mathematics

Article

Multiserver Retrial Queue with Two-Way Communication and Synchronous Working Vacation

Tzu-Hsin Liu [1], Kuo-Ching Chiou [1], Chih-Ming Chen [2,*] and Fu-Min Chang [1,*]

[1] Department of Finance, Chaoyang University of Technology, 168, Jifeng East Road, Wufeng District, Taichung City 41349, Taiwan; t2011119@cyut.edu.tw (T.-H.L.); kcchiou@cyut.edu.tw (K.-C.C.)
[2] Ph.D. Program of Business Administration in Industrial Development, Department of Business Administration, Chaoyang University of Technology, 168, Jifeng East Road, Wufeng District, Taichung City 41349, Taiwan
* Correspondence: tcming0170@gmail.com (C.-M.C.); fmchang@cyut.edu.tw (F.-M.C.)

Abstract: This work investigates a two-way communication retrial queue with synchronous working vacation and a constant retrial policy. During the idle time, a server makes an outgoing call after a random length. The service time of the incoming call and outgoing call obeys exponential distribution with different rates. If the incoming call finds all servers to be unavailable, it may or may not enter orbit. All servers immediately go on vacation simultaneously as soon as they find an empty system after the service finishes. During vacation, the servers can provide a service to those incoming calls, but this is at a lower-speed rate. The stationary probability distribution and the ergodic condition are obtained utilizing the matrix geometric technique. Some system characteristics are developed. Using MATLAB software, the variation in average orbit length, idle ratio, and the average number of servers in different server states is plotted for different values of the incoming/outgoing call rate and retrial rate. We further propose a multi-objective optimization model from which the optimal rate of outgoing calls and optimal vacation rate are explicitly obtained.

Keywords: two-way communication; synchronous working vacation; retrial queue; optimization

MSC: 60J25

check for updates

Citation: Liu, T.-H.; Chiou, K.-C.; Chen, C.-M.; Chang, F.-M. Multiserver Retrial Queue with Two-Way Communication and Synchronous Working Vacation. *Mathematics* **2024**, *12*, 1163. https://doi.org/10.3390/math12081163

Academic Editor: Quanxin Zhu

Received: 18 March 2024
Revised: 6 April 2024
Accepted: 11 April 2024
Published: 12 April 2024

1. Introduction

This paper deals with investigations into retrial queue with two-way communication and synchronous working vacation. For more information about retrial queues, please refer to [1–5]. The main reason for this is that retrial queues provide a suitable model for the performance evaluation of computer networks, communications systems, and call centers. The characteristic of the retrial queue is that an arriving customer who cannot be serviced immediately joins an orbit and tries the request again after a period of time. Therefore, the analysis of a retrial queue is more difficult than that of its counterpart model without retrial.

In most of the literature on retrial queue, servers only serve incoming arrivals. Servers wait for the next arrival (from outside or orbit) after completing service. However, in some real-world situations, there is an opportunity for servers to make an outgoing phone call at their idle state. This is especially true for a business organization, such as a call center, where agents not only handle incoming calls but can also make calls to sell, advertise, and promote the business's products and services. A characteristic of two-way communication is the fact that unoccupied servers can perform outgoing calls to the source. In such systems, the server's utilization is always a critical issue; for an example, see [6–9]. Ayyappan and Gowthami [10] performed a stationary analysis of a feedback retrial queue with impatient customers, vacations, and two types of arrivals. Lee et al. [11] analyzed the waiting time distribution of a two-way communication retrial queue. Sztrik et al. [12] examined the unreliable operation of a finite-source two-way communication retrial queue and compared the different distribution of failure time on performance measures. Tóth and Sztrik [13] studied a finite-source retrial queue with two-way communication.

Since the vacation queue reflects the situation of servers utilizing this idle time for different purposes like doing supplementary jobs, equipment maintenance, testing, and so on, it has been extensively studied for modeling and analyzing practical problems, including communication networks, call centers, service centers, production lines, manufacturing systems, and more. A special class of vacation queues includes working vacation, in which the server provides a service at a lower rate rather than stopping working completely. A work on retrial queue with working vacation and starting failure was conducted by Yang and Wu [14]. Li et al. [15] analyzed the retrial queue with Bernoulli working vacation interruption. A retrial queue with general retrial times and single working vacation was studied by Li et al. [16] by utilizing the supplementary variable method. Gupta and Kumar [17] studied a retrial queue with a single waiting server subject to breakdown and repair under working vacation and interruption. In the work of Muthusamy et al. [18], they considered a working vacation retrial queue with three different classes of customers and optional reservice during working vacations in the Bernoulli schedule. Pazhani et al. [19] used the supplementary variable technique to obtain the probability generating function for the number of customers and the mean number of customers in the invisible waiting area of a retrial queue with working vacation and a single waiting server. Shanmugam and Saravanarajan [20] investigated an unreliable retrial queueing system with working vacation. The orbit and system lengths were derived through the supplementary variable method. Chen et al. [21] studied a single server retrial queue incorporating random working vacation and improved service efficiency during vacation policies and examined its optimal queueing strategies. Sundararaman et al. [22] examined a new type of queue in two waiting queues (original queue and orbit queue) with working vacations.

In these articles mentioned above, the authors mostly focus on a single server. However, real systems such as telecommunications or call centers are often multiserver rather than single servers. There is limited research on multiserver two-way communication retrial queues in the literature because the analysis of multiserver two-way communication retrial queues is more complex than single-server queues. In this paper, we consider a multiserver retrial queue with two-way communication with working vacation. Our advantages and contributions are as follows. We carry out an extensive analysis for a multiserver two-way communication retrial queue with synchronous working vacation. With the support of the matrix geometric technique, the steady-state probabilities and ergodic condition are derived. The effect of the parameters on the system characteristics is displayed numerically. We also construct a multi-objective optimization analysis, from which the optimal rate of outgoing calls and optimal vacation rate are explicitly obtained.

The paper structure is as follows. The description of the model and the balance equations governing the system's behavior are given in Section 2. A detailed system analysis, including the expressions for the stationary probability distribution and the ergodicity condition, is presented in Section 3. In Section 4, the system characteristics are demonstrated. Section 5 provides some numerical illustrations followed by a concluding remark.

2. System Description and Mathematical Model

This section lists the assumptions made in this paper. This allows for the formulation of a set of balance equations that gives a starting point for the system analysis.

2.1. System Description

We consider a multiserver retrial queue in which incoming calls follow a Poisson arrival process with rate λ. The service times of an incoming call are exponentially distributed with rate μ_1. If an incoming call finds that the server is fully occupied, it may or may not enter orbit, and if it does, it retries to seek service after an exponentially distributed time at rate σ. Within the orbit, we apply a constant retrial policy, i.e., only the call at the head of orbit can request service. Suppose that an incoming call goes into orbit with probability b.

