

Article

Model Validation and Strategy Analysis in Retrial Queues with Delayed Vacations and Feedback Based on Monte Carlo Simulation

Yanling Huang, Ruiling Tian *  and Juntong Su

School of Science, Yanshan University, Qinhuangdao 066004, China

* Correspondence: tianrl@ysu.edu.cn

Abstract: Inspired by call centers, this paper models them as a constant retrial queue, with feedback and delayed vacations to balance high efficiency and low cost for service agents. After completing the service, the server randomly waits for an idle period. If customers arrive during this period, the service is provided immediately, otherwise, the server will take a vacation. We first derive steady-state probabilities and key performance measures. Then, the system cost is modeled. Particle Swarm Optimization (PSO), Ant Colony Algorithm (ACA) and Sparrow Search Algorithm (SSA) are applied to obtain the minimum system cost, respectively. To verify the correctness of the theoretical results of the system model, we simulate the model using Monte Carlo simulation to obtain the probabilities of different server states and the expected number of customers in the system, and then compare them with the theoretical values. At the same time, the sensitivity of the performance measures obtained by Monte Carlo simulation to the system parameters is also analyzed. Finally, customer behavior is analyzed, and equilibrium and socially optimal arrival rates are derived. In addition, the efficiency of the system is evaluated by examining efficiency indicators such as throughput and price of anarchy.

Keywords: delayed vacations; retrial queue; customer behavior; cost optimization; Monte Carlo simulation



Academic Editor: Alexander Zeifman

Received: 13 April 2025

Revised: 9 May 2025

Accepted: 20 May 2025

Published: 2 June 2025

Citation: Huang, Y.; Tian, R.; Su, J. Model Validation and Strategy Analysis in Retrial Queues with Delayed Vacations and Feedback Based on Monte Carlo Simulation. *Mathematics* **2025**, *13*, 1856. <https://doi.org/10.3390/math13111856>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

MSC: 90B22

1. Introduction

Since the retrial phenomenon is common in real life, retrial queueing systems have been widely used in computer systems, call centers, production management, traffic management, and other fields. For the purpose of energy saving and cost control, when there is no customer or a small number of customers in the system, the system will pause the work to take a vacation. Li and Yang [1] explored the M/G/1 retrial model with a finite number of customers and vacations, and the limiting probability distribution and some key performance measures were obtained using the complementary variables approach. In terms of multi-server queues, Ke et al. [2] dealt with an M/M/c balking retrial queue with vacation, and they optimized the cost function to obtain the optimal number of servers as well as the mean service rate. Sztrik et al. [3] investigated a retrial queue for a particular mechanism, namely two-way communication, where the server can fail at any time. Due to the unreliability of equipment, Shanmugam and Saravanarajan [4] proceeded with an M/G/1 unreliable retrial queue with a working vacation and two-phase service.

Recently, providing feedback has become highly demanded for service quality. Similarly to Ke et al. [2], Chang et al. [5] further analyzed the unreliable retrial queue with feedback and impatience using Quasi-Newton, Pattern Search, and Nelder–Mead Simplex Direct Search Methods for cost optimization. Later, a retrial queue with working vacation and impatient customers was studied by including the feature of imperfect service during a working vacation in Jain et al. [6], and customers got additional service if they were not satisfied with the primary work. Sundarapandiyan and Nandhini [7] proposed an M/G/1 retrial queue with feedback and delayed repair, where an application of a computerized processing system was presented.

In queueing systems, it is a natural tendency for customers to decide whether to join the system based on their personal utility. In contrast to the traditional analysis of queueing systems, customers can decide whether or not to join the system based on the information they receive upon arrival, i.e., to maximize their individual utility under the reward–cost structure. These systems can be modeled as a game between customers, and the fundamental problem is to determine the Nash equilibrium. Fundamental results in this area, with an extensive literature review, can be found in the monograph by Hassin and Haviv [8] and Hassin [9].

For the study of queues with vacations from an economic perspective, Economou et al. [10] considered the queue with vacations or failures and quantified the value of renegeing for customers. Zhang and Xu [11] studied equilibrium strategies in observable queues with variable vacation and vacation interruptions, and the model had potential applications in CRNs and cloud computing. However, in real life, the number of customers may be hidden to attract more customers, so it is more realistic to study the unobservable level of information. Zhang et al. [12] further studied the socially optimal strategies in the almost unobservable and completely unobservable cases. As an extension of the vacation policy, a working vacation is not a complete halt to work, but work is done at a lower rate during a vacation. A model with multiple working vacations was proposed by Sun and Li [13], which analyzed customers' behavior from an economic perspective. A large number of recent papers have explored working vacations, and interested readers can refer to [14–16].

Nowadays, a growing number of papers have studied retrial queues from an economic perspective. Economou and Kanta [17] considered equilibrium strategies and maximized social welfare with a constant retrial rate. With a classical retrial queue, Wang and Zhang [18] presented the equilibrium joining probability and equilibrium social welfare under two levels of information. Zhang [19] examined the joining strategy in a constant retrial queue with a single vacation. Sun and Wang [20] considered an equilibrium strategy in a constant retrial queue with Bernoulli vacation, where there is no buffer ahead of the server. In recent years, the economic analysis of delayed vacations was presented by Wang and Zhang [21], which focused on the delayed vacation under classical retrial queue, and the pricing strategy was obtained from the exploration of individual and social benefits. Then, Zhang and Wang [22] made a trade-off between power consumption and delay performance, and they studied the optimal pricing strategy for the M/G/1 retrial queue with setup time and delayed vacations. Gao et al. [23] studied an unreliable retrial queue with delayed vacations to derive the equilibrium strategies for customers in non-cooperative and cooperative cases. However, all of the above papers study classical retrial queues with delayed vacations. Based on Zhang and Wang [22], Wang et al. [24] investigated the constant retrial queue with reserved time and vacation, and the equilibrium strategies of customers and the socially optimal strategy were derived. Then, Shi and Liu [25] analyzed the same problem under the retrial queue, with delayed vacations and two types of customers.

In the call center of an insurance company, the server can not only handle some requests from external calls, such as a basic understanding of car insurance products, consulting the claims process, making insurance appointments, etc., but also make calls to potential customers to advertise or sell new products. The primary incoming call arrives according to the Poisson process. Incoming calls that find the server idle receive their services immediately, or either register in the waiting queue and retry to seek service independently of others after a period of time, or leave forever. There will be a chance that the caller will be dissatisfied with the service after completion, and the caller will continue to be queued for the next service. When the server is idle, free time is generated in which to wait for the next incoming call. If there are no incoming calls during the idle period, the server can make outgoing calls for some time. In this scenario, queueing, retrial-seeking service policies, idle cycles, calls corresponding to orbit, retrial policies, limited idle periods, and customers are the queueing terms.

The main contributions of this paper are as follows. (1) Using the generating function method, the steady-state probability and the important performance measures are obtained. The system cost model is developed to investigate the effect of different system parameters on the decision variables and minimum cost through three optimization algorithms (PSO, ACA, and SSA). (2) Due to the complexity of the model, Monte Carlo simulation is used to simulate and analyze the system, and important system parameters are compared with theoretical values to verify the correctness of the theory. (3) Based on the performance measures and reward–cost structure, the optimal arrival rates of customers are analyzed and efficiency indicators such as system throughput and the price of anarchy are calculated to evaluate the efficiency of the system. The decision-maker can detect these two indicators to optimize system performance while reducing losses to find the optimal balance between system performance and individual interests.

The rest of the paper is organized as follows. In Section 2, we focus on a detailed description of the model. Section 3 analyzes the steady-state probabilities and some important performance measures through the generating functions. Section 4 optimizes the cost function based on three optimization algorithms. Monte Carlo simulations verify the accuracy of the theoretical part of the model in Section 5. Individual equilibrium and socially optimal strategies are studied in Section 6. Then, in Section 7, we focus on understanding the effects of the price of anarchy (PoA) and throughput. Finally, we explore the equilibrium arrival rate and the maximum social welfare under different system parameters through numerical analysis in Section 8.

2. Model Description

Consider a retrial queue with delayed vacations and feedback, where arriving customers can get neither the state of the server nor the number of customers in the system. Potential customers arrive individually, according to a Poisson process, with rate Λ . Upon arrival, the customers join the system with a probability q , and balk with the probability $1 - q$. Then, the effective arrival rate is $\lambda = \Lambda q$. When a customer enters the system, if the server is idle, the customer immediately accepts the service, and the service time follows an exponential distribution with rate μ . Otherwise, the customer enters a retrial space with infinite capacity to retry, and the retrial process follows a Poisson process with parameter θ .

After completing the service, the server randomly keeps a period of idle time, which is exponentially distributed with rate α . If a customer arrives during this idle period, the server immediately provides service. Otherwise, the server will have a vacation, which is exponentially distributed with rate β . Upon completion of a vacation period, the server re-enters an idle period, waiting for potential arrivals. After the service, customers will

leave the system with a probability b , and will return to the orbit with probability $1 - b$ due to dissatisfaction with the service. It is assumed that inter-arrival times, service times, retrial times, and vacation times are independent of each other.

Suppose that each customer gains a reward of R units after completing the service while having to pay a waiting cost of C per unit time to remain in the system. All customers are indistinguishable and rational to pursue their own profit maximization, and they have the right to decide whether or not to enter the system at the time of their arrival. Explicitly, customers tend to enter if they gain more from the service than they pay for the cost, otherwise, they prefer to balk. On the other hand, if the reward is equal to the cost, customers do not care whether they enter or not.

