Chapter 4

Operating Strategies in Markovian Environment

4.1 Introduction

During the last few decades, the queue-based service systems with realistic behaviors like server's vacation or working vacation and reneging have been broadly investigated due to their applications in various fields, such as computer systems, communication networks, production systems, etc. In real-life, the impatience behavior is the most prominent feature for an individual customer who wants to experience service but needs to queue. For the characterization of customer's impatient behavior, the most common terminology is reneging, which is defined as the customer who joins the queue but leaves without being served after waiting for sometimes. The concept of reneging was introduced by [96]. Recently, [51] surveyed in-depth for the effects of time limitations on decision-making via queueing theoretic approach and had given broad insight on the reneging behavior of the customers. The main assumption for the reneging in the literature is that the customers execute independent abandonment, *i.e.* each one of them sets a period of own impatience and abandons the service system as soon as that time is over. In the present chapter, we consider that customers are impatient, but they perform synchronized abandonment. In the synchronized abandonment, the abandonment opportunities arise according to a certain point process with which all present customers decide simultaneously but independently of the others whether they will abandon the service system or not.

In this chapter, we consider that at the abandonment epochs, every present customer remains in the service system with probability q and abandons the system with the complementary probability $p = (1 - q)$, independent of the others. Economou and Probab [69] analyzed an exclusive model in which the population evolved according to a Poisson fashion and exhibited synchronized reneging following binomial distribution and transitions occurred according to a renewal process. Economou and

Kapodistria [70] and [71] used q-hypergeometric series for computing the performance measures of single server queue with synchronized abandonments behavior for unreliable server and gated service respectively. Furthermore, [123] carried out the extensive analysis, including the stationary distribution, the busy period, and the conditional sojourn time distribution on deriving exact formulas for a single server and multi-server queueing model with synchronized abandonments. Recently, [228] studied for point process reneging behavior in finite population queueing model.

Servi and Finn [222] proposed a special kind of semi-vacation policy in which the server provides the service at a slower rate instead of being completely terminate the service. Such type of vacation policy is known as working vacation (WV). Liu et al.[185] demonstrated the stochastic decomposition structures of the waiting time and queue length in a single server Markovian queueing model with a working vacation. Zhang and Zhou [317] explored different types of working vacation and established stochastic decomposition results of stationary indices by QBD process and generalized eigenvalues method. Recently, [232] surveyed many research articles having impatience behavior and vacation policy for the finite population queueing model in detail.

In the working vacation policy, we generally assume that the server returns to a normal working level only when the system is non-empty at the end of a vacation. Such an assumption seems much more confining in real-world situations. To overcome that kind of limitation, [170] introduced the vacation interruption policy in a single server queueing model with working vacations. In that controllable vacation policy, if there are waiting customers more than pre-specified threshold at the epoch of a service completion in the vacation period, the server immediately ends own vacation and resumes the normal working level. Otherwise, server continues the working vacation until the system is non-empty after a service or a vacation ends.

In the last decade, some investigations had been done for Markovian and non-Markovian queue-based service system with vacation interruption by employing different methodology (cf. [23], [254], [220], [173], [255]) for designing better service system. Liu et al. [181] deliberated a working vacation and working vacation interruption with active and standby redundancy. Tao et al. [174] derived the formula for the probability generating functions of the number of customers in orbit by using the stability conditions for an $M/M/1$ retrial queueing model with working vacation interruption. Later, [264] established the customers' equilibrium and socially optimal joining-balking behavior in single-server Markovian queues considering the system states as observable, partially observable, and unobservable, respectively. Bouchentouf and Yahiaoui [35] deduced the explicit expressions of the system sizes when the server is in a normal service period and in a Bernoulli scheduled vacation interruption in Markovian feedback queueing model with reneging and retention of reneged customers, multiple working vacations, and Bernoulli scheduled vacation interruption. Recently, for studying the effect of newly introduced emergency vacation policy on service systems, [233] provided a comparative analysis of optimal service strategies using Bat and PSO algorithms.

The efficiency of the metaheuristic algorithm plays an important role in finding optimal solutions for minimizing the cost function involves in the queueing problem having complexity of calculation. Particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), etc. are some common useful nature-inspired algorithm associated with optimal analysis of queueing problem ($cf.$ [178], [279], [299], [302]). In the present chapter, we use a metaheuristic optimization algorithm, cuckoo search (CS), for efficient search of the optimized solution up to a certain level in a stochastic manner. This algorithm was developed by [305] in 2009 on the motivation of brood parasitism behavior of cuckoos. Cuckoos lay their eggs in communal nests of other species to achieve reproduction of cuckoos. The cuckoo search algorithm is very much based on the identification of cuckoo eggs among host eggs in host nest. After identifying cuckoo eggs, host birds can either leave the nest and build a new nest or abandon these eggs. Some evolved cuckoos are also specialized in mimicking the eggs of host birds such that probability of identification and leaving nest decreases, and the likelihood of reproduction increases. The newly evolved soft computing technique, cuckoo search performs much better than PSO and other algorithms. Due to its guaranteed global convergence property, it is suitable for multi-modal optimization problem also. A number of algorithmic parameters to be tuned in cuckoo search is less than those in PSO and some other population-based optimization algorithms. Thus, it covers a large set of optimization problems being a more generic approach ($cf.$ [276], [36], [268], [306], [309]). Unfortunately, we have very few investigations in the literature on the cuckoo search for queueing problems. Woźniak [288] used computational power of cuckoo search for positioning $GI/M/1/N$ finite buffer queue with single vacation policy.