Otherwise, the incoming call starts service immediately. During server's idle time, it makes an outgoing call with rate α and provides the service for an exponentially distributed time with rate μ_2. We denote s as the number of servers in the system. All the servers take the vacation simultaneously when they find the system empty, and vacation duration is also exponentially distributed with parameter θ. During the vacation period, the servers can serve incoming calls at a low service rate μ_3, but they will not make outgoing calls. The work flow of the two-way communication retrial queue with synchronous working vacation is described in Figure 1.

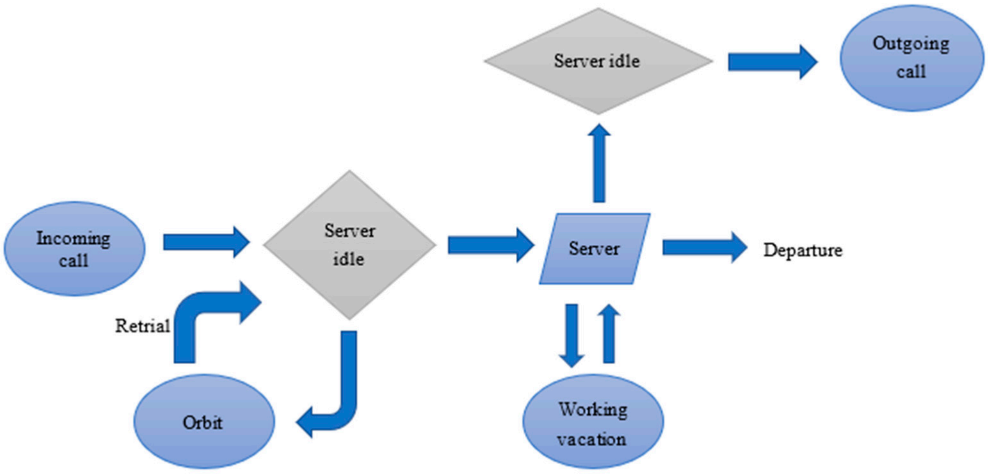

Figure 1. Two-way communication retrial queue with synchronous working vacation.

2.2. Markov Chain and Balance Equations

We denote the number of servers busy serving incoming and outgoing calls at time t by $n_1(t)$ and $n_2(t)$, respectively. Also, $C(t)$ and $M(t)$ are the server state and the number of customers in orbit at time t, respectively, where

$$C(t) = \begin{cases} 0, & \text{all servers are on vacation at time } t, \\ 1, & \text{no servers are on vacation at time } t. \end{cases}$$

Thus, the process $\{(n_1(t), n_2(t), \mathrm{C}(t), \mathrm{M}(t)); t \geq 0\}$ is a continuous time Markov chain on the state space $\Omega = \{(i,j,k,n) : 0 \leq i \leq s, j = 0, k = 0, n \geq 0\} \cup \{(i,j,k,n) : 0 \leq i + j \leq s, k = 1, n \geq 0\}$.

Let $\pi_{i,j,k}(n) = \lim_{t\to\infty} \mathrm{P}\{(n_1(t),\ n_2(t),\ \mathrm{C}(t),\ \mathrm{M}(t))\}$ be the limiting probabilities for the steady-state distribution. The system of governing steady-state equations is framed as follows:

$$(\lambda+\theta)\pi_{0,0,0}(0) = \mu_3\pi_{1,0,0}(0) + \mu_1\pi_{1,0,1}(0) + \mu_2\pi_{0,1,1}(0), \tag{1}$$

$$(\lambda+s\alpha)\pi_{0,0,1}(0) = \theta\pi_{0,0,0}(0), \tag{2}$$

$$(\lambda+\theta+\sigma)\pi_{i,0,1}(n) = \mu_3\pi_{1,0,0}(n),\ n \geq 1, \tag{3}$$

$$(\lambda+s\alpha+\sigma)\pi_{i,0,1}(n) = \mu_1\pi_{1,0,1}(n) + \mu_2\pi_{0,1,1}(n) + \theta\pi_{0,0,0}(n), n \geq 1, \tag{4}$$

$$\begin{gathered}(\lambda+i\mu_3+\theta+(1-\delta_{n,0})\sigma)\pi_{i,0,0}(n) = \\ \lambda\pi_{i-1,0,0}(n) + (i+1)\mu_3\pi_{i+1,0,0}(n)\ +\ \sigma\pi_{i-1,0,0}(n+1), \\ 1 \leq i \leq s-1,\ n \geq 0,\end{gathered} \tag{5}$$

$$\begin{gathered}(\lambda+i\mu_1+(s-i)\alpha+(1-\delta_{n,0})\sigma)\pi_{i,0,1}(n) = \\ \lambda\pi_{i-1,0,1}(n) + (i+1)\mu_1\pi_{i+1,0,1}(n) + \mu_2\pi_{i,1,1}(n) + \theta\pi_{i,0,0}(n) + \sigma\pi_{i-1,0,1}(n+1), \\ 1 \leq i \leq s-1,\ n \geq 0,\end{gathered} \tag{6}$$

$$\begin{gathered}(\lambda + i\mu_1 + j\mu_2 + (s-i-j)\alpha + (1-\delta_{n,0})\sigma)\pi_{i,j,1}(n) = \\ (s-i-j+1)\alpha\pi_{i,j-1,1}(n) + \lambda\pi_{i-1,j,1}(n) + (i+1)\mu_1\pi_{i+1,j,1}(n) + \\ (j+1)\mu_2\pi_{i,j+1,1}(n) + \sigma\pi_{i-1,j,1}(n+1), \\ 0 \le i \le s-j-1,\ 1 \le j \le s-1,\ n \ge 0,\end{gathered} \tag{7}$$

$$\begin{gathered}(b\lambda + s\mu_3 + \theta)\pi_{s,0,0}(n) = \\ b\lambda\pi_{s,0,0}(n-1) + \lambda\pi_{s-1,0,1}(n) + \sigma\pi_{s-1,0,0}(n+1), \\ n \ge 0,\end{gathered} \tag{8}$$

$$\begin{gathered}(b\lambda + (s-j)\mu_1 + j\mu_2)\pi_{s-j,j,1}(n) = \\ b\lambda\pi_{s-j,j,1}(n-1) + \alpha\pi_{s-j,j-1,1}(n) + \\ \lambda\pi_{s-j-1,j,1}(n) + \theta\delta_{j,0}\pi_{s-j,j,0}(n) + \sigma\pi_{s-j-1,j,1}(n+1), \\ 0 \le j \le s,\ n \ge 0.\end{gathered} \tag{9}$$

The normalizing condition is

$$\sum_{n=0}^{\infty}\sum_{i=0}^{s}\pi_{i,0,0}(n) + \sum_{n=0}^{\infty}\sum_{i=0}^{s}\sum_{j=0}^{s-i}\pi_{i,j,1}(n) = 1, \tag{10}$$

where $\delta_{n,0}$ represents Kronecker's delta and $\pi_{i,j,k}(n) = 0$ if $(i,\ j,\ k,\ n) \notin \Omega$.