Let $(I(t), N(t))$ represent the system state at time t , where $I(t)$ is the status of the server and $N(t)$ is the number of customers in the orbit, and

$$I(t) = \begin{cases} 0, & \text{the server is idle at time } t, \\ 1, & \text{the server is busy at time } t, \\ 2, & \text{the server is on vacation at time } t. \end{cases}$$

We obtain the state space as $\Omega = \{(i, n) : i = 0, 1, 2, n \geq 0\}$, and the state transition diagram is shown in Figure 1. The stationary queue length distribution is denoted by $\{\pi_{i,n} : (i, n) \in \Omega\}$.

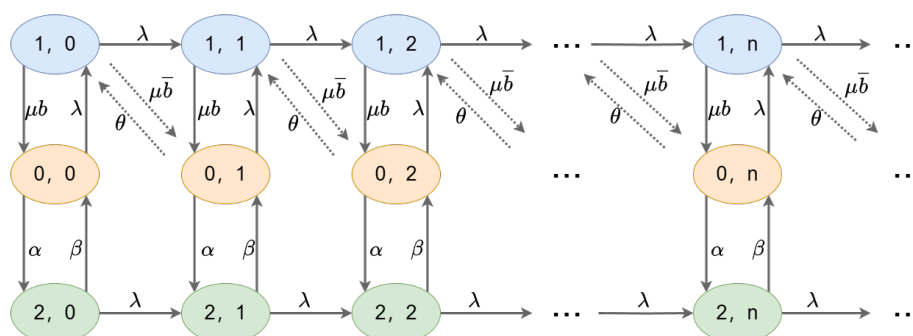


Figure 1. State transition diagram.

Based on the method of matrix analysis, the stability condition of the system is given by $\frac{\lambda\beta(\lambda + \theta) + \lambda\mu(\alpha + \beta\bar{b})}{\theta\mu b\beta} < 1$.

The proof can be found in Appendix A.

3. Steady-State Analysis of the System

Here, we consider the completely unobservable case. Assume that the system is stable; then, we can easily get the balance equations:

$$(\lambda + \mu b + \mu\bar{b})\pi_{1,0} = \lambda\pi_{0,0} + \theta\pi_{0,1}, \quad (1)$$

$$(\lambda + \mu b + \mu\bar{b})\pi_{1,n} = \lambda\pi_{0,n} + \theta\pi_{0,n+1} + \lambda\pi_{1,n-1}, \quad n \geq 1, \quad (2)$$

$$(\lambda + \alpha)\pi_{0,0} = \mu b\pi_{1,0} + \beta\pi_{2,0}, \quad (3)$$

$$(\lambda + \alpha + \theta)\pi_{0,n} = \mu b\pi_{1,n} + \beta\pi_{2,n} + \mu\bar{b}\pi_{1,n-1}, \quad n \geq 1, \quad (4)$$

$$(\lambda + \beta)\pi_{2,0} = \alpha\pi_{0,0}, \quad (5)$$

$$(\lambda + \beta)\pi_{2,n} = \alpha\pi_{0,n} + \lambda\pi_{2,n-1}, \quad n \geq 1. \quad (6)$$

Define the probability-generating functions of each state i :

$$Q_i(z) = \sum_{n=0}^{\infty} \pi_{i,n} z^n, \quad i = 0, 1, 2.$$

Theorem 1. Consider the retrial queue with delayed vacations and feedback.

(i) The probabilities that the server is idle, busy, on vacation, and $\pi_{0,0}$ are as follows:

$$Q_0(1) = \frac{\beta(\mu b - \lambda)}{\mu b(\alpha + \beta)},$$

$$Q_1(1) = \frac{\lambda}{\mu b},$$

$$Q_2(1) = \frac{\alpha(\mu b - \lambda)}{\mu b(\alpha + \beta)},$$

$$\pi_{0,0} = \frac{F}{\mu \theta b(\alpha + \beta)},$$

where $F = -\beta\lambda^2 - \lambda(\mu\bar{b}\beta + \alpha\mu + \theta\beta) + \theta\mu b\beta$.

(ii) The expected sojourn time in the system of a tagged customer who decides to enter the system is given by

$$W(\lambda) = \frac{(\beta(\alpha + \beta) - \alpha\mu b)\lambda^2 + M\lambda + \mu^2 b((\alpha + \beta)(\alpha + \beta\bar{b}) + \alpha\theta b)}{\mu b(\alpha + \beta)F},$$

where $M = (\alpha + \beta)(\beta\mu + \beta\theta + \alpha\mu\bar{b}) + \alpha\mu b(\mu b - \theta)$.

Proof. From the balance equations above, we multiply Equations (2), (4), and (6) with z^n and sum over n , then adding (1), (3), and (5), respectively, gives

$$\begin{cases} (\lambda + \mu - \lambda z)Q_1(z) = \left(\lambda + \frac{\theta}{z}\right)Q_0(z) - \frac{\theta}{z}\pi_{0,0}, \\ (\lambda + \alpha + \theta)Q_0(z) = (\mu b + \mu\bar{b}z)Q_1(z) + \beta Q_2(z) + \theta\pi_{0,0}, \\ (\lambda + \beta - \lambda z)Q_2(z) = \alpha Q_0(z). \end{cases}$$

Then, we simplify the above equations and obtain the generating functions as follows:

$$\begin{cases} Q_0(z) = \frac{t\theta - \frac{\theta}{z}y}{\left(\lambda + \alpha + \theta - \frac{\alpha\beta}{m}\right)t - \lambda y - \frac{\theta y}{z}}\pi_{0,0}, \\ Q_1(z) = \frac{(\theta - \frac{\theta}{zt}y)(\lambda + \frac{\theta}{z})}{\left(\lambda + \alpha + \theta - \frac{\alpha\beta}{m}\right)t - \lambda y - \frac{\theta y}{z}}\pi_{0,0} - \frac{\theta}{zt}\pi_{0,0}, \\ Q_2(z) = \frac{\alpha}{m} \frac{t\theta - \frac{\theta}{z}y}{\left(\lambda + \alpha + \theta - \frac{\alpha\beta}{m}\right)t - \lambda y - \frac{\theta y}{z}}\pi_{0,0}, \end{cases} \quad (7)$$

where $m = \lambda + \beta - \lambda z$, $t = \lambda + \mu - \lambda z$ and $y = \mu b + \mu\bar{b}z$. Bringing $z = 1$ into (7), we have

$$Q_0(1) = \frac{\lambda\theta - \mu\theta b}{\lambda(\lambda + \theta) + \frac{\lambda\alpha\mu}{\beta} - \theta\mu b + \lambda\mu\bar{b}}\pi_{0,0}, \quad (8)$$

$$Q_1(1) = \frac{\lambda + \theta}{\mu} \frac{\lambda\theta - \mu\theta b}{\lambda(\lambda + \theta) + \frac{\lambda\alpha\mu}{\beta} - \theta\mu b + \lambda\mu\bar{b}}\pi_{0,0} - \frac{\theta}{\mu}\pi_{0,0}, \quad (9)$$

$$Q_2(1) = \frac{\alpha}{\beta} \frac{\lambda\theta - \mu\theta b}{\lambda q(\lambda + \theta) + \frac{\lambda q \alpha \mu}{\beta} - \theta \mu b + \lambda \mu \bar{b}} \pi_{0,0}. \quad (10)$$

According to the normalization condition $Q_0(1) + Q_1(1) + Q_2(1) = 1$, we obtain the probabilities in each state and $\pi_{0,0}$.

Next, we will calculate customers' expected sojourn time. The generating functions in each state are obtained in (7). By taking the derivative of z , we derive $Q'_i(z)$ ($i = 0, 1, 2$). Then, we put $z = 1$ into $Q'_i(z)$, as follows:

$$\begin{cases} Q'_0(1) = \frac{-\theta \mu b F + \theta(\lambda q - \mu b) \left(\frac{\alpha}{\beta} (\lambda q)^2 (1 - \frac{\mu}{\beta}) - \theta \mu b \right)}{\mu \theta b F (\alpha + \beta)}, \\ Q'_1(1) = \frac{\lambda + \theta}{\mu} Q'_0(1) + \frac{\lambda q(\lambda + \theta) - \theta \mu}{\mu^2} Q_0(1) - \frac{\lambda q \theta - \theta \mu}{\mu^2} \pi_{0,0}, \\ Q'_2(1) = \frac{\lambda q \alpha}{\beta^2} Q_0(1) + \frac{\alpha}{\beta} Q'_0(1). \end{cases} \quad (11)$$

By simplifying (11), the expected number of customers in the system is

$$\begin{aligned} N &= Q_1(1) + Q'_0(1) + Q'_1(1) + Q'_2(1) \\ &= \frac{(\beta(\alpha + \beta) - \alpha \mu b) \lambda^3 + M \lambda^2 + \lambda \mu^2 b ((\alpha + \beta)(\alpha + \beta \bar{b}) + \alpha \theta b)}{\mu b (\alpha + \beta) F}. \end{aligned}$$

According to Little's formula, $W(\lambda) = \frac{N}{\lambda}$. $\square$

4. Cost Analysis

A cost function plays an essential role in the queueing model as it represents the total cost incurred by the system. The prime goal of the queueing analysis is to increase revenue, but providing customers with high-quality service and less waiting time will increase the cost of the system. Therefore, we need to optimize the cost function. The different cost notations involved in the cost function are illustrated as follows:

- C_i : cost per unit time the server is idle,
 - C_b : cost per unit time the server is busy,
 - C_v : cost per unit time the server is on vacation,
 - C_o : holding cost per unit time of a customer staying in the orbit,
 - C_s : cost per unit time of providing a service rate μ .
- The cost function is defined as follows:

$$Cost(\mu, \beta) = C_i Q_0(1) + C_b Q_1(1) + C_o N + C_v Q_2(1) + \mu C_s. \quad (12)$$

The cost function given by (12) is nonlinear with respect to (μ, β) . Therefore, using classical optimization techniques to evaluate the exact analysis results of (μ^*, β^*) is a difficult task. We use PSO, ACA, and SSA to obtain the optimal decision variables (μ^*, β^*) . Since the results obtained are randomized, we average the results over 20 runs. Meanwhile, for the settings in PSO, population size = 500; maximum number of iterations = 40; and the learning factors are 0.4 and 0.3, respectively. SSA: sparrow count = 40; maximum iteration count = 40; Safety threshold = 0.6. For ACA, ant count = 500; maximum iteration count = 40; pheromone evaporation coefficient = 0.9; and transfer probability constant = 0.2.