The target of the vacation interruption is to control the service process, which focuses on reducing the likelihood of the synchronized reneging of customers, which are to be served in the system and lessening expected waiting time. In real-world applications, the model discussed in this chapter is quite useful due to the consideration of synchronized reneging of the customers and vacation interruption of the server's working vacation. Such a queueing model frequently occurs in the area of computer processing, transportation systems and so on. A practical problem related to a computer processing system is provided for illustration purposes also. If the processor is unavailable due to working vacation, indicating that it is not currently working on packets, which are to be processed. Then, the packet is temporarily stored in a finite-size buffer for being served sometime later. When the buffer runs full at any time, newly arriving packets are lost. Packets may also lose some vital information due to behavior in processing and likely to follow synchronized reneging from the processing system. On the other hand, the processor will switch a normal processing rate whenever all waiting packets overload the processing system more than a pre-specified threshold. However, the processor can switch a lower processing rate for the random period when there is no waiting packet to be processed in the buffer and continues in that state until any packet request arrives at any time. It is helpful to prevent a computing device from becoming overloaded, overheating, and enhance the device's performance.

The investigated queue-based model is applicable in many areas of the service systems and optimal strategies with promising efficiency. The real purpose of the present study is threefold. The first objective is to develop a stochastic model via queueing theoretical approach for the service system with contrary assumptions for the customers and the server like synchronized reneging, working vacation, and vacation interruption. The second objective is to deliberate newly developed efficient nature-inspired optimization technique, cuckoo search, for queueing/waiting line problem. The third objective is to provide an exclusive optimal strategic repair policy for the governing queueing model with intensive sensitivity analysis.

The body of the chapter is as follows: In section 4.2, we describe the model using some assumptions and notations and formulate the governing differential-difference equations. In section 4.3, we derive some measure of effectiveness for performance analysis. In section 4.4, we establish the expected cost function to obtain the optimal operating policy with the minimal anticipated cost of the service system. Next, for the optimal analysis, the nature-inspired CS algorithm is used to obtain optimal system design parameters in subsection 4.5.1. The findings of CS algorithm are compared with the global optimization technique: PSO algorithm and semi-classical optimizers QN & DS method in subsections 4.5.2, 4.5.3, 4.5.4, respectively. In section 4.6, we summarize the model numerically for analyzing the system performance and future designing of the system. Finally, in section 4.7, we remark the conclusion and discuss the future scope.

4.2 Model Description

In this section, the mathematical modeling of a service system having finite capacity K is presented using the queueing terminologies, namely, working vacation, vacation interruption, and synchronized reneging. The specialty of such a queueing environment is that the customers waiting in the service system abandon the system simultaneously rather than the standard assumption in which they abandon the service system independently. For more justified analysis, the following are some useful assumptions and notations which we have used in the present study.

Arrival Process

- The customers arrive in the service system according to the Poisson process with a mean rate λ. If there is no customer in the waiting line, the arrived customer gets the required service immediately. Otherwise, customer joins the queue and waits for his turn.
- The newly arrived customers join the system following the service discipline *First Come First Serve (FCFS).*

Service Process

- The service time follows an identically and independently distributed exponential distribution with rate parameter μ_b in normal working mode.
- During vacation, the server continues to provide the service to the customers in the waiting line with exponentially distributed service times having a slower mean rate parameter $\mu_v(<\mu_b)$ and the prospective customer continues to join the waiting line of the service system.

Vacation Policy

- On completion of all the services, if there is no waiting customer in the service system, the server adopts a working vacation of the random interval of time. The time-of-vacation is exponentially distributed with mean time $\left(\frac{1}{\theta}\right)$. The server is permitted to continue for another working vacation at the end of the vacation period if there is no customer waiting in the service system *i.e.* the server follows multiple working vacation policies.
- In working vacation, if the server finds at least S waiting customers in the service system at the epoch of completion of his service for any customer, the server's vacation is interrupted immediately, and the server resumes the normal working attributes.

Impatient Behavior

- Due to long expected waiting time when the server is on a working vacation, some or all waiting customers may abandon simultaneously from the service system following binomial distribution with parameter p, where p is the probability of abandon of each customer who abandons the service system independent to the other at any epoch.

- The time-to-abandon follows an exponential distribution with rate parameter ζ and the abandoned customer will never rejoin the service system.

All processes or events are independent of the state of the other.
Next, for the brevity, we have used the notation $\phi_k^i = \binom{i}{k} p^k q^{i-k} \zeta$; $q = (1-p)$ in the developed finite capacity service system. The governing system of Chapman-Kolmogorov differential-difference equations for computing the steady-state probabilities associated with all system states are constructed by taking appropriate transition rates. For the Markovian modeling of the finite capacity service system with vacation interruption and synchronized reneging, the states of the service system at any instant t are defined as

$$J(t) \equiv \begin{cases} 0; & \text{Server is on working vacation at the instant } t \\ 1; & \text{Server is on normal busy state at the instant } t \end{cases}$$

and

$$N(t) \equiv \text{Number of the customers in the service system at any instant } t.$$

Then, $\{J(t), N(t); t \geq 0\}$ represents the continuous-time Markov chain (CTMC) on the state space $\Theta \equiv \{(0,0) \cup (j,n); j = 0,1 \,\text{and}\, n = 1,2,\ldots,K\}$. Let us define the state probabilities at any instant t as

$$P_{0,n}(t) = \text{Prob}[J(t) = 0 \,\&\, N(t) = n]; n = 0,1,\ldots,K$$
$$P_{1,n}(t) = \text{Prob}[J(t) = 1 \,\&\, N(t) = n]; n = 1,\ldots,K$$

For steady-state analysis, we have limiting probabilities $P_{0,n} = \lim_{n\to\infty} P_{0,n}(t)$ and $P_{1,n} =$

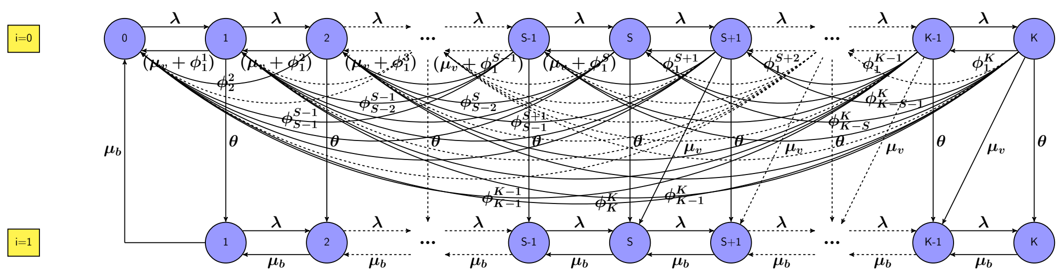

Figure 4.1: Transtion-state diagram

$\lim_{n\to\infty} P_{1,n}(t)$ which satisfy following set of forward Chapman-Kolomogrov homogenous system of differential-difference equations governed by Fig. 4.1.