3. System Analysis

This section provides a system analysis. First, a quasi-birth-and-death process formulation is provided to identify all the components of the involved infinitesimal generator and block matrices. Next, the steady-state probabilities are derived in matrix form, and ergodic condition is exported.

3.1. Infinitesimal Generator and Matrices

According to the matrix geometric method, the infinitesimal generator of the continuous time Markov process $\{(n_1(t),\ n_2(t),\ \mathrm{C}(t),\ \mathrm{M}(t)); t \ge 0\}$ is written in the form of a block matrix $\mathbf{Q}$ as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{G}^0 & \mathbf{G}^u & & \\ \mathbf{G}^l & \mathbf{G} & \mathbf{G}^u & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix}, \tag{11}$$

where $\mathbf{G}^0$, $\mathbf{G}^u$, $\mathbf{G}^l$, and $\mathbf{G}$ are square matrices of order $(s + 1)(s + 4)/2$.

For the concerned model, we can represent the block structures of these matrices as

$$\mathbf{G}^u = \begin{bmatrix} \mathbf{G}_0^u & & & \\ & \mathbf{G}_1^u & & \\ & & \ddots & \\ & & & \mathbf{G}_s^u \end{bmatrix},$$

$$\mathbf{G}^l = \begin{bmatrix} \mathbf{G}_0^l & & & \\ & \mathbf{G}_1^l & & \\ & & \ddots & \\ & & & \mathbf{G}_s^l \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & & & & \\ & \mathbf{A}_0 & \mathbf{A}_0^+ & & & \\ & \mathbf{A}_1^- & \mathbf{A}_1 & \mathbf{A}_1^+ & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{A}_{s-1}^- & \mathbf{A}_{s-1} & \mathbf{A}_{s-1}^+ \\ & & & & \mathbf{A}_s^- & \mathbf{A}_S \end{bmatrix},$$

where $\mathbf{G}_j^u$ $(0 \le j \le s)$, $\mathbf{G}_j^l$ $(0 \le j \le s)$, $\mathbf{B}_0$, $\mathbf{B}_1$, $\mathbf{A}_j$ $(0 \le j \le s)$, $\mathbf{A}_j^+$ $(0 \le j \le s-1)$, and $\mathbf{A}_j^-$ $(1 \le j \le s)$ are $(s - j + 1) \times (s - j + 1)$, $(s - j + 1) \times (s - j + 1)$, $(s + 1) \times (s + 1)$, $(s + 1) \times (s + 1)$, $(s - j + 1) \times (s - j + 1)$, $(s - j + 1) \times (s - j)$, and $(s - j + 1) \times (s - j + 2)$ matrices, respectively, with entries given by

$$\mathbf{G}_j^u[k,k\prime] = \begin{cases} b\lambda, \; k' = k = s - j + 1, \\ 0, \; \text{in other cases;} \end{cases}$$

$$\mathbf{G}_j^l[k,k\prime] = \begin{cases} \sigma, \; k' = k + 1, \; 1 \le k \le s - j, \\ 0, \; \text{in other cases;} \end{cases}$$

$$\mathbf{B}_0[k,k'] = \begin{cases} \lambda, & k' = k + 1, \; 1 \le k \le s, \\ (k-1)\mu_3, & k' = k - 1, \; 2 \le k \le s + 1, \\ -(\lambda + (k-1)\mu_3 + \theta + \sigma), & k' = k, \; 1 \le k \le s, \\ -(b\lambda + s\mu_3 + \theta), & k' = k = s + 1, \\ 0, & \text{in other cases;} \end{cases}$$

$$\mathbf{B}_1[k,k\prime] = \begin{cases} \theta, \; k' = k, \; 1 \le k \le s + 1, \\ 0, \; \text{in other cases;} \end{cases}$$

$$\mathbf{A}_j^+[k,k\prime] = \begin{cases} (s - j + 1 - k)\alpha, \; k' = k, \; 1 \le k \le s - j, \\ 0, \quad \text{in other cases;} \end{cases}$$

$$\mathbf{A}_j[k,k\prime] = \begin{cases} \lambda, & k' = k + 1, \; 1 \le k \le s - j, \\ k\mu_1, & k' = k - 1, \; 2 \le k \le s - j + 1, \\ -(\lambda + (k-1)\mu_1 + j\mu_2 + (s - j + 1 - k)\alpha + \sigma), & k' = k, \; 1 \le k \le s, \\ -(b\lambda + (s-j)\mu_1 + j\mu_2), & k' = k = s + 1, \\ 0, & \text{in other cases;} \end{cases}$$

$$\mathbf{A}_j^-[k,k\prime] = \begin{cases} j\mu_2, \; k' = k, \; 1 \le k \le s - j + 1, \\ 0, \; \text{in other cases.} \end{cases}$$

The matrix $\mathbf{G}^0$ is the same as $\mathbf{G}$, but the element of $\mathbf{G}^0[s+3, \; s+2]$ is shifted to the position of $\mathbf{G}^0[s+3, \; 1]$, the element of $\mathbf{G}^0[2s+3, \; s+2]$ is shifted to the position of $\mathbf{G}^0[2s+3, 1]$, and the retrial rate σ is ignored.

3.2. Stationary Distribution

Assume that $\mathbf{\Pi} = [\mathbf{\Pi}(0), \quad \mathbf{\Pi}(1), \quad \cdots]$ represents the steady-state probability vector of $\mathbf{Q}$. Also, $\mathbf{\Pi}(n) = [\pi_{0,0,0}(n) \quad \cdots \quad \pi_{s,0,0}(n) \quad \boldsymbol{\pi}_0(n) \quad \cdots \quad \boldsymbol{\pi}_s(n)]$ and $\boldsymbol{\pi}_j(n) = [\pi_{0,j,1}(n) \quad \cdots \quad \pi_{s-j,j,1}(n)]$ for $0 \le j \le s$.

We can rewrite Equations (1)–(10) in the matrix form as $\mathbf{\Pi Q}$ = 0 and $\sum_{n=0}^{\infty} \mathbf{\Pi}(n)\mathbf{e} = 1$, where $\mathbf{0}$ and $\mathbf{e}$ represent a row and a column vector with an appropriate size with all zero and all one entries, respectively.

