models. with an application to risk assessment', *Communications in Statistics-Theory and Methods*, Vol. 9, No. 17, pp.1749–1842.

Kumar, B.K. and Arivudainambi, D. (2000) 'Transient solution of an $M/M/1$ queue with catastrophes', *Computers & Mathematics with Applications*, Vol. 40, Nos. 10–11, pp.1233–1240.

Kumar, B.K. and Madheswari, S.P. (2002) 'Transient behaviour of the $M/M/2$ queue with catastrophes', *Statistica*, Vol. 62, No. 1, pp.129–136.

Laxmi, P.V. and Kassahun, T.W. (2020) 'Transient analysis of multi-server Markovian queueing system with synchronous multiple working vacations and impatience of customers', *International Journal of Mathematics in Operational Research*, Vol. 16, No. 2, pp.217–237.

Lucantoni, D.M., Meier-Hellstern, K.S. and Neuts, M.F. (1990) 'A single-server queue with server vacations and a class of non-renewal arrival processes', *Advances in Applied Probability*, Vol. 22, No. 3, pp.676–705.

Manurung, J. (2019) 'Application of FIFO algorithm (first in first out) to simulation queue', *INFOKUM*, Vol. 7, No. 2, pp.44–47.

McKay, M.D., Beckman, R.J. and Conover, W.J. (1979) 'A comparison of three methods for selecting values of input variables in the analysis of output from a computer code', *Technometrics*, Vol. 21, No. 2, pp.239–245.

Mishra, P.P., Poongodi, T., Yadav, S.K. and Mishra, S.S. (2021) 'Algorithmic approach to time-cost analysis of queued commodity flowing through critical path', *International Journal of Mathematics in Operational Research*, Vol. 18, No. 2, pp.169–186.

Ourbih-Tari, M., Guebli, S. and Aloui, A. (2019) 'Applying refined descriptive sampling on the vibrating string model', *International Journal of Computing Science and Mathematics*, Vol. 10, No. 3, pp.276–287.

Priya, R.S. and Sudhesh, R. (2018) 'Transient analysis of a discrete-time infinite server queue with system disaster', *International Journal of Mathematics in Operational Research*, Vol. 12, No. 1, pp.91–101.

Saggou, H., Sadeg, I., Ourbih-Tari, M. and Bourennane, E.B. (2019) 'The analysis of unreliable $M^{[X]}/G/1$ queuing system with loss, vacation and two delays of verification', *Communications in Statistics-Simulation and Computation*, Vol. 48, No. 5, pp.1366–1381.

Shen, H., Hong, L.J. and Zhang, X. (2018) 'Enhancing stochastic kriging for queueing simulation with stylized models', *IISE Transactions*, Vol. 50, No. 11, pp.943–958.

Stein, M. (1987) 'Large sample properties of simulations using Latin hypercube sampling', *Technometrics*, Vol. 29, No. 2, pp.143–151.

Stidham Jr., S. (1974) 'Stochastic clearing systems', *Stochastic Processes and their Applications*, Vol. 2, No. 1, pp.85–113.

Sudhesh, R. (2010) 'Transient analysis of a queue with system disasters and customer impatience', *Queueing Systems*, Vol. 66, No. 1, pp.95–105.

Sun, K. and Wang, J. (2019) 'Equilibrium joining strategies in the single server queues with negative customers', *International Journal of Computer Mathematics*, Vol. 96, No. 6, pp.1169–1191.

Tian, N. and Zhang, Z.G. (2002) 'The discrete-time $GI/Geo/1$ queue with multiple vacations', *Queueing Systems*, Vol. 40, No. 3, pp.283–294.

Vijayashree, K.V. and Anjuka, A. (2019) 'Exact stationary solution for a fluid queue driven by an $M/M/1$ queue with disaster and subsequent repair', *International Journal of Mathematics in Operational Research*, Vol. 15, No. 1, pp.92–109.

Wu, K., McGinnis, L.F. and Zwart, B. (2010) 'Approximating the performance of a batch service queue using the $M/M^k/1$ model', *IEEE Transactions on Automation Science and Engineering*, Vol. 8, No. 1, pp.95–102.

Yang, W.S., Kim, J.D. and Chae, K.C. (2002) 'Analysis of $M/G/1$ stochastic clearing systems', *Stochastic Analysis and Applications*, Vol. 20, No. 5, pp.1083–1100.

Yue, D., Yue, W. and Zhao, G. (2014) 'Analysis of an $M/M/c$ queueing system with impatient customers and synchronous vacations', *Journal of Applied Mathematics*, Vol. 2014, Article ID 893094, pp.1–11 [online] http://dx.doi.org/10.1155/2014/893094.

Zhang, L. and Li, J. (2015) 'The $M/M/c$ queue with mass exodus and mass arrivals when empty', *Journal of Applied Probability*, Vol. 52, No. 4, pp.990–1002.

Zidani, N. and Djellab, N. (2018) 'On the multiserver retrial queues with negative arrivals', *International Journal of Mathematics in Operational Research*, Vol. 13, No. 2, pp.219–242.

Notes

1 Tools for Biostatistics, Public Policy, and Law available in https://CRAN.R-project.org/package=randtests.