$$-\lambda P_{0,0}+\mu_v P_{0,1}+\zeta\left\{\sum_{m=1}^{K} p^m P_{0,m}\right\}+\mu_b P_{1,1}=0 \tag{4.1}$$

$$-(\lambda+\theta+\mu_v+\zeta)P_{0,n}+\lambda P_{0,n-1}+\zeta\left\{\sum_{m=n}^{K}\binom{m}{m-n}p^{m-n}q^n P_{0,m}\right\}+\mu_v P_{0,n+1}=0;\quad 1\le n\le S-1 \tag{4.2}$$

$$-(\lambda+\theta+\mu_v+\zeta)P_{0,S}+\lambda P_{0,S-1}+\zeta\left\{\sum_{m=S}^{K}\binom{m}{m-S}p^{m-S}q^S P_{0,m}\right\}=0 \tag{4.3}$$

$$-(\lambda+\theta+\mu_v+\zeta)P_{0,n}+\lambda P_{0,n-1}+\zeta\left\{\sum_{m=n}^{K}\binom{m}{m-n}p^{m-n}q^n P_{0,m}\right\}=0;\quad S+1\le n\le K-1 \tag{4.4}$$

$$-(\theta+\mu_v+\zeta)P_{0,K}+\lambda P_{0,K-1}+q^K\zeta P_{0,K}=0 \tag{4.5}$$

$$-(\lambda+\mu_b)P_{1,1}+\theta P_{0,1}+\mu_b P_{1,2}=0 \tag{4.6}$$

$$-(\lambda+\mu_b)P_{1,n}+\theta P_{0,n}+\mu_b P_{1,n+1}+\lambda P_{1,n-1}=0;\ 2\le n\le S-1 \tag{4.7}$$

$$-(\lambda+\mu_b)P_{1,n}+\theta P_{0,n}+\mu_v P_{0,n+1}+\lambda P_{1,n-1}+\mu_b P_{1,n+1}=0;\ S\le n\le K-1 \tag{4.8}$$

$$-\mu_b P_{1,K}+\theta P_{0,K}+\lambda P_{1,K-1}=0 \tag{4.9}$$

The normalizing condition of probability is given by

$$P_{0,0}+\sum_{j=0}^{1}\sum_{n=1}^{K}P_{j,n}=1. \tag{4.10}$$

The governing steady-state differential-difference equations (4.1)-(4.9) of the finite capacity service system under some realistic assumptions can be expressed in the form of homogeneous system of linear equations

$$\mathbf{QX}=\mathbf{0} \tag{4.11}$$

where, $\mathbf{Q}$ is the coefficient matrix of order $(2K+1)$ and $\mathbf{X}$ is the column vector of state probabilities having elements $[P_{0,0}, P_{0,1}, P_{0,2}, \ldots, P_{0,K}, P_{1,1}, P_{1,2}, \ldots, P_{1,K}]^T$ and $\mathbf{0}$ is the null column vector of order $(2K+1)$. After using the normalization condition (4.10), the above mentioned system of linear equations (4.11) is converted into the form

$$\mathbf{Q}^*\mathbf{X} = \mathbf{B} \tag{4.12}$$

where, $\mathbf{Q}^*$ is the matrix obtained by replacing all the entries in last row of the matrix $\mathbf{Q}$ by one *i.e.* last row of $\mathbf{Q}$ by row vector $\mathbf{e}_{2K+1} = [1, 1, \ldots, 1\,(2K+1)\text{times}]$ and $\mathbf{B}$ is the column vector having the form $[0, 0, \ldots, 1]^T$ of order $(2K+1)$.

Since, $\mathbf{Q}^*$ is non-singular in nature and system of equations (4.12) is non-homogeneous, the solution can easily be determined as $\mathbf{X} = (\mathbf{Q}^*)^{-1}\mathbf{B}$, which satisfies basic law of probabilities.

4.3 System Performance Measures

In performance modeling, there are certain generic performance measures using which the performance of the studied finite queue-based service system with synchronized reneging and vacation interruption can be described for the decision purpose. These measures are quite interrelated, and each assumes increased importance in a particular context. In this section, we obtain these service system performance measures by using the steady-state probabilities computed in the previous section as follows

- Expected number of the customers in the service system

$$L_S = \sum_{j=0}^{1} \sum_{n=j}^{K} n P_{j,n} \tag{4.13}$$

- Expected number of the customers in the service system when the server is on working vacation

$$L_{S_v} = \sum_{n=1}^{K} n P_{0,n} \tag{4.14}$$

- Expected number of the customers in the service system when the server is on regular busy period

$$L_{S_b} = \sum_{n=1}^{K} n P_{1,n} \tag{4.15}$$

- Probability that the server is busy

$$P_B = \sum_{n=1}^{K} P_{1,n} \tag{4.16}$$

- Probability that the server is on working vacation

$$P_{WV} = \sum_{n=0}^{S} P_{0,n} \tag{4.17}$$

- Probability that server's vacation is interrupted

$$P_{VI} = \sum_{n=S+1}^{K} P_{0,n} \tag{4.18}$$

- Effective reneging rate during vacation period

$$RR = \sum_{n=1}^{K} (1-q^n)\,\zeta P_{0,n} \tag{4.19}$$

- Throughput of the service system

$$\tau_p = \sum_{n=1}^{K} \mu_v P_{0,n} + \sum_{n=1}^{K} \mu_b P_{1,n} \tag{4.20}$$

- Expected waiting time of the customers in the service system

$$W_S = \frac{L_S}{\lambda_{\text{eff}}} \tag{4.21}$$

where, $\lambda_{\text{eff}} = \sum_{n=0}^{K-1} \lambda P_{0,n} + \sum_{n=1}^{K-1} \lambda P_{1,n}$, a effective arrival rate.