With both μ and β in the interval range of $[4, 10]$ and $\Lambda = 1.5$, $\lambda = 1$, $b = 0.9$, $\alpha = 1$, $\theta = 5$, we perform the cost optimization with three different cost notations. Cost set I: $C_i = 10$; $C_b = 20$; $C_v = 3$; $C_o = 5$; $C_s = 10$. Cost set II: $C_i = 20$; $C_b = 30$; $C_v = 4$; $C_o = 6$; $C_s = 12$. Cost set III: $C_i = 30$; $C_b = 40$; $C_v = 5$; $C_o = 9$; $C_s = 15$.

In the following, we present three optimization algorithms and some important parameters, respectively, and the effect of different system parameters on the decision variables and the minimum system cost is investigated.

4.1. Implementation of PSO

PSO is a type of evolutionary algorithm and was proposed by Kennedy and Eberhart [26]. The algorithm was originally inspired by the regularity of the activity of clusters of flying birds and seeks to find an optimal solution through collaboration and information sharing between individuals in the group. This algorithm has attracted academic attention for its easy implementation, high accuracy, and fast convergence, and has demonstrated its superiority in solving practical problems.

We optimize μ and β to obtain the corresponding minimum value of the cost function. Figure 2a shows the logic diagram of the PSO algorithm. The cost convergence graphs for the three cases are shown in Figure 3, based on Cost set I, which demonstrates the change in the optimization results of the cost function as the number of iterations increases when $\alpha = 0.5, 1, 1.5$, respectively. The fitness value usually decreases rapidly during the initial iterations as the particles explore the solution space, gradually approaching the optimal region. As the number of iterations increases, the rate of convergence may slow down and the curve flattens out, indicating that the algorithm may have found a local or global optimum.

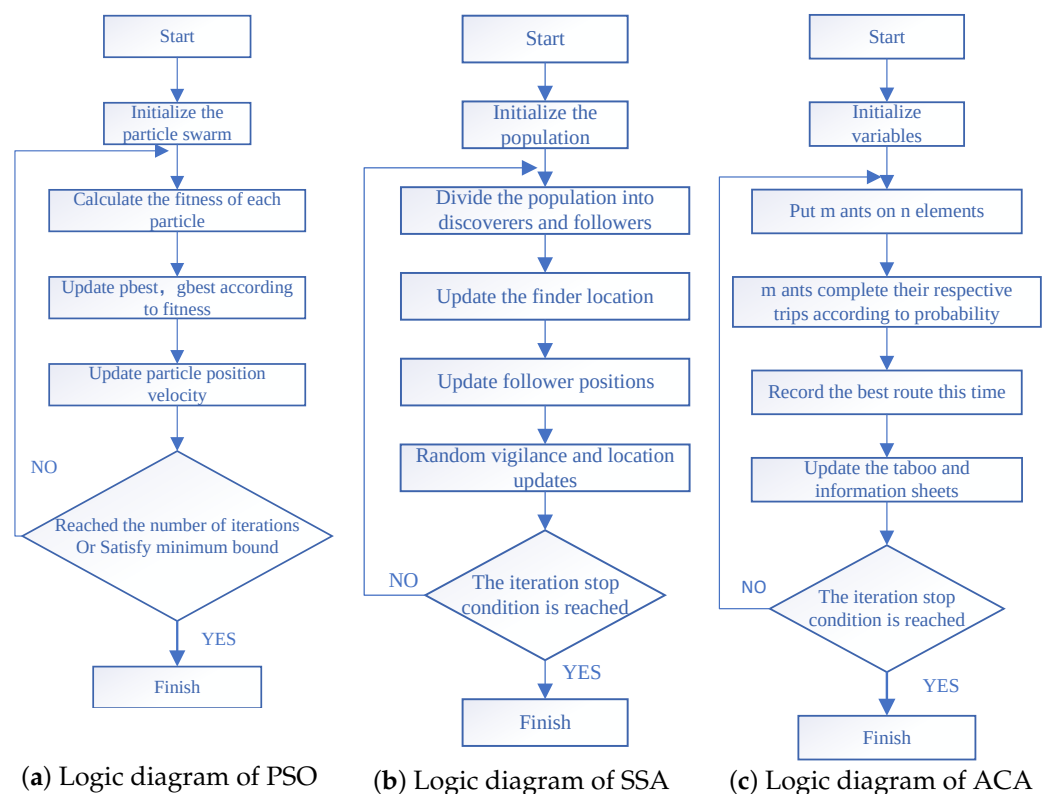


Figure 2. Logic diagram of the optimization algorithm.

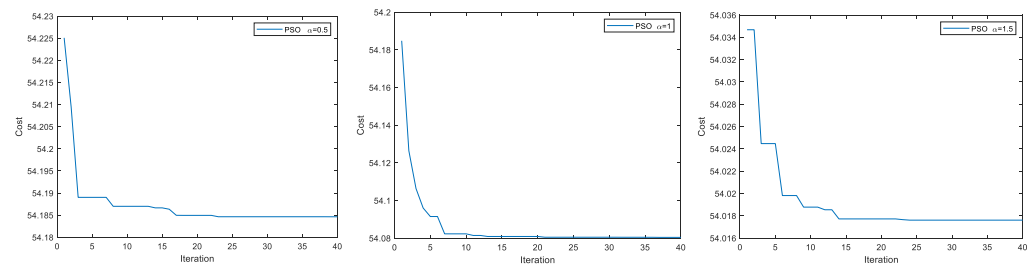


Figure 3. Convergence graph of PSO on cost.

4.2. Implementation of SSA

The SSA is an emerging meta-heuristic algorithm, which is a swarm intelligence algorithm optimized based on the social characteristics of a group. The algorithm simulates the foraging and anti-predator behavior of sparrows by constantly updating individual positions. Compared with the traditional algorithms, the SSA has a simple structure, few control parameters, and strong local search capability.

The SSA mainly achieves positional optimization by mimicking sparrows' foraging and anti-predation behaviors to find the local optimum. In the preset of this algorithm, the sparrow population is internally categorized into two roles: discoverer and follower, while mimicking real predation scenarios and adding a danger warning mechanism for sparrows. The safety threshold is empirically determined to avoid premature convergence. Based on the logic diagram in Figure 2b, we perform an optimization analysis of the decision variables μ , β , and $Cost(\mu^*, \beta^*)$. The number of iterations is set to 40, in line with the PSO. The convergence plots of the cost function and the number of iterations in the three scenarios are shown in Figure 4.

In Figure 4, we find that the initial period of convergence decreases rapidly, which is because the discoverers will quickly disperse to different regions of the search space to actively explore the potential optimal solutions, and the followers will directly gather to the better positions among the discoverers to quickly reduce the search scope, forming a 'centralized' local development and avoiding ineffective random searches.

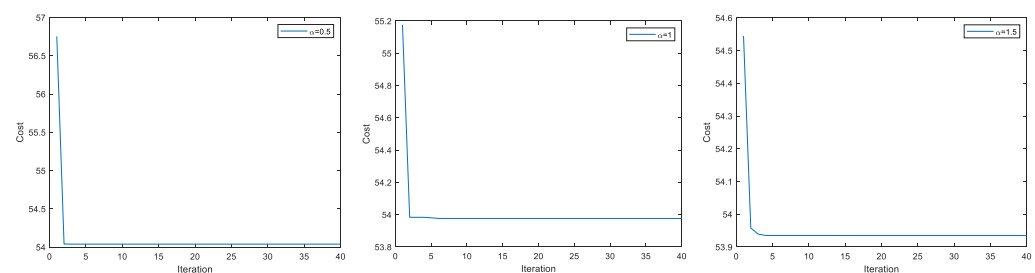


Figure 4. Convergence graph of SSA on cost.

4.3. Implementation of ACA

The ACA is a heuristic algorithm that simulates the foraging behavior of ants in nature, and it is widely used in the solution of optimization problems. The basic idea of ACA is to place a group of ants in the solution space of the problem and let them gradually find the optimal solution by transferring and volatilizing pheromones.

Figure 2c shows the exact steps of the ACA algorithm. The cost convergence graphs for the three cases are shown in Figure 5. At the beginning of the iteration, pheromone distribution is uniform, ants randomly choose paths, and convergence is slow. As the pheromone concentration of the high-quality path increases significantly, the ants are guided to concentrate their search, accelerating the convergence and subsequently finding

the locally optimal solution. So, the cost convergence graphs show a trend of very slow iterations followed by a rapid decline until a locally or globally optimal solution is sought.