Note that from Neuts [23], there is a matrix $\mathbf{R}$, such that $\mathbf{\Pi}(n) = \mathbf{\Pi}(n-1)\mathbf{R}$. This implies that $\mathbf{\Pi}(n) = \mathbf{\Pi}(0)\mathbf{R}^n$. Here, $\mathbf{R}$ is the minimal non-negative solution of

$$\mathbf{G}^u + \mathbf{RG} + \mathbf{R}^2\mathbf{G}^l = 0, \tag{12}$$

with a spectral radius less than one. Since it is difficult for $\mathbf{R}$ to obtain the explicit expression by solving Equation (12), researchers could use different methods such as a successive substitution approach to evaluate **R**. In the numerical illustration, the following procedure will be implemented to approximate **R**.

We can define **I** as an identity matrix. The first equation of **ΠQ** = 0 can be written as the following by replacing $\mathbf{\Pi}(1)$ with $\mathbf{\Pi}(1) = \mathbf{\Pi}(0)\mathbf{R}$:

$$\mathbf{\Pi}(0)\mathbf{G}^0 + \mathbf{\Pi}(0)\mathbf{R}\mathbf{G}^l = 0. \quad (13)$$

And $\sum_{n=0}^{\infty} \mathbf{\Pi}(n)\mathbf{e} = 1$ can be re-written as

$$\sum_{n=0}^{\infty} \mathbf{\Pi}(n)\mathbf{e} = \mathbf{\Pi}(0)(\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} = 1, \quad (14)$$

where **I** is an identity matrix. These two equations can be used to determine **Π**(0).

3.3. Ergodicity

Consider that the matrix $\mathbf{H} = \mathbf{G}^u + \mathbf{G} + \mathbf{G}^l$=. Let $\mathbf{p} = \begin{bmatrix} \mathbf{p}_{0,0} & \mathbf{p}_{0,1} & \cdots & \mathbf{p}_{s,1} \end{bmatrix}$ is the invariant probability vector of **H**, where $\mathbf{p}_{0,0} = \begin{bmatrix} p_{0,0,0} & p_{1,0,0} & \cdots & p_{s,0,0} \end{bmatrix}$ and $\mathbf{p}_{j,1} = \begin{bmatrix} p_{0,j,1} & p_{1,j,1} & \cdots & p_{s-j,j,1} \end{bmatrix}$ $0 \le \text{j} \le \text{s}$. Therefore, **p** satisfies the following criteria: **pH** = 0 and **pe** = 1. The condition for the ergodicity of the Markov chain $\{(n_1(t), n_2(t), \mathrm{C}(t)); t \ge 0\}$ is $\mathbf{p}\mathbf{G}^u\mathbf{e} < \mathbf{p}\mathbf{G}^l\mathbf{e}$. The ergodicity condition can be transformed to

$$(b\lambda + \sigma)P_B^{loss} < \sigma, \quad (15)$$

where $P_B^{loss} = \sum_{j=0}^{s} p_{s-j,j,1}$ denotes the blocking probability of the corresponding loss system.

The ergodicity condition proposed by $\mathbf{p}\mathbf{G}^u\mathbf{e} < \mathbf{p}\mathbf{G}^l\mathbf{e}$ itself is unambiguous because the number of states for $\{(n_1(t),\ n_2(t),\ \mathrm{C}(t)); t \ge 0\}$ is finite. However, it does not seem to be easy to obtain a simple scalar form in terms of the given parameters. Below, a simple explicitness is obtained from (15) for the special case of s = 1. For the matrices, we have

$$\mathbf{G}^u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & b\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b\lambda & 0 \\ 0 & 0 & 0 & 0 & b\lambda \end{bmatrix},$$

$$\mathbf{G}^l = \begin{bmatrix} 0 & \sigma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} -(\lambda+\theta+\sigma) & \lambda & \theta & 0 & 0 \\ \mu_3 & -(b\lambda+\mu_3+\theta) & 0 & \theta & 0 \\ 0 & 0 & -(\lambda+\alpha+\sigma) & \lambda & \alpha \\ 0 & 0 & \mu_1 & -(b\lambda+\mu_1) & 0 \\ 0 & 0 & \mu_2 & 0 & -(b\lambda+\mu_2) \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} -(\lambda+\theta+\sigma) & \lambda+\sigma & \theta & 0 & 0 \\ \mu_3 & -(\mu_3+\theta) & 0 & \theta & 0 \\ 0 & 0 & -(\lambda+\alpha+\sigma) & \lambda+\sigma & \alpha \\ 0 & 0 & \mu_1 & -\mu_1 & 0 \\ 0 & 0 & \mu_2 & 0 & -\mu_2 \end{bmatrix}.$$

We have $p_{0,0,0} = 0$, $p_{1,0,0} = 0$, $p_{0,1,1} = (\boldsymbol{\lambda} + \boldsymbol{\sigma})p_{0,0,1}/\boldsymbol{\mu}_1$, and $p_{1,0,1} = \alpha p_{0,0,1}/\boldsymbol{\mu}_2$. Hence, the ergodicity condition is

$$b\lambda\left(\frac{\boldsymbol{\lambda}+\boldsymbol{\sigma}}{\boldsymbol{\mu}_1} + \frac{\boldsymbol{\alpha}}{\boldsymbol{\mu}_2}\right) < \sigma.$$

If we set $b = 1$, then the ergodicity condition is consistent with Equation (15) in Phung-Duc and Rogiest [24].

4. System Characteristics and Cost Function

We compute system characteristics and average operating cost in terms of the steady-state probabilities in this section.

4.1. System Characteristics

If the system is in stability condition, we can obtain the expressions of the system characteristics listed below.