4.4 Cost Analysis

In this section, we develop a steady-state expected total cost function for the finite queue-based service system with synchronized reneging and vacation interruption, in which three decision variables S, μ_v and μ_b are considered. The decision variable S is required to be a positive integer (Z^+), and the continuous variables μ_v and μ_b are non-negative real numbers. Our main objective is to decide the optimum pre-specified threshold level S, say S^* and the optimum values of service rates μ_v, μ_b, say μ_v^*, μ_b^* respectively so as to minimize the unconstrained expected total cost function.

Having a quantitative idea about the system performance, we formulate the expected total cost function using various cost elements associated with the states of the service system. Now, we define the specific cost elements as follows

$C_h \equiv$ Cost per unit time associated with each customer in the service system.

$C_b \equiv$ Cost incured due to the regular busy period of the server.

$C_i \equiv$ Cost per unit time associated with server's vacation interruption.

$C_r \equiv$ Cost per unit time associated with reneging of each customer.

$C_1 \equiv$ Service cost per unit time per customer during a normal working period.

$C_2 \equiv$ Service cost per unit time per customer during a working vacation period.

Thus, on the basis of the definitions of cost elements listed above, the expected total cost function per unit time is framed as

$$TC(S,\mu_v,\mu_b) = C_h L_S + C_b P_B + C_i \mu_v P_{VI} + C_r RR + C_1 \mu_b + C_2 \mu_v; \qquad S \in Z^+ \text{ and } \mu_v, \mu_b \in R^+ \tag{4.22}$$

The cost minimization problem of the studied service model can be presented mathematically as unconstraint problem

$$TC(S^*,\mu_v^*,\mu_b^*) = \underset{(S,\mu_v,\mu_b)}{\text{minimize}} \quad TC(S,\mu_v,\mu_b) \tag{4.23}$$

The foremost objective is to find the optimal predefined threshold value S^* and the values of service rates μ_v^* and μ_b^* simultaneously, which minimize the expected total cost function. The analytic analysis of the expected total cost function would be extremely difficult since cost function is non-linear of high order in nature and can not be represented explicitly. We adopt metaheuristic optimizing techniques for determining the optimal value of decision variables. We employ nature inspired optimizing techniques, cuckoo search and particle swarm optimization, for global optimization of expected total cost function of governing model.

4.5 Optimal Analysis

In general, mostly stochastic optimization problems are of high non-linear and complex nature because of the involvement of various complex constraints. Therefore, it is an arduous task for scholars and researchers to solve such problems analytically. In addition, several associated cost elements with numerous quality performance measures also make them more complicated. So, for the system designers and engineers, it is necessary to opt for an efficient alternative technique, such as heuristics and metaheuristics, as these techniques can solve such complex optimization problems efficiently. In the present study, we first implement the CS algorithm to obtain the optimal strategies of system design parameters along with the minimal expected cost of the service system. Then for the validity of the obtained results by CS algorithm, we compare the findings with the swarm intelligence based optimization technique: particle swarm optimization (PSO) algorithm and semi-classical optimization techniques: quasi-Newton (QN) method and direct search (DS) method. For that purpose, we use S, μ_v, μ_b, TC instead of x_1, x_2, x_3, and f respectively in the following sub-sections. The detailed algorithmic steps, along with their pseudo-codes for all the algorithms, are provided in the next sub-sections.

4.5.1 Cuckoo Search Algorithm

For the detailed study of the CS algorithm, refer the section 1.10.4 and its pseudo-code.

4.5.2 Particle Swarm Optimization

For the PSO algorithm, refer the section 1.10.3.

4.5.3 Quasi-Newton Method

For implementing the Quasi-Newton algorithm for continuous system design parameters μ_v and μ_b, refer the section 1.10.1. Further, to see the algorithmic steps of the Quasi-Newton method for queueing related models, cite the brief explanation given in research papers [233], [279], [301], and references therein.

4.5.4 Direct-Search Method

To search the optimal value of the discrete system design parameter S, refer the Direct-Search algorithm explained in section 1.10.2.

Now, to find the optimal solution of the governing cost optimization problem, some numerical experiments are performed, and the results are depicted in various graphs and tables using CS, PSO algorithm, QN method, and DS method for the comparative analysis purpose in the next section.

4.6 Numerical Results

In this section, we accomplish the numerical results through various tables and graphs to validate our formulation. As the analytical solution of the performance, measures are not sufficient to analyze the capability of the service system, and hence we obtain the performance indices numerically. The numerical illustration of the performance measures will be of great help to the system engineers and decision-makers in improving the system performance and future designing of the service system.