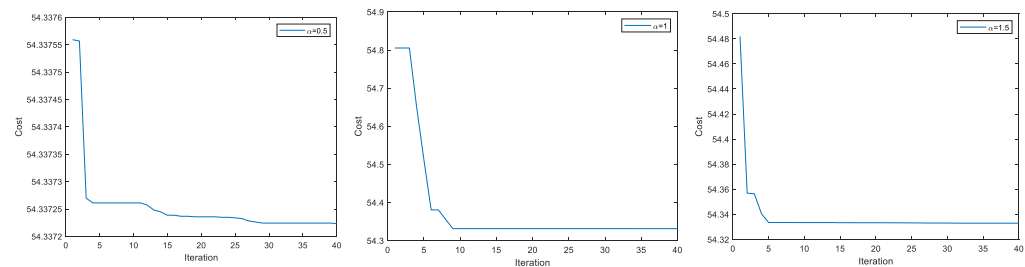


Figure 5. Convergence graph of ACA on cost.

4.4. The Effect of Parameters on the Decision Variables and Cost

As α and θ varies, decision variables (μ^*, β^*) and $Cost(\mu^*, \beta^*)$ are shown in Tables 1 and 2. The data show that there is little difference between the results of the three optimization algorithms, but the results for SSA are slightly lower and the results for 20 optimizations are more stable. The convergence plot of cost versus number of iterations shows that PSO is slightly slower to optimize than the other two algorithms, while SSA has the fastest iterations. The SSA algorithm has a short run time, while the PSO algorithm has a relatively long run time. The SSA algorithm is relatively slightly better than the other two algorithms.

Table 1. Effect of α on $(\mu^*, \beta^*, Cost(\mu^*, \beta^*))$.

α	Algorithm	Cost Set I	Cost Set II	Cost Set III
0.5	PSO	(4, 6.71, 54.24)	(4, 4.38, 63.32)	(4, 4.52, 94.19)
	SSA	(4, 6.346, 54.04)	(4, 4, 63.27)	(4, 4, 94.12)
	ACA	(4, 6.48, 54.12)	(4, 5.20, 63.42)	(4, 5.48, 94.33)
1	PSO	(4, 7.86, 53.98)	(4, 4.28, 62.72)	(4, 4.44, 93.46)
	SSA	(4, 7.88, 53.98)	(4, 4, 62.67)	(4, 4, 93.40)
	ACA	(4, 7.75, 54.02)	(4, 5.29, 62.88)	(4, 5.34, 93.67)
1.5	PSO	(4, 8.68, 53.94)	(4, 4.27, 62.3)	(4, 4.58, 93.02)
	SSA	(4, 9.42, 53.94)	(4, 4, 62.29)	(4, 4.20, 93.00)
	ACA	(4, 8.66, 53.94)	(4, 5.14, 62.45)	(4, 5.87, 93.20)

Table 2. Effect of θ on $(\mu^*, \beta^*, Cost(\mu^*, \beta^*))$.

θ	Algorithm	Cost Set I	Cost Set II	Cost Set III
3	PSO	(4, 9.36, 54.94)	(4, 5.33, 64.12)	(4, 6.99, 95.76)
	SSA	(4, 10, 54.93)	(4, 5.28, 64.11)	(4, 6.90, 95.76)
	ACA	(4, 8.76, 54.97)	(4, 6.14, 64.14)	(4, 7.27, 95.77)
4	PSO	(4, 9.08, 54.32)	(4, 4.46, 63.22)	(4, 4.67, 94.32)
	SSA	(4, 10, 54.31)	(4, 4, 63.18)	(4, 4.34, 94.31)
	ACA	(4, 8.30, 54.34)	(4, 5.09, 63.29)	(4, 5.16, 94.39)
5	PSO	(4, 7.86, 53.98)	(4, 4.28, 62.72)	(4, 4.44, 93.46)
	SSA	(4, 7.897, 53.98)	(4, 4, 62.67)	(4, 4, 93.40)
	ACA	(4, 7.75, 54.02)	(4, 5.29, 62.88)	(4, 5.34, 93.67)

From these tables, it is evident that as α decreases, β^* decreases and $Cost(\mu^*, \beta^*)$ increases. Additionally, with an increase in θ , both $Cost(\mu^*, \beta^*)$ and β^* decrease. However, μ^* remains relatively stable throughout these changes. As α decreases, $Cost(\mu^*, \beta^*)$ increases, while β^* decreases. As α increases, the reserved idle time decreases. The expected number of customers in the system and the probability of the server being busy decreases, so the optimal cost value decreases. As θ varies, the minimum value of the cost function decreases

with the decrease of the μ , so the optimal service rate remains unchanged under the change in θ and μ . Moreover, $Cost(\mu^*, \beta^*)$ shows a significant increase as cost notations rise.

Figure 6 demonstrates the effect of different system parameters on the cost function. These graphs also confirm our analysis of the optimization results. The red dots indicate the points corresponding to the minimum cost values. In Figure 6a, the cost function shows an increasing trend as α decreases, and $Cost(\mu^*, \beta^*)$ also increases. Because as α increases, the reserved idle time decreases. The average number of customers in the orbit and the probability that the server is busy decrease, so the optimal cost value decreases, which is consistent with the changes in the data in Table 1, suggesting that shorter idle periods reduce both orbit congestion and operational costs. From Figure 6b, it can be seen that as θ increases, the cost function shows a decreasing trend. Because as θ increases, the retrial interval for customers in the orbit decreases, consequently leading to a downward trend in the cost function. As the other parameters vary, the point of the minimum cost value is always shifted on the curve with $\mu = 4$ in Figure 6c,d. The minimum value of the cost function decreases with the decrease of the μ . As the service rate increases, the number of customers in the orbit and the probability that the service desk is busy decrease, but the partial increase in the cost of providing service dominates.

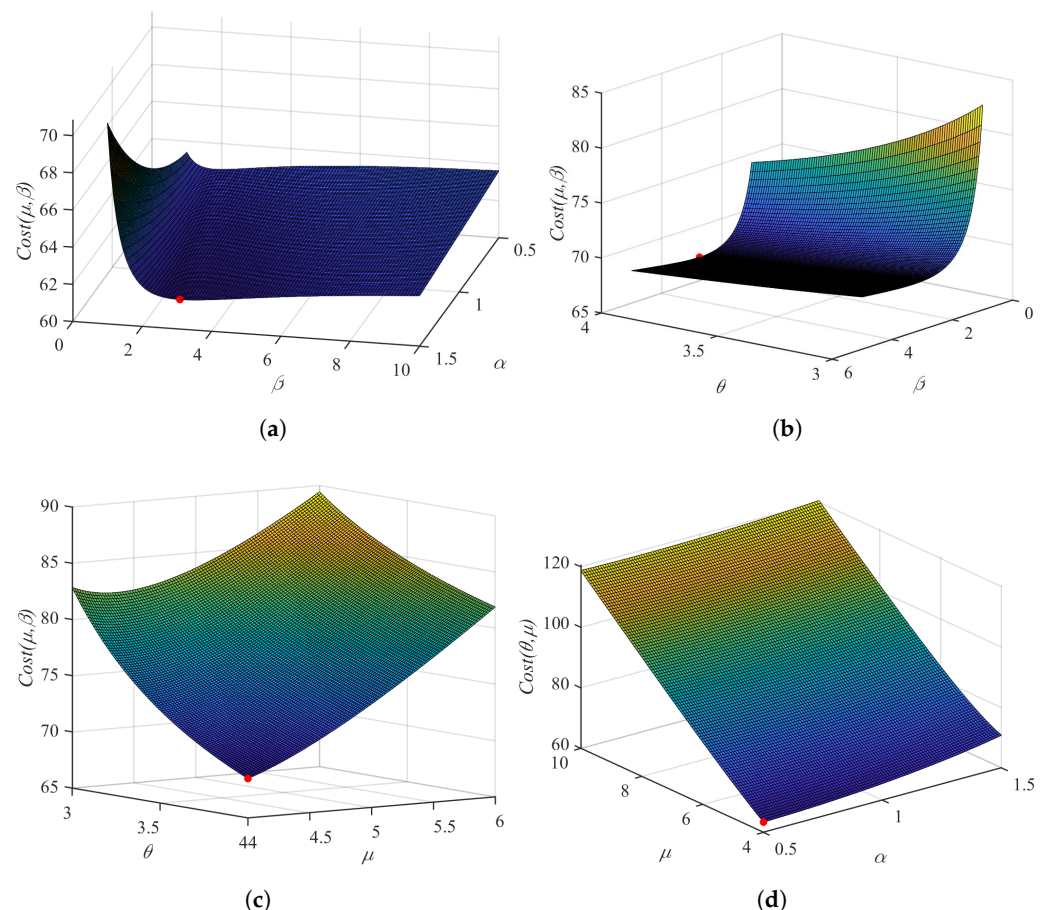


Figure 6. Cost vs. system parameters.

These findings of cost optimization are highly beneficial for the systems as they facilitate efficient resource allocation, reduction of wastage, and improved performance. By carefully managing costs, managers can enhance their competitiveness, profitability, and overall effectiveness. This not only supports sustainable operations but also enables better resource utilization, ultimately contributing to long-term success and growth in a competitive landscape.