The average number of calls in orbit (average orbit length) is

$$E[N] = \sum_{n=0}^{\infty} n\left(\sum_{i=0}^{s} \pi_{i,0,0}(n)\right) + \sum_{n=0}^{\infty} n\left(\sum_{j=0}^{s} \sum_{i=0}^{s-j} \pi_{i,j,1}(n)\right).$$

By Little's Law, the average waiting time of a call in orbit is

$$E[W] = \frac{E[N]}{b\lambda} = \frac{1}{b\lambda} \sum_{n=0}^{\infty} n\left(\sum_{i=0}^{s} \pi_{i,0,0}(n)\right) + \sum_{n=0}^{\infty} n\left(\sum_{j=0}^{s} \sum_{i=0}^{s-j} \pi_{i,j,1}(n)\right).$$

The average number of incoming calls during the vacation period is

$$E[S_0] = \sum_{n=0}^{\infty} \sum_{i=0}^{s} i \times \pi_{i,0,0}(n).$$

The average number of incoming calls during the non-vacation period is

$$E[S_1] = \sum_{n=0}^{\infty} \sum_{j=0}^{s} \sum_{i=0}^{s-j} i \times \pi_{i,j,1}(n).$$

The average number of outgoing calls during the non-vacation period is

$$E[S_2] = \sum_{n=0}^{\infty} \sum_{j=0}^{s} j \sum_{i=0}^{s-j} \pi_{i,j,1}(n).$$

The average number of servers being busy during the non-vacation period is defined as

$$E[B] = \sum_{n=0}^{\infty} \sum_{j=0}^{s} \sum_{i=0}^{s-j} (i+j)\pi_{i,j,1}(n) = E[S_1] + E[S_2].$$

The average number of servers on vacation, whether busy or not, is calculated as

$$E[V] = \sum_{n=0}^{\infty} s\left(\sum_{i=0}^{s} \pi_{i,0,0}(n)\right).$$

The average number of idle servers is calculated as

$$E[I] = s - E[B] - E[V].$$

The server utilization at the steady state is

$$U = \frac{E[S_0] + E[S_1] + E[S_2]}{s}.$$

4.2. Cost Function

Assume that the structure of the average operating cost is linear cost structure based on the cost elements associated with different system characteristics. First, define the cost elements involved in the average operating cost as follows:

$r_1 \equiv$ the cost of a retrial customer;
$r_2 \equiv$ the cost of an idle server during the non-vacation period;
$r_3 \equiv$ the cost of a busy server during the non-vacation period;
$r_4 \equiv$ the cost of a vacation server;

$r_5 \equiv$ the cost of providing a vacation.

Then, the average operating cost is

$$OC = r_1 E[N] + r_2 E[I] + r_3 E[B] + r_4 E[V] + r_5 \theta.$$

5. Numerical Illustration

If the ergodicity condition is satisfied, this section presents some numerical results on the system characteristics and optimization results. We perform sensitivity analysis to illustrate the effect of system descriptors on the system characteristics.

5.1. Sensitivity Analysis

For the sensitivity analysis, the default parameters are fixed as s = 5, λ = 4, α = 3, μ_1 = 1.5, μ_2 = 1.5, μ_3 = 0.8, σ = 10, θ = 0.3, and b = 0.8, unless their values are mentioned in tables and figures.

Table 1 and Figure 2 clearly show that the average orbit length increases whenever the incoming call rate increases. At the same time, the server idle time is reduced, and the average number of vacation servers decreases.

Table 1. Effect of incoming call rate.

λ	$E[N]$	1-U	$E[B]$	$E[V]$
3	0.224	0.224	3.720	0.308
3.2	0.494	0.214	3.777	0.283
3.4	0.588	0.204	3.832	0.259
3.6	0.700	0.195	3.887	0.236
3.8	0.833	0.186	3.940	0.214
4	0.991	0.177	3.993	0.193
4.2	1.182	0.168	4.045	0.173
4.4	1.415	0.160	4.097	0.153
4.6	1.704	0.152	4.147	0.135
4.8	2.070	0.144	4.197	0.117
5	2.546	0.136	4.246	0.100

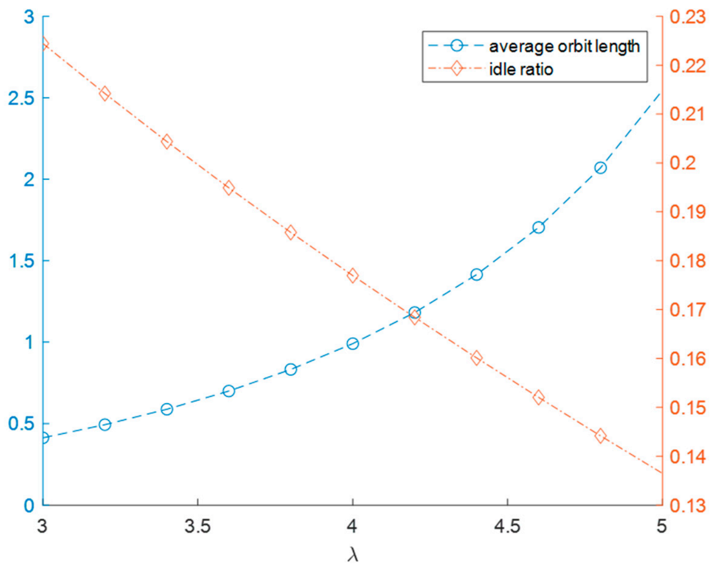

Figure 2. Average orbit length and idle ratio vs. λ.

Table 2 and Figure 3 demonstrate that the increasing nature of the average orbit length first decreases and then increases as the outgoing call rate increases. For the larger outgoing call rate, if the server makes outgoing calls more frequently, it will have fewer opportunities to service incoming calls, resulting in a longer orbit length. On the other hand, the smaller the outgoing call rate, the longer the server takes to make outgoing calls, which results in the fact that the longer the server is idle, the more incoming calls it can serve. This threshold is related to the arrival rate of incoming calls. And the average number of vacation servers decreases. Simultaneously, the server idle time is reduced. Table 3 and Figure 4 display that the server idle time decreases slightly whenever the retrial rate increases. At the same time, the average number of vacation servers increases, and the average orbit length of incoming calls decreases.

Table 2. Effect of outgoing call rate.

α	$E[N]$	**1-U**	$E[B]$	$E[V]$
0	0.659	0.377	1.886	1.947
0.2	0.635	0.361	2.126	1.691
0.4	0.619	0.344	2.359	1.458
0.6	0.613	0.327	2.577	1.249
0.8	0.616	0.310	2.779	1.065
1	0.626	0.293	2.962	0.907
1.2	0.643	0.277	3.127	0.771
1.4	0.667	0.262	3.274	0.656
1.6	0.695	0.248	3.405	0.559
1.8	0.728	0.236	3.521	0.477
2	0.764	0.224	3.624	0.407

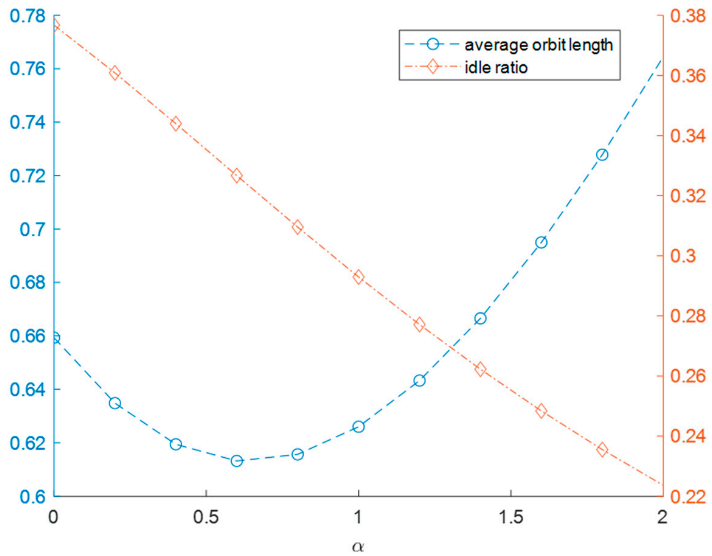

Figure 3. Average orbit length and idle ratio vs. α.