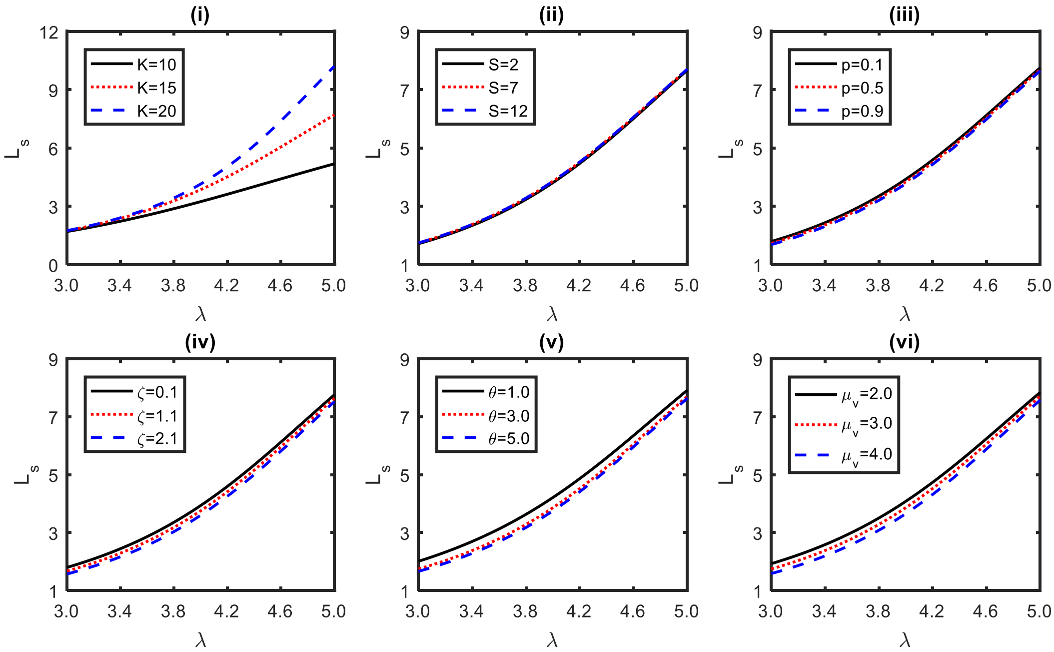

Figure 4.2: Expected number of customer in the service system (L_S) wrt λ for (i) K, (ii) S, (iii) p, (iv) ζ, (v) θ, and (vi) μ_v

Using the MATLAB program, we obtain the probabilities for the different states of the service system. For the computation purposes, we fix the default parameters of the service system as S=7, K=15, λ=4.0, μ_v=3.0, μ_b=5.0, θ=3.0, p=0.4, and ζ=0.5 and results are depicted in Figs. 4.2–4.7 and Tables 4.1–4.2. Figs. 4.2 and 4.3 represent the variability of the expected number of the customers in the service system (L_S) for arrival rate λ and service rate μ_b at normal state, respectively for different values of K, S, p, ζ, θ, and μ_v. From Figs. 4.2 & 4.3, It is depicted that L_S is increasing for the increased value of λ and decreasing wrt to μ_b, respectively. Both trends of change are obvious. It is also depicted in Figs. 4.2(i) and 4.3(i) that L_S is higher for higher capacity of the system K. No difference in L_S is observed for the change in the threshold value S in Fig. 4.2(ii). Similarly for any test instance, if we fix the value of λ and increase the value of ζ & μ_v, the length of the system decreases. Fig. 4.2(v)

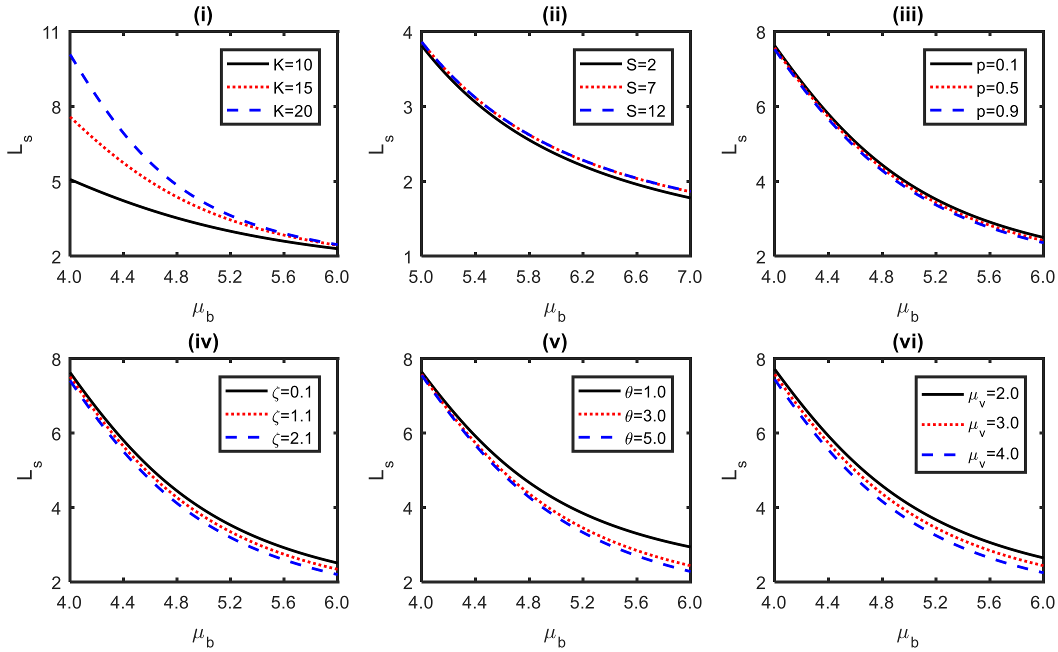

Figure 4.3: Expected number of customer in the service system (L_S) wrt μ_b for (i) K, (ii) S, (iii) p, (iv) ζ, (v) θ, and (vi) μ_v

reveals that as we increase the value of θ, the length of the system decreases. It is apparent from the fact that the mean vacation rate (θ) is inversely proportional to the vacation time. Henceforth, the incremental change in θ decrease the vacation time of the server which results in the decrease in the value of L_S. Fig. 4.3 also illustrates that a higher service rate decreases the system length (L_S) to some extent only. The extra effort for maintaining the higher service rate is unworthy and needs to establish optimal service rate.

Figs. 4.4 and 4.5 represent the trends of the throughput of the service system (τ_p) for arrival rate λ and service rate μ_b respectively. The high value of throughput is obviously observed if there is a large number of the customers in the service system. The throughput of the service system is high for the high value of λ and low value of μ_b. It is also observed that τ_p is increasing for K and decreasing for p, and ζ. Fig. 4.4(ii) shows that different values of S are ineffective for the measurement of throughput. Figs. 4.4(v) and 4.5(v) exhibit interesting fact for vacation time. In Fig. 4.5(i)-(ii), one can easily observed that as we increase the value of service rate μ_b, the value of τ_p increases upto a certain threshold and then decreases. Therefore, as a conclusive remark, high service facility is not always useful to system analysts and engineers in decision making in order to maintain the quality performance of service systems.