5. Monte Carlo Simulation

Monte Carlo simulation is a mathematical technique based on random numbers that is used to simulate complex systems and computational problems, especially those that involve a large number of variables and a large amount of uncertainty. Details can be found in [27]. In the field of queueing theory, customers' arrival times and service times are uncertain. The majority of queueing systems operate in a stochastic state, and the complexity of the study of queueing systems lies in the randomness of the system, which fits with the Monte Carlo simulation. Monte Carlo simulations have begun to appear in the papers of many researchers in the areas of queueing theory and so on (see Liu [28] and Fitzgerald et al. [29]). Now, let us run a Monte Carlo simulation of this model. The steps of the simulation are described as follows:

Step 1: Input: number of simulations (n), the cost notations, system parameters, simulation time (t).

Step 2: Define the simulation variables: current server status (S), the time of the next transition (T), number of customers in system (N), service variable (VS), the arrival time of customers in orbit ($time_{orbit}$), time of commencement of vacation ($time_{break}$), current time, the arrival time of the source customer ($time_{main}$), the service completion time ($time_{finish}$), vacation end time ($time_{breakend}$).

Step 3: Initialization variables: set $S = 0$, $T = 0$, $N = 0$, $VS = 0$, $time_{orbit} = inf$, and $time_{break} = inf$.

Step 4: Simulation: by simulating and analyzing the model, the specific algorithm is shown in Algorithm 1.

Algorithm 1 Monte Carlo Simulation

```

1: for  $i$  in 1:  $n$  do
2:   Service the first customer. Update  $time_{finish}$  and  $time_{main}$ . We set  $T = \min(time_{finish}, time_{main})$  and  $t = T$ ;
3:   while current time  $< t$  do
4:     if  $T = time_{main}$  then
5:       Update  $S = 1$  and  $VS = 1$ ;
6:     end if
7:     if  $T = time_{orbit}$  then
8:       Set  $N = N - 1$  and update  $S = 1$ ,  $VS = 1$ ;
9:     end if
10:    if  $VS = 1$  then
11:      Update  $time_{finish}$  and  $time_{main}$ . Set  $T = \min(time_{finish}, time_{main})$ , current time =  $T +$  current time and  $N = N + 1$ ;
12:      Keep performing the above steps until the service ends;
13:    end if
14:    Set  $VS = 0$ , if the customer is unsatisfied with the service, then  $N = N + 1$ ;
15:    if  $T = time_{break}$  then
16:      Set  $S = 2$ , update  $time_{breakend}$  and  $time_{main}$ , set  $N = N + 1$  and  $T = \min(time_{breakend}, time_{main})$  until vacation ends;
17:    end if
18:    initialize state  $S = 0$ , update  $time_{orbit}$ ,  $time_{main}$ ,  $time_{break}$  according to exponential distribution, and  $T = \min(time_{orbit}, time_{main}, time_{break})$ ;
19:  end while
20: end for

```

Step 5: Count the number of customers in each state and calculate 95% confidence intervals for n simulations.

First, some important performance measures are obtained, which illustrate the accuracy of the model. The details are shown in Tables 3 and 4. The system parameters are

$\Lambda = 1.5; \lambda = 1; \mu = 5; \theta = 4; \beta = 1$; and $b = 0.9$. The model is further validated by comparison with theoretical values. All data are within 95% confidence intervals.

Next, we analyze the costs through Monte Carlo simulation. In Figure 7, the theoretical values of the expected number of customers in the system follow the same trend as the simulated experimental values. We can also observe that the difference between the exact number of customers and the simulation results is small and follows the same trend. Therefore, in cases where the system is complex or no expression is shown, we can consider using the simulation to model cost minimization.

Table 3. Simulation results and theoretical values with μ .

System Parameter	$\mu = 3$		$\mu = 4$		$\mu = 5$		$\mu = 6$	
	Exact	Sim	Exact	Sim	Exact	Sim	Exact	Sim
P_0	0.315	0.314	0.361	0.357	0.389	0.386	0.407	0.395
P_1	0.370	0.363	0.278	0.274	0.222	0.226	0.185	0.183
P_2	0.315	0.323	0.361	0.369	0.389	0.388	0.407	0.422
N	4.250	4.271	2.702	2.722	2.211	2.250	1.975	1.968

Table 4. Simulation results and theoretical values with β .

System Parameter	$\beta = 1$		$\beta = 2$		$\beta = 3$		$\beta = 4$	
	Exact	Sim	Exact	Sim	Exact	Sim	Exact	Sim
P_0	0.361	0.380	0.482	0.469	0.542	0.529	0.578	0.586
P_1	0.278	0.260	0.278	0.280	0.278	0.264	0.278	0.258
P_2	0.361	0.361	0.241	0.251	0.181	0.177	0.144	0.141
N	2.702	2.700	1.018	1.029	0.698	0.686	0.572	0.620

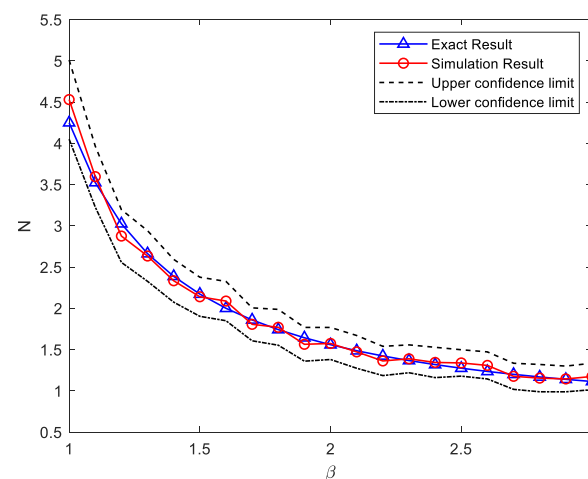


Figure 7. The expected number of customers in the system with β for $\Lambda = 1.5; \lambda = 1; \alpha = 1; \theta = 4; \mu = 3$; and $b = 0.9$.

6. Customer Strategy Analysis

In this section, we derive customers' equilibrium behaviors and the socially optimal arrival rates of customers, respectively.

In the following analysis, we assume that the system is stable, i.e., $F > 0$; then, λ ranges in $[0, \lambda_{max})$, where λ_{max} is the only non-negative solution for $F = 0$. We discuss customers' equilibrium strategies. The equilibrium arrival rate of customers is analyzed based on the reward–cost function and game theory. Next, in order to obtain the optimal

arrival rate, we need to derive the monotonicity of the expected sojourn time. The following lemma provides the result of the monotonicity of the expected sojourn time.

Lemma 1. *The monotonicity of $W(\lambda)$ is discussed in the classification:*

- (i) *When $A \geq 0$, $W(\lambda)$ is monotonically increasing with $\lambda \in [0, \lambda_{max})$;*
- (ii) *When $A < 0$ and $\bar{\lambda} \geq \lambda_{max}$, $W(\lambda)$ is monotonically growing with $\lambda \in [0, \lambda_{max})$, where $\bar{\lambda}$ is the only positive solution which satisfies $W'(\lambda) = 0$;*
- (iii) *When $A < 0$ and $\bar{\lambda} \in [0, \lambda_{max})$, $W(\lambda)$ will initially increase and then decrease, where $A = \beta^2 - \alpha\beta + \alpha\mu$.*

Proof. The derivative of $W(\lambda)$ with respect to λ is given by

$$\frac{dW(\lambda)}{d\lambda} = \frac{A\lambda^2 + B\lambda + D}{F^2},$$

where

$$A = \beta^2 - \alpha\beta + \alpha\mu,$$

$$B = 2\beta(\beta\theta + \alpha\mu + \beta\mu\bar{b}),$$

$$D = \beta\mu\theta(1 + \bar{b})(\alpha + \beta) + \alpha\mu^2(2\beta\bar{b} + \alpha + \theta b) + \beta^2(\mu^2\bar{b}^2 + \theta^2).$$

We know that $B, D > 0$, so $W(0) \geq 0$.

In case (i), when $A \geq 0$, $W(\lambda)$ is increasing, since $W'(\lambda) \geq 0$ with $\lambda \in [0, \lambda_{max})$.

In case (ii), when $A < 0$ and $\bar{\lambda} \notin [0, \lambda_{max})$, $W'(\lambda) > 0$ with $\lambda \in [0, \lambda_{max})$, where $\bar{\lambda}$ is the only solution satisfying $W'(\lambda) = 0$. So, $W(\lambda)$ grows monotonically with $\lambda \in [0, \lambda_{max})$.

In case (iii), when $A < 0$ and $\bar{\lambda} \in [0, \lambda_{max})$, $W'(\lambda)$ is first greater than 0 in $[0, \bar{\lambda}]$ and then less than 0 in $[\bar{\lambda}, \lambda_{max})$, where $\bar{\lambda}$ is the only solution satisfying $W'(\lambda) = 0$. So, $W(\lambda)$ first increases and then decreases. $\square$

Under the reward–cost structure, the expected net benefit of an arriving tagged customer who finds the server unavailable and decides to enter the system is

$$U(\lambda) = R - CW(\lambda).$$

Next, we derive customers' equilibrium behaviors based on the reward–cost function in the following Theorem.

Theorem 2. *Consider the M/M/1 retrial queue with delayed vacations and feedback.*

Case I. *When $A \geq 0$ or When $A < 0$ and $\bar{\lambda} \geq \lambda_{max}$,*

$$\lambda_e = \begin{cases} 0, & \frac{R}{C} \leq \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}, \\ \min\{\lambda_1, \Lambda\}, & \frac{R}{C} > \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}, \end{cases}$$

where λ_1 satisfies $R - CW(\lambda) = 0$ and $\lambda_1 \in [0, \lambda_{max})$.