Table 3. Effect of retrial rate.

σ	$E[N]$	**1-U**	$E[B]$	$E[V]$
3	8.602	0.173	4.107	0.049
4	3.314	0.174	4.065	0.102

Table 3. *Cont.*

σ	E[N]	1-U	E[B]	E[V]
5	2.170	0.175	4.041	0.134
6	1.669	0.176	4.025	0.154
7	1.389	0.176	4.013	0.168
8	1.209	0.177	4.005	0.179
9	1.083	0.177	3.998	0.187
10	0.991	0.177	3.993	0.193

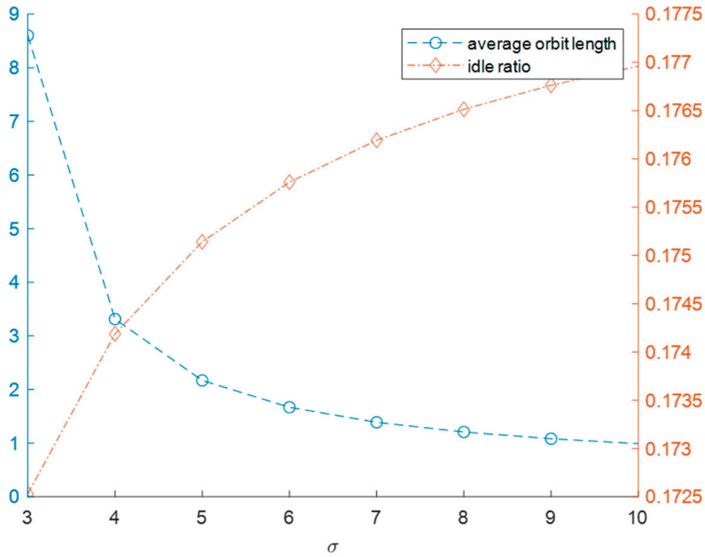

Figure 4. Average orbit length and idle ratio vs. σ.

Table 4 and Figure 5 clearly show that the non-balking rate has an increasing trend for the average orbit length, while the non-balking rate has an increasing tendency for the server idle time and the average number of vacation servers. This is reasonable since when the non-balking rate is increasing, this means that there are more incoming calls joining the orbit.

Table 4. Effect of non-balking rate.

b	E[N]	1-U	E[B]	E[V]
0.0	0.000	0.225	3.697	0.302
0.1	0.045	0.221	3.725	0.292
0.2	0.101	0.216	3.755	0.281
0.3	0.172	0.210	3.787	0.270
0.4	0.262	0.205	3.822	0.257
0.5	0.376	0.198	3.860	0.243
0.6	0.526	0.192	3.900	0.228
0.7	0.723	0.185	3.945	0.211
0.8	0.991	0.177	3.993	0.193
0.9	1.365	0.169	4.046	0.173
1.0	1.908	0.160	4.104	0.151

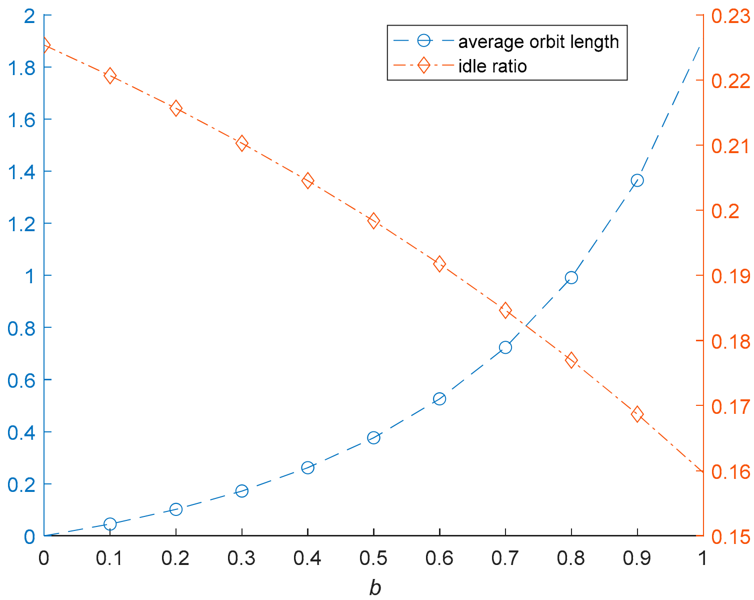

Figure 5. Average orbit length and idle ratio vs. *b*.

5.2. Optimization

The aim of optimization is to determine the optimal outgoing calls rate and optimal vacation rate. Except for operating cost consideration, average orbit length and server utilization are also important system characteristics in the modeling of any queue. Therefore, we will design a tri-objective optimization problem as follows:

$$\begin{aligned} \text{Minimize:} \quad & y_1(\alpha, \theta) = OC \\ \text{Minimize:} \quad & y_2(\alpha, \theta) = 1\text{-}U \\ \text{Minimize:} \quad & y_3(\alpha, \theta) = E[W] \\ \text{Subject to:} \quad & \mathbf{pG}^u\mathbf{e} < \mathbf{pG}^l\mathbf{e}, \\ & \alpha > 0, \\ & \theta > 0. \end{aligned}$$

As can be seen from the expressions of the average operating cost and loss function, the analytical solution is difficult to derive. Computational software is applied to solve the optimization problem numerically. Here, the computer software MATLAB R2022a is used to implement a multi-objective genetic algorithm to solve the above tri-objective optimization problem. Readers can refer to Konak et al. [25] for a detailed overall flow of the general multi-objective genetic algorithm. The multi-objective genetic algorithm pseudocode is shown below (Algorithm 1).