For the different system parameters, the variation in the value of the expected

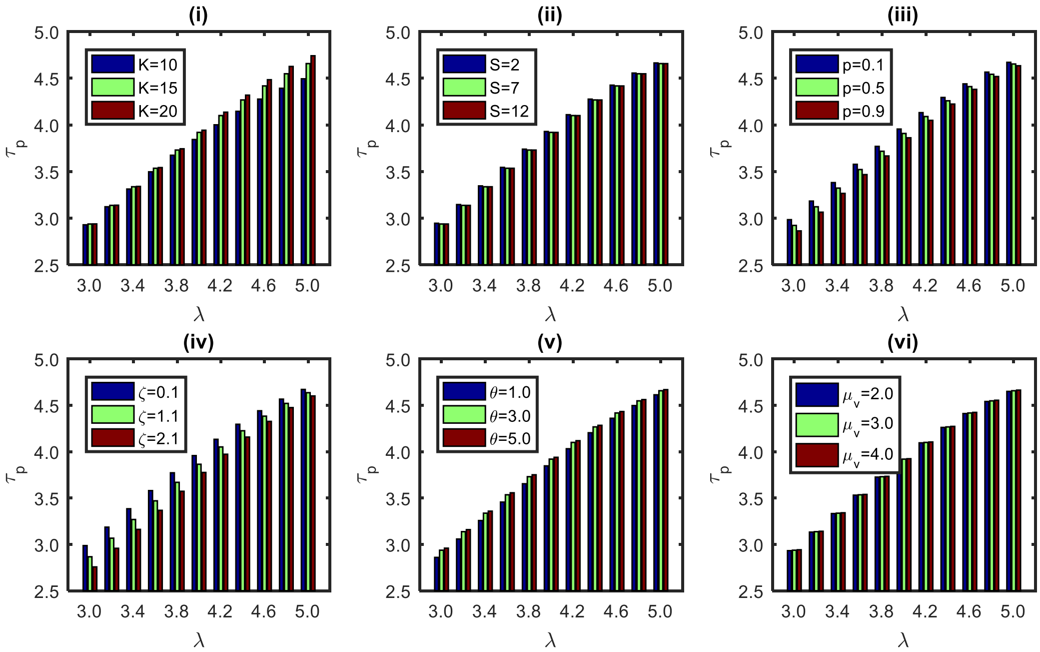

Figure 4.4: Throughput of the service system (τ_p) wrt λ for (i) K, (ii) S, (iii) p, (iv) ζ, (v) θ, and (vi) μ_v

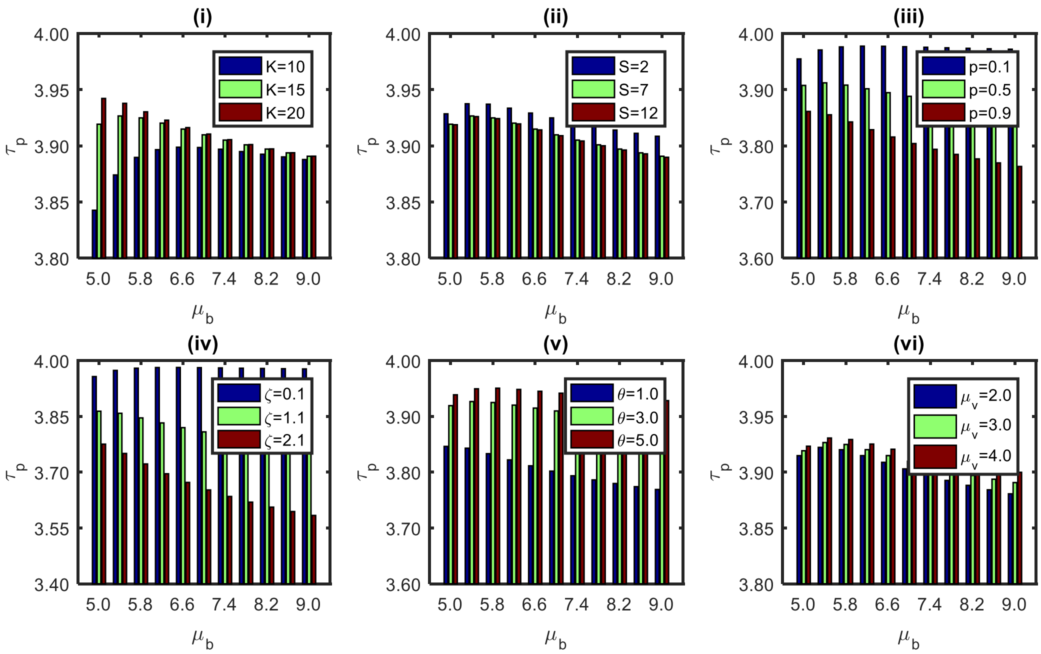

Figure 4.5: Throughput of the service system (τ_p) wrt μ_b for (i) K, (ii) S, (iii) p, (iv) ζ, (v) θ, and (vi) μ_v

total cost (TC) defined in eq^n(4.22) is exhibited wrt arrival rate λ and service rate μ_b in the normal busy state in Figs. 4.6 and 4.7 respectively. For the computation of the TC, we fix following unit cost: $C_h = 100$, $C_b = 80$, $C_i = 20$, $C_r = 300$, $C_1 = 60$, and