Case II. *When $A > 0$, $\bar{\lambda} \in [0, \lambda_{max})$.*

$$1. \quad \text{If } \frac{R}{C} \leq \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)},$$

$$\lambda_e = \begin{cases} 0, & \lambda_2 > \Lambda, \\ \{0, \lambda_2, \Lambda\}, & \lambda_2 \leq \Lambda, \end{cases}$$

where λ_2 is the nonnegative solution of $R - CW(\lambda) = 0$.

2. If $\frac{R}{C} > \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}$, there are two cases.
If $U(\bar{\lambda}) > 0$, $\lambda_e = \Lambda$. If $U(\bar{\lambda}) < 0$, then

$$\lambda_e = \begin{cases} \Lambda, & \lambda_3 > \Lambda, \\ \lambda_3, & \lambda_3 \leq \Lambda < \lambda_4, \\ \{\lambda_3, \lambda_4, \Lambda\}, & \lambda_4 \leq \Lambda, \end{cases}$$

where λ_3 and λ_4 ($\lambda_4 \geq \lambda_3$) are the solutions of $R - CW(\lambda) = 0$.

Proof. In case I, according to the monotonicity of $W(\lambda)$, when $A > 0$ or $A < 0$ and $\bar{\lambda} \geq \lambda_{max}$, $W(\lambda)$ increases with $\lambda \in [0, \lambda_{max}]$; thus, $U(\lambda)$ decreases in λ . $U(\lambda)$ reaches its maximum at $\lambda = 0$, so we have

$$\lim_{\lambda \rightarrow 0} W(\lambda) = \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}.$$

When $\frac{R}{C} \leq \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}$, $U(\lambda) \leq 0$ for $\lambda \in [0, \lambda_{max}]$. The customer's best response is to balk, so $\lambda_e = 0$.

When $\frac{R}{C} > \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}$, there is a unique root $\lambda_1 \in [0, \lambda_{max}]$ of $R - CW(\lambda) = 0$. If $\lambda_1 > \Lambda$, $U(\lambda) \geq 0$ for every $\lambda \in [0, \lambda_{max}]$. If $\lambda_1 \leq \Lambda$, λ_1 will be the mix strategy. Therefore, $\lambda_e = \min\{\lambda_1, \Lambda\}$.

In case II, when $A > 0$ and $\bar{\lambda} \in [0, \lambda_{max}]$, $U(\lambda)$ first decreases and then grows based on the monotonicity of $W(\lambda)$.

When $\frac{R}{C} \leq \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}$, $\lambda_e = 0$ is always the equilibrium arrival rate, since nobody can obtain any profits by entering the system when others choose to balk, and $\lambda_2 \in [0, \lambda_{max}]$ is the root of $U(\lambda) = 0$. If $\lambda_2 > \Lambda$, $U(\lambda) \leq 0$, so the best response is balk and $\lambda_e = 0$. If $\lambda_2 \leq \Lambda$, $\lambda_e = \{0, \lambda_2, \Lambda\}$, because customers are always willing to join if $\frac{R}{C} \geq W(\lambda)$.

When $\frac{R}{C} > \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}$, if $U(\bar{\lambda}) > 0$, the minimum value of the individual utility function is greater than 0 and the best response is joining, so $\lambda_e = \Lambda$. If $U(\bar{\lambda}) < 0$, we know $\frac{R}{C} > \frac{\alpha + \beta\bar{b}}{\beta\theta b} + \frac{\alpha}{\beta(\alpha + \beta)}$; thus, the individual utility function has two points of intersection with the x-axis. There are two roots (λ_3 and λ_4) of $U(\lambda) = 0$, then the analysis is as above. $\square$

Now, we investigate social welfare, which is the sum of all the customers' expected net benefits. The goal of the social planner is to maximize social welfare, and the expression of the social welfare is given by

$$S(\lambda) = \lambda(R - CW(\lambda)).$$

Due to the complexity of social welfare, we solve the socially optimal arrival rate by the PSO in Section 8.

7. Efficiency Measures: PoA and Throughput

In this section, we focus on understanding the price of anarchy (PoA) and throughput. The goal is to find an optimal balance between system performance and individual benefit.

This information holds significance for aiding social planners in making well-informed decisions.

Throughput denotes the number of customers that can be served within a given time. Higher throughput means better system performance and more efficient service to customers within the system. We get the throughput as follows:

$$TH(\lambda) = \lambda \left(\sum_{n=0}^{\infty} \pi_{0,n} + \sum_{n=0}^{\infty} \pi_{1,n} + \sum_{n=0}^{\infty} \pi_{2,n} \right)$$

A common measure of this availability is the price of anarchy (PoA), which is the ratio of socially optimal welfare to equilibrium social welfare. The PoA is usually calculated as the ratio between a worst-case scenario, reflecting non-cooperative behavior, and a best-case scenario, representing the outcome of full cooperation. For more details, see [30]. The PoA is defined as follows:

$$PoA = \frac{S(\lambda^e)}{S(\lambda^*)}$$

A price of anarchy (PoA) of 1 indicates that there is no loss of efficiency due to selfish or uncoordinated behavior, which means that the system is operating at optimal efficiency. In contrast, a PoA less than 1 indicates that selfish behavior or lack of coordination between individuals leads to suboptimal performance, resulting in a loss of efficiency.

8. Numerical Results

In this section, we first analyze the relationship between the equilibrium and the socially optimal arrival rates under different parameters and analyze the influence of system parameters on the maximum social welfare. Then, we analyze the variation of throughput and PoA under different parameters.

8.1. Equilibrium Strategy and Social Welfare Analysis

Figures 8–10 show the trend of the equilibrium and socially optimal arrival rates as the parameters μ , α , and β vary. First, in Figure 8, the equilibrium arrival rate declines as α rises. This is because customers are less likely to accept service with less reserved time and more waiting time, so customers are more inclined to balk. Thus, the arrival rate is reduced in both cooperative and uncooperative states.

Figures 9 and 10 similarly demonstrate that both equilibrium and socially optimal arrival rates increase with higher values of μ and β . The μ represents the service rate and β represents the vacation rate. It is easy to understand that by reducing the service time and the vacation time, the waiting time for customers will decrease. Therefore, they are more willing to join the system. At this point, there will be a phenomenon of following the crowd, thinking that there will be excessive customers in the system. For an FCFS queueing system, impose more waiting time will be imposed on any other customer who arrives after this initial one once a customer decides to join it. This is called a negative externality, and because of it, the upward arrival rate of the equilibrium arrival rate will gradually slow down.

From the above analysis, we find that the equilibrium arrival rate is not smaller than the socially optimal arrival rate. This can be explained from an economic point of view. Selfish customers always ignore the negative externalities they bring to the system. As a result, public resources are over-utilized and the system is more crowded than the social planner expects. The classical approach for this is to charge customers for joining the system to make the equilibrium arrival rate consistent with the socially optimal arrival.

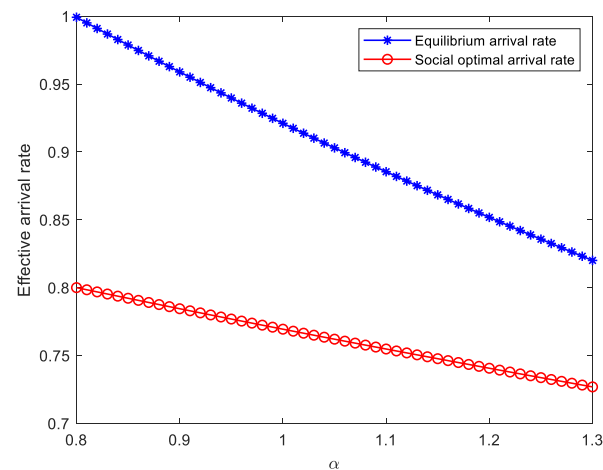


Figure 8. The arrival rate with α for $\Lambda = 1.5$; $R = 6$; $C = 1$; $\beta = 1$; $\theta = 4$; $\mu = 3$; and $b = 0.8$.

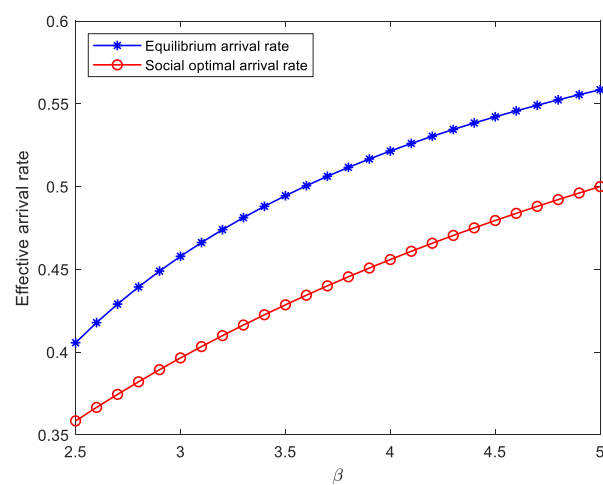


Figure 9. The arrival rate with β for $\Lambda = 1.5$; $R = 1$; $C = 1$; $\alpha = 1$; $\theta = 3$; $\mu = 3$; and $b = 0.7$.