Algorithm 1. Multi-Objective Genetic Algorithm

Begin
  Initialize population
  Evaluate fitness of each individual in the population
While not termination criteria **do**
  Select the best individuals to reproduce
  Apply crossover and mutation operations
  Evaluate population
  Update population
End While
  Output the best solution
End

The optimization analysis has been performed for the default parameters, fixed as follows:

$$s = 5, \lambda = 4, \mu_1 = 1.5, \mu_2 = 1.5, \mu_3 = 0.8, \sigma = 10, b = 0.8, r_1 = 20, r_2 = 25, r_3 = 100, r_4 = 50, r_5 = 5.$$

Figure 6 reveals that the higher the average operating cost, the lower the expected loss is. Table 5 presents the Pareto optimal solution sets. At this time, the operation manager needs to decide which set of solutions to implement. There is no right or wrong choice; it depends on the decision maker's own experience and judgment. To help the decision maker make decisions based on the Pareto optimal solution sets, **Solution #3** and **Solution #4** are selected from Table 5 as reference schemes. **Solution #3** is the solution that achieves the minimum average waiting time in orbit. This is a good solution if the decision maker cares about service quality. **Solution #4** is chosen based on the assigned weight of these three objectives (0.5, 0.2, 0.3). The steps are as follows:

- Step 1: Normalize the three objective values for each solution.
- Step 2: Multiply their respective weights and add these three values.
- Step 3: Sort the values obtained in step 2, and the first solution is the optimal solution.

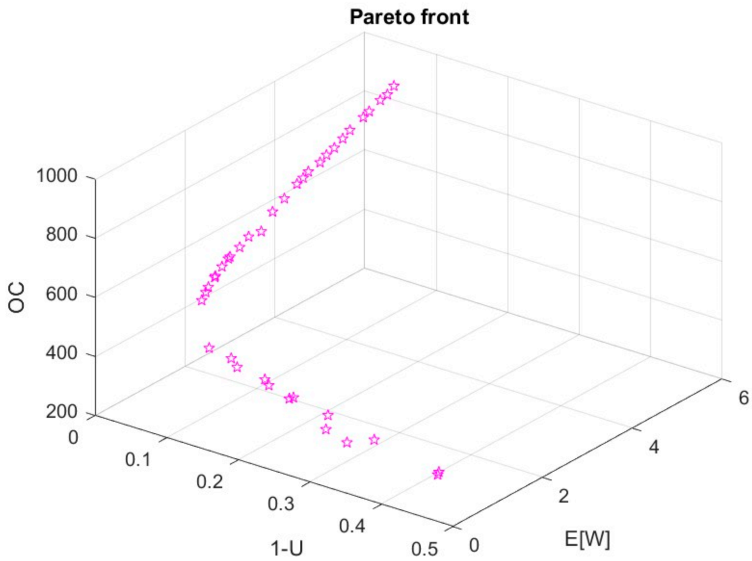

Figure 6. Pareto front solutions found by multi-objective genetic algorithm.

Table 5. Optimal solutions for the multi-objective optimization model.

#	α^*	θ^*	$y_2(\alpha^*, \theta^*)$	$y_3(\alpha^*, \theta^*)$	$y_1(\alpha^*, \theta^*)$	#	α^*	θ^*	$y_2(\alpha^*, \theta^*)$	$y_3(\alpha^*, \theta^*)$	$y_1(\alpha^*, \theta^*)$
1	0.0029	0.1719	0.3303	0.3469	328.0764	21	1.8214	6.1886	0.2302	0.2028	458.0735
2	11.9983	9.3615	0.0607	5.6859	887.9584	22	11.9658	7.9205	0.0608	5.5383	871.2552
3	0.0037	6.9089	0.4770	0.0577	362.5594	23	10.4485	7.6545	0.0682	2.3350	662.1581
4	0.0225	0.1094	0.2909	0.5078	329.4854	24	11.2708	8.7965	0.0639	3.4762	742.4771
5	11.6941	8.1659	0.0620	4.5257	807.2304	25	0.0034	4.6110	0.4753	0.0581	352.2698
6	9.1094	8.1264	0.0764	1.4475	604.6404	26	0.8523	7.8445	0.3177	0.1256	429.1866
7	9.8066	8.6442	0.0718	1.8235	632.9824	27	11.3963	9.2969	0.0634	3.7387	762.0079
8	1.3339	7.3381	0.2670	0.1639	447.7312	28	11.6304	7.6043	0.0623	4.3340	792.0429
9	8.3530	7.7555	0.0819	1.1600	582.2932	29	2.7920	5.0038	0.1808	0.2834	475.5974
10	1.4171	5.0161	0.2608	0.1703	438.9278	30	9.1510	8.3405	0.0761	1.4663	607.0263

Table 5. *Cont.*

#	α^*	θ^*	$y_2(\alpha^*, \theta^*)$	$y_3(\alpha^*, \theta^*)$	$y_1(\alpha^*, \theta^*)$	#	α^*	θ^*	$y_2(\alpha^*, \theta^*)$	$y_3(\alpha^*, \theta^*)$	$y_1(\alpha^*, \theta^*)$
11	3.0315	8.2727	0.1712	0.3045	496.8050	31	4.5492	1.8434	0.1310	0.4534	489.1407
12	11.9288	8.7767	0.0610	5.3783	865.2380	32	10.2003	7.3972	0.0695	2.1116	646.0564
13	11.5663	8.1376	0.0626	4.1553	783.1670	33	11.3412	8.5499	0.0636	3.6194	750.5374
14	10.9316	7.1698	0.0656	2.9075	697.3245	34	11.1095	7.7431	0.0647	3.1834	718.1778
15	0.0225	0.1094	0.2909	0.5078	329.4854	35	11.7452	8.9743	0.0618	4.6904	821.8985
16	0.4143	7.1925	0.3845	0.0905	399.4445	36	11.8315	8.6844	0.0614	4.9937	839.9988
17	1.9076	8.6592	0.2241	0.2100	473.0613	37	11.8663	8.8001	0.0612	5.1255	849.0706
18	10.7258	1.9636	0.0667	2.6367	653.5573	38	11.5096	7.4507	0.0628	4.0075	770.1729
19	7.7684	8.3220	0.0869	0.9913	572.4912	39	9.8691	8.6911	0.0715	1.8647	635.9991
20	0.0029	0.1719	0.3303	0.3469	328.0764	40	8.6137	8.1724	0.0799	1.2486	590.8052

6. Conclusions

This work investigated a synchronous working vacation model of a retrial queue, in which there are two types of arrivals, incoming calls made by customers and outgoing calls made by idle servers. The servers have no information about how many calls exist in orbit. So even with many calls in orbit, outgoing call activity is still possible. We utilized the matrix geometric technique to derive the stationary probabilities, ergodic condition, and system characteristics. At last, we implemented numerical examples to observe the trends of system characteristics for different parametric values and proposed a related optimization issue. The sensitivity analysis in this study can provide useful insights to decision makers to improve service quality. Through numerical examples, it is found that the server idle time is reduced with an increase in the incoming/outgoing call rate and retrial rate. The orbit length is found to decrease with an increase in the incoming/outgoing call rate and retrial rate. The average number of vacation servers decreases with an increase in the incoming call rate and outgoing call rate. The graph behavior was found to be consistent with theoretical expectations. Via a multi-objective genetic algorithm, the entire set of Pareto optimal solutions is determined. We also provide some reference solutions to illustrate how the decision maker can make decisions based on the Pareto optimal solution set.