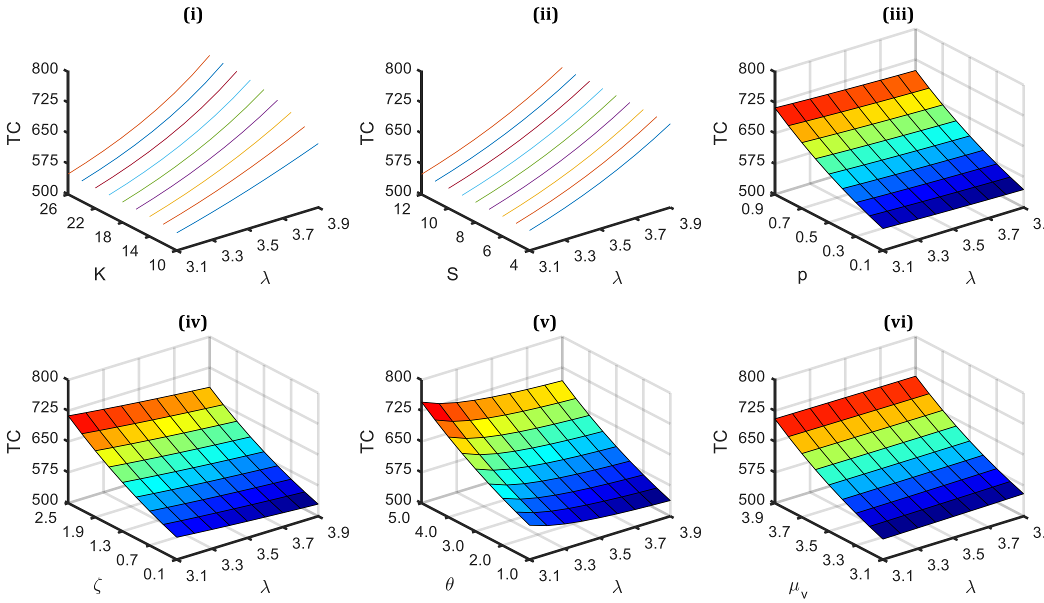

Figure 4.6: Expected total cost (TC) wrt λ for (i) K, (ii) S, (iii) p, (iv) ζ, (v) θ, and (vi) μ_v

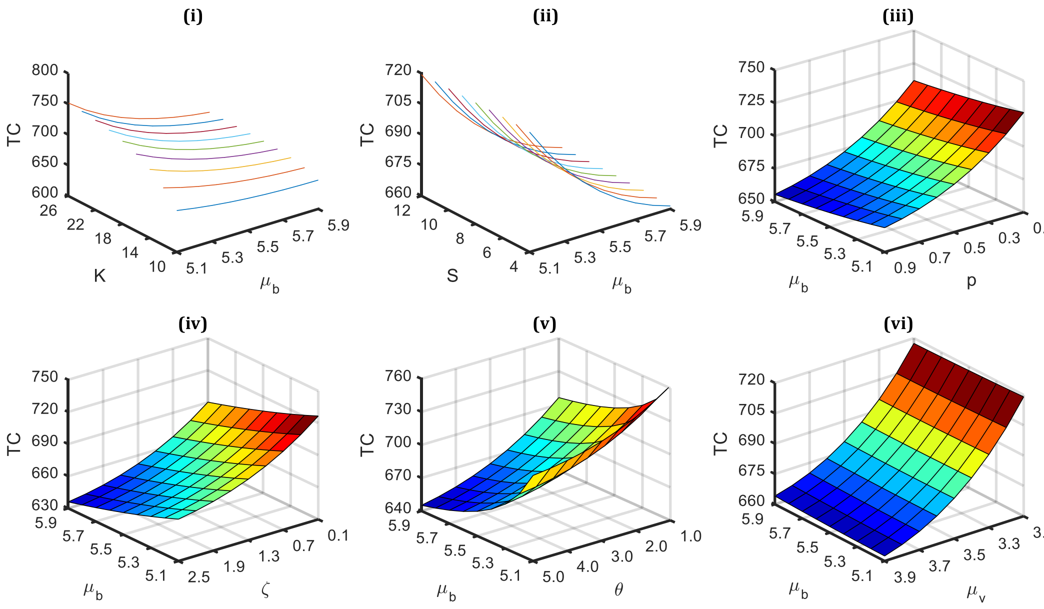

Figure 4.7: Expected total cost (TC) wrt μ_b for (i) K, (ii) S, (iii) p, (iv) ζ, (v) θ, and (vi) μ_v

$C_2 = 10$, as default cost elements. Figs. 4.6 and 4.7 prompt that all governing system parameters are worthy in system modeling and play key role in the service system designing. These figures also exhibit that some decision variables need to be optimal for the minimum expected total cost.

Table 4.1: Numerical simulation of various system performance measures on varying K, λ, μ_b, and θ.

$(K, \lambda, \mu_b, \theta)$	L_{S_v}	L_{S_b}	P_{WV}	P_{VI}	RR	W_S
(10, 3.5, 4.5, 2.5)	0.283675	2.699614	0.357928	3.00E-04	4.35E-02	0.870860
(15, 3.5, 4.5, 2.5)	0.269379	3.194820	0.339860	2.85E-04	4.13E-02	0.995484
(20, 3.5, 4.5, 2.5)	0.265563	3.397597	0.335046	2.81E-04	4.07E-02	1.048305
(10, 4.0, 4.5, 2.5)	0.227307	3.826611	0.245288	4.03E-04	3.36E-02	1.066965
(10, 4.5, 4.5, 2.5)	0.165350	4.986089	0.155598	4.46E-04	2.36E-02	1.263826
(10, 3.5, 5.0, 2.5)	0.347812	2.077275	0.438853	3.68E-04	5.33E-02	0.700249
(10, 3.5, 5.5, 2.5)	0.399864	1.655934	0.504530	4.23E-04	6.13E-02	0.590627
(10, 3.5, 4.5, 3.0)	0.241478	2.714610	0.344686	1.74E-04	3.80E-02	0.862657
(10, 3.5, 4.5, 3.5)	0.210198	2.724746	0.334303	1.05E-04	3.38E-02	0.856293

Table 4.2: Numerical simulation of various system performance measures on varying S, μ_v, ζ, and p.