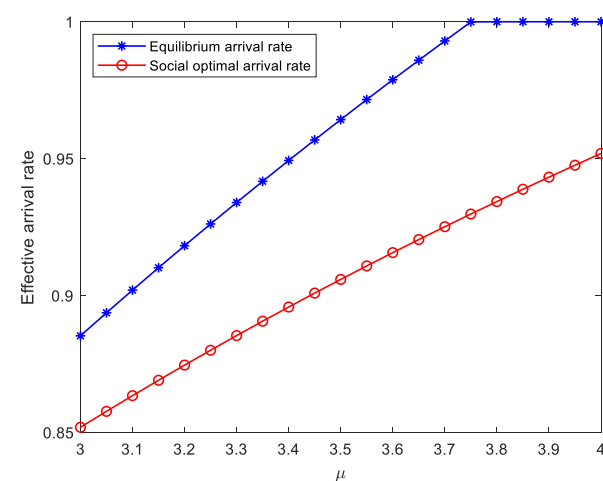


Figure 10. The arrival rate with μ for $\Lambda = 1.5$; $R = 5$; $C = 1$; $\alpha = 3$; $\theta = 3$; $\beta = 4$; and $b = 0.8$.

Then, we show the effect of the system parameters on the maximum social welfare in Figures 11–13. As μ and θ increase, customers' sojourn time is reduced and customers become more willing to join the system in Figure 11, which will contribute to increasing the maximum social welfare. It is clear that as R increases, the socially optimal welfare increases, which is in line with our intuition. Similarly, as C increases, it leads to higher costs per unit of time and is not favorable for socially optimal welfare. In Figure 13, as α

increases, the reserved idle time and sojourn time increase, so social welfare decreases. As θ increases, the customer's sojourn time decreases, which is favorable to increasing social welfare.

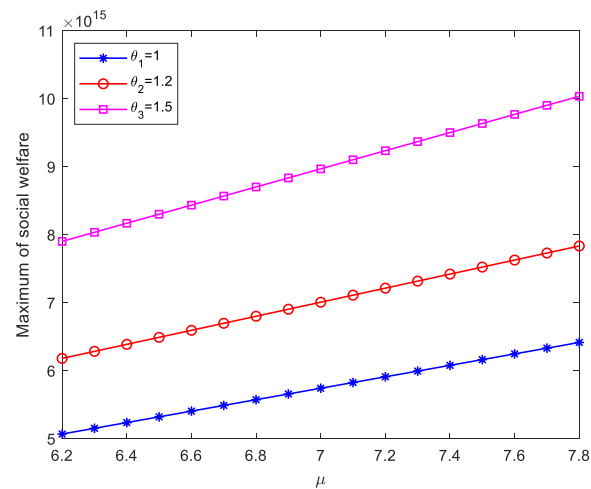


Figure 11. The maximum of social welfare with μ for $\Lambda = 1.5$; $\lambda = 1$; $R = 8$; $C = 1$; $\alpha = 3.5$; $\beta = 1.1$; and $b = 0.6$.

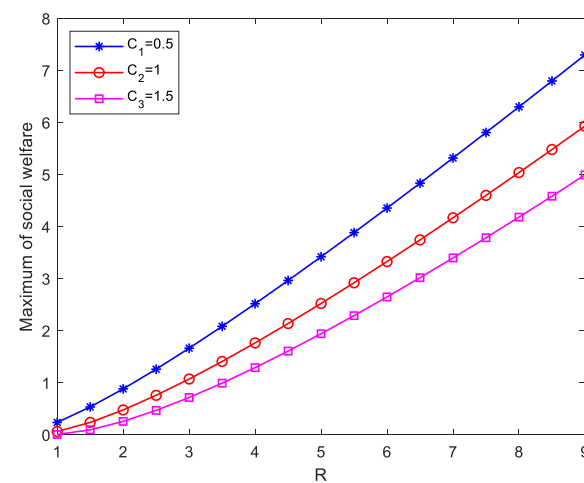


Figure 12. The maximum of social welfare with R for $\Lambda = 1.5$; $\lambda = 1$; $\alpha = 2$; $\theta = 4$; $\mu = 3$; $\beta = 2$; and $b = 0.9$.

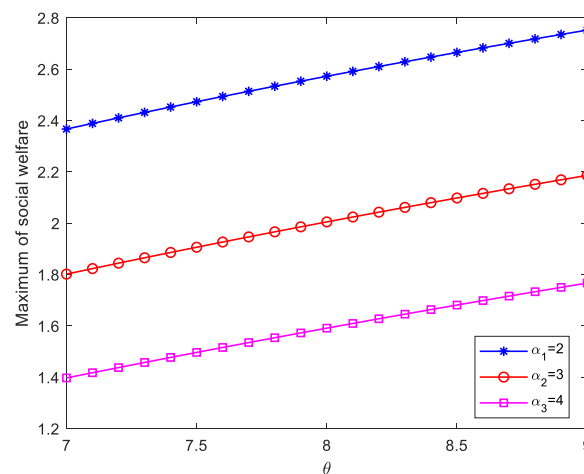


Figure 13. The maximum of social welfare with θ for $\Lambda = 1.5$; $\lambda = 1$; $R = 5$; $C = 2$; $\mu = 4.5$; $\beta = 0.3$; and $b = 0.9$.

8.2. Throughput and PoA

To evaluate the performance of the throughput and PoA with respect to socially optimal and individual equilibrium strategies, we analyze the throughput and PoA vs. the system parameters. The results are presented in Table 5.

Firstly, it can be demonstrated that $TH(\lambda^e)$ is greater than or equal to $TH(\lambda^*)$ under identical parameters. As α increases, $TH(\lambda^e)$ and $TH(\lambda^*)$ decrease. It can be observed that as the value of β increases, $TH(\lambda^e)$ and $TH(\lambda^*)$ increase. An increase in both μ and θ results in a reduction in $TH(\lambda^e)$, while $TH(\lambda^*)$ initially rises and then decreases after reaching a certain level.

Table 5. Comparison of the throughput and the PoA with individual and social equilibrium vs. some system parameters.

System Parameter	Parameter Values	$TH(\lambda^e)$	$TH(\lambda^*)$	PoA
μ	4	0.8194	0.6105	0.9093
	5	0.8056	0.6690	0.9928
	6	0.6944	0.6944	1
	7	0.6845	0.6845	1
α	1	0.8194	0.8194	1
	2	0.7292	0.6680	0.9797
	3	0.6750	0.4700	0.3065
	4	0.6708	0.4549	0.1291
θ	3	0.7593	0.5562	0.4734
	4	0.7292	0.6680	0.9797
	5	0.7111	0.7111	1
	6	0.6991	0.6991	1
β	1	0.6790	0.5841	0
	2	0.7593	0.5562	0.4734
	3	0.8074	0.7600	0.9893
	4	0.8395	0.8395	1

It can be shown that when PoA is equal to 1, $TH(\lambda^e)$ and $TH(\lambda^*)$ are identical. In the case of minimal α values, the PoA stabilizes at a value of 1 and then decreases as the value of α increases. As illustrated by Figures 14 and 15, PoA rises in conjunction with the value of μ and θ , reaching a maximum value of 1.

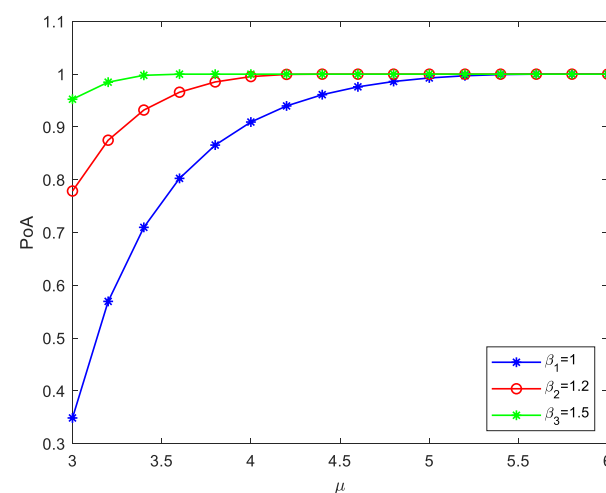


Figure 14. PoA with μ for $\Lambda = 1.5$; $\lambda = 1$; $R = 5$; $C = 1$; $\theta = 4$; $\alpha = 2$; and $b = 0.9$.

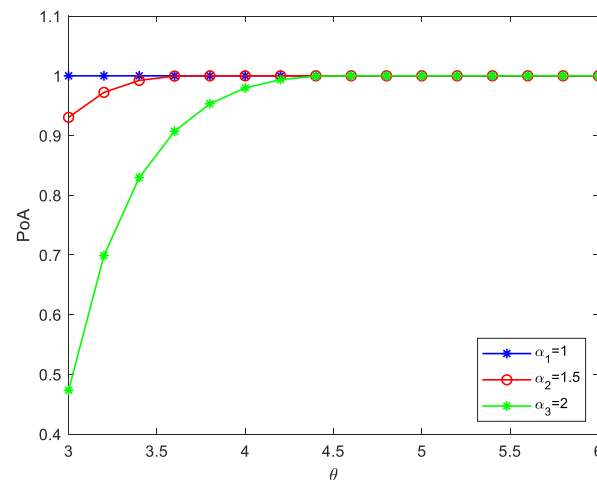


Figure 15. PoA with θ for $\Lambda = 1.5$; $\lambda = 1$; $R = 5$; $C = 1$; $\mu = 4$; $\beta = 1$; and $b = 0.9$.