One limitation of this paper is to assume that all the involved random variables in the model introductions are exponentially distributed. Extending our analysis to consider the general distribution is a future research direction. Additionally, similar models incorporating practical concepts such as working breakdown, batch arrival, immediate feedback, optional type service, negative customers, priority service, and many others can be studied.

Author Contributions: Conceptualization, T.-H.L., K.-C.C., C.-M.C. and F.-M.C.; methodology, T.-H.L. and K.-C.C.; software, T.-H.L. and C.-M.C.; validation, T.-H.L., K.-C.C. and F.-M.C.; writing—original draft preparation, T.-H.L. and C.-M.C.; writing—review and editing, K.-C.C.; supervision, F.-M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by National Science of Technology Council under grants 112-2221-E-324-016-MY2.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kim, J.; Kim, B. A survey of retrial queueing systems. *Ann. Oper. Res.* **2016**, *247*, 3–36. [CrossRef]
2. Phung-Duc, T. Retrial Queueing Models: A Survey on Theory and Applications. *arXiv* **2019**, arXiv:1906.09560.

3. Ke, J.C.; Chang, F.M.; Liu, T.H. M/M/c balking retrial queue with vacation. *Qual. Technol. Quant. Manag.* **2019**, *16*, 54–66. [CrossRef]
4. Zhang, Y. Strategic behavior in the constant retrial queue with a single vacation. *RAIRO Oper. Res.* **2020**, *54*, 569–583. [CrossRef]
5. Liu, T.H.; Chang, F.M.; Ke, J.C.; Sheu, S.H. Optimization of retrial queue with unreliable servers subject to imperfect coverage and reboot delay. *Qual. Technol. Quant. Manag.* **2022**, *19*, 428–453. [CrossRef]
6. Govindan, A.; Jayaraj, U. Analysis of mixed priority retrial queueing system with two way communication and working breakdown. *J. Math. Model.* **2018**, *6*, 195–212.
7. Nazarov, A.; Phung-Duc, T.; Paul, S. Slow retrial asymptotics for a single server queue with two-way communication and Markov modulated Poisson Input. *J. Syst. Sci. Syst. Eng.* **2019**, *28*, 181–193. [CrossRef]
8. Ayyappan, G.; Udayageetha, J.; Somasundaram, B. Analysis of non-pre-emptive priority retrial queueing system with two-way communication, Bernoulli vacation, collisions, working breakdown, immediate feedback and reneging. *Int. J. Math. Oper. Res.* **2020**, *16*, 480–498. [CrossRef]
9. Kumar, M.S.; Dadlani, A.; Kim, K. Performance analysis of an unreliable M/G/1 retrial queue with two-way communication. *Oper. Res.* **2020**, *20*, 2267–2280.
10. Ayyappan, G.; Gowthami, R. Analysis of MAP, PH_2^{OA}/PH_1^{I}, $PH_2^{O}/1$ retrial queue with vacation, feedback, two-way communication and impatient customers. *Soft Comput.* **2021**, *25*, 9811–9838. [CrossRef]
11. Lee, S.W.; Kim, B.; Kim, J. Analysis of the waiting time distribution in M/G/1 retrial queues with two way communication. *Ann. Oper. Res.* **2022**, *310*, 505–518. [CrossRef]
12. Sztrik, J.; Tóth, Á.; Pintér, Á. The Effect of Operation Time of the Server on the Performance of Finite-Source Retrial Queues with Two-Way Communications to the Orbit. *J. Math. Sci.* **2022**, *267*, 196–204. [CrossRef]
13. Tóth, Á.; Sztrik, J. Analysis of retrial queueing systems with two-way communication and impatient customers using simulation. *Ann. Math. Et Inform.* **2023**, *58*, 160–169. [CrossRef]
14. Yang, D.Y.; Wu, C.H. Performance analysis and optimization of a retrial queue with working vacations and starting failures. *Math. Comput. Model. Dyn. Syst.* **2019**, *25*, 463–481. [CrossRef]
15. Li, T.; Zhang, L.; Gao, S. An M/G/1 retrial queue with balking customers and Bernoulli working vacation interruption. *Qual. Technol. Quant. Manag.* **2019**, *16*, 511–530. [CrossRef]
16. Li, T.; Zhang, L.; Gao, S. An M/G/1 retrial queue with single working vacation under Bernoulli schedule. *RAIRO-Oper. Res.* **2020**, *54*, 471–488. [CrossRef]
17. Gupta, P.; Kumar, N. Performance analysis of retrial queueing Model with working vacation, interruption, waiting server, breakdown and repair. *Int. J. Sci. Res.* **2021**, *13*, 833–844. [CrossRef]
18. Muthusamy, S.; Devadoss, N.; Ammar, S.I. Reliability and optimization measures of retrial queue with different classes of customers under a working vacation schedule. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 6806104. [CrossRef]
19. Pazhani, S.; Murugan, B.; Keerthana, R. An M/G/1 feedback retrial queue with working vacation and a waiting server. *J. Comput.* **2023**, *31*, 61–79.
20. Shanmugam, B.; Saravanarajan, M.C. Unreliable retrial queueing system with working vacation. *AIMS Math.* **2023**, *8*, 24196–24224. [CrossRef]
21. Chen, Z.; Xu, H.; Huo, H. Optimal queuing strategies for an M/G/1 retrial queue system with RWV and ISEV policies. *ANZIAM J.* **2024**, 1–27. [CrossRef]
22. Sundararaman, M.; Narasimhan, D.; Rajadurai, P. Performance Analysis of Two Different Types of Waiting Queues with Working Vacations. *J. Math.* **2024**, *2024*, 5829171. [CrossRef]
23. Neut, M.F. Matrix-Geometric Solutions in Stochastic Models. Master's Thesis, John Hopkins University, Baltimore, MD, USA, 1981.
24. Phung-Duc, T.; Rogiest, W. Two Way Communication Retrial Queues with Balanced Call Blending. In Proceedings of the Analytical and Stochastic Modeling Techniques and Applications. ASMTA 2012, Grenoble, France, 4–6 June 2012.
25. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007. [CrossRef]