9. Conclusions

This paper focuses on the socially optimal and equilibrium arrival rates of the M/M/1 constant retrial queue with delayed vacations and feedback. Firstly, we obtain the key performance measures of the system according to the probability-generating functions. Subsequently, the cost function is optimized to provide suggestions for decision-makers through three optimization algorithms. Due to the complexity of the calculation, we use the Monte Carlo method to simulate customers' behavior. Some important performance measures are obtained and we then compare them with the theoretical values, which further verify the accuracy of the model. The expected net benefit is derived from the reward–cost function to further analyze the equilibrium and the socially optimal arrival rates. Then, we focus on understanding the effects of the price of anarchy (PoA) and throughput. Finally, we analyze the customers' strategy, the throughput, and PoA with the system parameters. Future research can investigate systems with general service rates. The model can be further extended by combining the N -policy and the customer's hesitation behavior.

Author Contributions: Conceptualization, Y.H. and R.T.; writing—original draft preparation, Y.H., R.T. and J.S.; writing—review and editing, Y.H., R.T. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China under Grant No. 71971189 and the Local Science and Technology Development Projects of the Central Committee: S&T Program of Hebei (grant number 246Z0305G).

Data Availability Statement: Data are contained within the article.

Acknowledgments: Many thanks to the editors and reviewers for their work and consideration in the publication of our paper. We would like to express our sincere gratitude to the editors and reviewers.

Conflicts of Interest: No potential conflicts of interest are reported by the authors.

Appendix A

To analyze the steady-state conditions, we employ the Quasi-Birth–Death (QBD) process. From the state transition diagrams, the transition rate matrix Q is structured as

$$Q = \begin{pmatrix} A_0 & C & & \\ B & A & C & \\ & B & A & C \\ & & \ddots & \ddots & \ddots \end{pmatrix}, \quad (A1)$$

where

$$A_0 = \begin{pmatrix} -(\lambda + \alpha) & \lambda & \alpha \\ \mu b & -(\lambda + \mu) & 0 \\ \beta & 0 & -(\lambda + \beta) \end{pmatrix}, \quad (A2)$$

$$A = \begin{pmatrix} -(\lambda + \alpha + \theta) & \lambda & \alpha \\ \mu b & -(\lambda + \mu) & 0 \\ \beta & 0 & -(\lambda + \beta) \end{pmatrix}, \quad (A3)$$

$$B = \begin{pmatrix} 0 & \theta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 0 & 0 \\ \mu \bar{b} & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix}, \quad (A4)$$

It is found that the process $\{(I(t), N(t)), t \geq 0\}$ is a QBD process. By using the theory of QBD, we can obtain the stability condition. Defining the matrix $H = A + B + C$, we have

$$H = \begin{pmatrix} -(\lambda + \theta + \alpha) & \lambda + \theta & \alpha \\ \mu & -\mu & 0 \\ \beta & 0 & -\beta \end{pmatrix}, \quad (A5)$$

The steady-state distribution vector $\pi = (\pi_1, \pi_2, \pi_3)$ satisfies the equations

$$\pi H = 0, \quad \pi e = 1.$$

By solving the above equations, the steady state distribution is given by

$$\begin{cases} \pi_1 = \frac{\beta \mu}{\beta \mu + (\alpha + \theta) \beta + \alpha \mu}, \\ \pi_2 = \frac{\beta(\alpha + \theta)}{\beta \mu + (\alpha + \theta) \beta + \alpha \mu}, \\ \pi_3 = \frac{\alpha \mu}{\beta \mu + (\alpha + \theta) \beta + \alpha \mu} \end{cases}$$

The system is ergodic if $\pi B e > \pi C e$ is true. After simplification, the steady state condition of the system is $\frac{\lambda \beta (\lambda + \theta) + \lambda \mu (\alpha + \beta \bar{b})}{\theta \mu b \beta} < 1$.

References

- Li, H.; Yang, T. A single-server retrial queue with server vacations and a finite number of input sources. *Eur. J. Oper. Res.* **1995**, *85*, 149–160. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Sztrik, J.; Tóth, Á.; Pintér, Á.; Bács, Z. 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*. [\[CrossRef\]](#)
- Shanmugam, B.; Saravananarajan, M.C. Unreliable retrial queueing system with working vacation. *AIMS Math.* **2023**, *8*, 24196–24224. [\[CrossRef\]](#)
- Chang, F.M.; Liu, T.H.; Ke, J.C. On an unreliable-server retrial queue with customer feedback and impatience. *Appl. Math. Model.* **2018**, *55*, 171–182. [\[CrossRef\]](#)
- Jain, M.; Dhibar, S.; Sanga, S.S. Markovian working vacation queue with imperfect service, balking and retrial. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–17. [\[CrossRef\]](#)
- Sundarapandian, S.; Nandhini, S. Non-Markovian Feedback Retrial Queue with Two Types of Customers and Delayed Repair Under Bernoulli Working Vacation. *Contemp. Math.* **2024**, 2093–2122.
- Hassin, R.; Haviv, M. *To Queue or Not to Queue: Equilibrium Behavior in Queueing systems*; Springer Science & Business Media: Philadelphia, PA, USA, 2003.
- Hassin, R. *Rational Queueing*; CRC Press: Boca Raton, FL, USA, 2016.

10. Economou, A.; Logothetis, D.; Manou, A. The value of reneging for strategic customers in queueing systems with server vacations/failures. *Eur. J. Oper. Res.* **2022**, *299*, 960–976. [\[CrossRef\]](#)
11. Zhang, Y.; Xu, X. Equilibrium analysis of cloud user request based on the Markov queue with variable vacation and vacation interruption. *RAIRO-Oper. Res.* **2021**, *55*, 2807–2825. [\[CrossRef\]](#)
12. Zhang, Y.; Xu, X.; Liu, M. Optimal strategy analysis of a Markovian queue with variable vacation and vacation interruptions under unobservable cases. *Qual. Technol. Quant. Manag.* **2024**, *21*, 556–586. [\[CrossRef\]](#)
13. Sun, W.; Li, S. Equilibrium and optimal behavior of customers in Markovian queues with multiple working vacations. *Top* **2014**, *22*, 694–715. [\[CrossRef\]](#)
14. Azhagappan, A.; Deepa, T. Variant impatient behavior of a Markovian queue with balking reserved idle time and working vacation. *RAIRO-Oper. Res.* **2020**, *54*, 783–793. [\[CrossRef\]](#)
15. Wang, Y.; Wang, J.; Zhang, G. The effect of information on the strategic behavior in a Markovian queue with catastrophes and working vacations. *Qual. Technol. Quant. Manag.* **2024**, *21*, 688–721. [\[CrossRef\]](#)
16. Wang, Y.; Tian, R. Strategic analysis in Markovian queues with a single working vacation and multiple vacations. *Int. J. Innov. Comput. Inf. Control* **2020**, *16*, 349–366.
17. Economou, A.; Kanta, S. Equilibrium customer strategies and social-profit maximization in the single-server constant retrial queue. *Nav. Res. Logist. (NRL)* **2011**, *58*, 107–122. [\[CrossRef\]](#)
18. Wang, J.; Zhang, F. Strategic joining in M/M/1 retrial queues. *Eur. J. Oper. Res.* **2013**, *230*, 76–87. [\[CrossRef\]](#)
19. Zhang, Y. Strategic behavior in the constant retrial queue with a single vacation. *RAIRO-Oper. Res.* **2020**, *54*, 569–583. [\[CrossRef\]](#)
20. Sun, K.; Wang, J. Equilibrium joining strategies in the single-server constant retrial queues with Bernoulli vacations. *RAIRO-Oper. Res.* **2021**, *55*, S481–S502. [\[CrossRef\]](#)
21. Wang, J.; Zhang, F. Monopoly pricing in a retrial queue with delayed vacations for local area network applications. *IMA J. Manag. Math.* **2016**, *27*, 315–334. [\[CrossRef\]](#)
22. Zhang, Y.; Wang, J. Equilibrium pricing in an M/G/1 retrial queue with reserved idle time and setup time. *Appl. Math. Model.* **2017**, *49*, 514–530. [\[CrossRef\]](#)
23. Gao, S.; Dong, H.; Wang, X. Equilibrium and pricing analysis for an unreliable retrial queue with limited idle period and single vacation. *Oper. Res.* **2021**, *21*, 621–643.
24. Wang, L.L.; Liu, L.W.; Chai, X.D.; Wang, Z. On equilibrium in a constant retrial queueing system with reserved time and vacations. *J. Oper. Res. Soc. China* **2022**, 1–16. [\[CrossRef\]](#)
25. Shi, X.; Liu, L. Equilibrium joining strategies in the retrial queue with two classes of customers and delayed vacations. *Methodol. Comput. Appl. Probab.* **2023**, *25*, 52. [\[CrossRef\]](#)
26. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948.
27. Mooney, C.Z. *Monte Carlo Simulation*; Number 116; Sage: Thousand Oaks, CA, USA, 1997.
28. Liu, Y. M/M/1 Queue Model with Uncertain Parameters by Monte Carlo Simulation. *Appl. Mech. Mater.* **2014**, *556*, 3541–3544. [\[CrossRef\]](#)
29. Fitzgerald, K.; Pelletier, L.; Reznec, M. A Queue-Based Monte Carlo Analysis to Support Decision Making for Implementation of an Emergency Department Fast Track. *J. Healthc. Eng.* **2017**, *2017*, 6536523. [\[CrossRef\]](#)
30. Gilboa-Freedman, G.; Hassin, R.; Kerner, Y. The price of anarchy in the Markovian single server queue. *IEEE Trans. Autom. Control* **2013**, *59*, 455–459. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.