(S, μ_v, ζ, p)	L_{S_v}	L_{S_b}	P_{WV}	P_{VI}	RR	W_S
(5, 2.5, 0.1, 0.2)	0.243023	3.802614	0.270832	2.25E-03	4.16E-03	1.020173
(7, 2.5, 0.1, 0.2)	0.250337	3.808712	0.273169	5.14E-04	4.25E-03	1.023620
(9, 2.5, 0.1, 0.2)	0.252677	3.810215	0.273719	1.17E-04	4.27E-03	1.024616
(5, 3.0, 0.1, 0.2)	0.238414	3.697225	0.285005	1.83E-03	4.12E-03	0.992122
(5, 3.5, 0.1, 0.2)	0.233984	3.595023	0.299254	1.49E-03	4.08E-03	0.964959
(5, 2.5, 0.5, 0.2)	0.242300	3.762731	0.275367	2.07E-03	2.08E-02	1.009800
(5, 2.5, 0.9, 0.2)	0.241530	3.724134	0.279842	1.91E-03	3.75E-02	0.999748
(5, 2.5, 0.1, 0.4)	0.242797	3.793635	0.271932	2.22E-03	7.27E-03	1.017824
(5, 2.5, 0.1, 0.6)	0.242529	3.785880	0.272981	2.20E-03	9.66E-03	1.015778

Tables 4.1 & 4.2 summarize the performance measures viz expected number of customers (L_{S_v}, L_{S_b}) in the different states of the service system, different state probabilities (P_{WV}, P_{VI}), effective reneging rate (RR) and expected waiting time of the customers in the service system (W_S) for different combinations of system parameters. These tables display the variability at a glance of performance measures wrt system parameters, which provides more accurate estimation to uncertain environments. The research findings would help system analyst to make a better decision in order to have the optimal service strategy based on the desired performance measures.

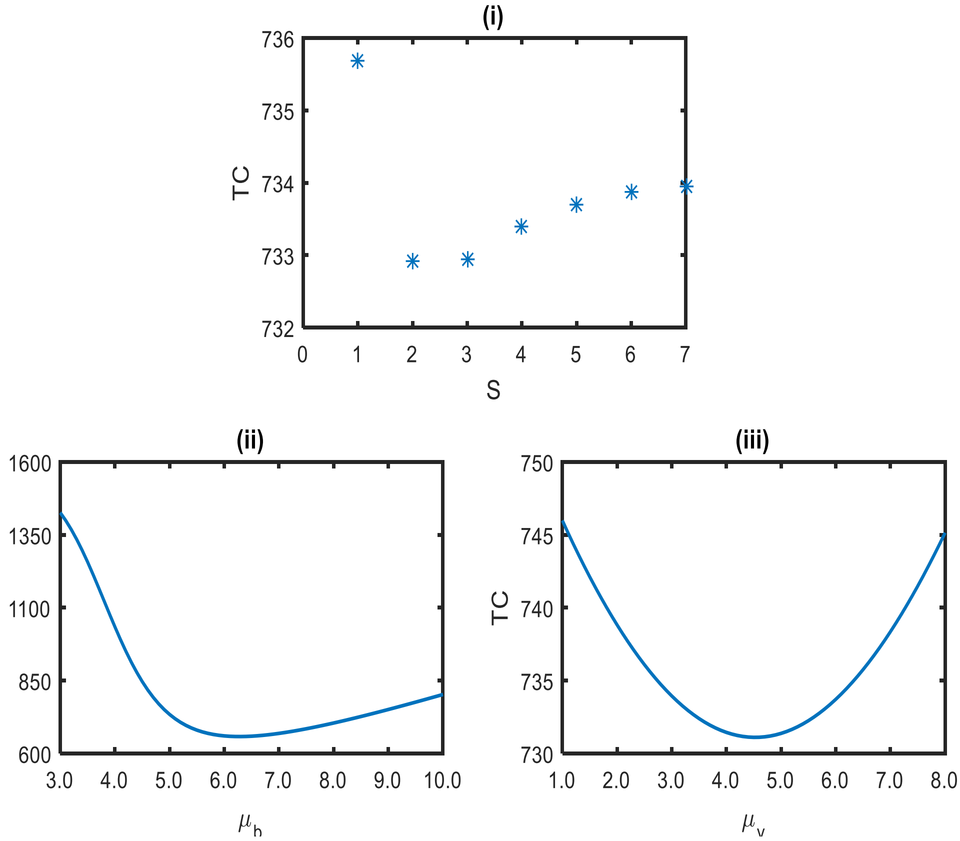

Figure 4.8: Convex expected total cost function (TC) wrt parameter (i) S, (ii) μ_b, and (iii) μ_v

For predicting the optimal strategy of the studied service system using some optimizing technique, we prove that the governing expected total cost of the service system is convex in nature graphically since analytical proof has a high degree of difficulty due to intrinsic nature of the cost function and typical to evaluate. In Fig. 4.8, we depict the shape of the expected total cost function for decision system parameters considering the cost elements similar as in Figs. 4.6 & 4.7. Fig. 4.8 infers that the studied expected total cost function is convex in nature wrt to decision parameters S, μ_v and μ_b.

Table 4.3: Optimal design parameters S^*, μ_v^*, and μ_b^* with minimal expected cost using CS algorithm

$(K, \lambda, p, \zeta, \theta)$	S^*	$\mu_v{}^*$	$\mu_b{}^*$	TC^*	$Mean\left\{\frac{TC}{TC^*}\right\}$	$Max\left\{\frac{TC}{TC^*}\right\}$	CPU time
$(15, 4.0, 3.0, 0.5, 0.4)$	3	2.455630	6.313114	655.930864	1.0000002470	1.0000004449	366.431
$(20, 4.0, 3.0, 0.5, 0.4)$	3	2.459998	6.363811	657.205981	1.0000000885	1.0000002577	672.697
$(25, 4.0, 3.0, 0.5, 0.4)$	3	2.459746	6.372171	657.363019	1.0000000769	1.0000002213	933.448
$(15, 3.8, 2.5, 1.0, 0.5)$	4	2.345192	6.023757	628.645873	1.0000000159	1.0000000342	646.049
$(15, 4.2, 2.5, 1.0, 0.4)$	3	2.856273	6.507472	679.760661	1.0000000021	1.0000000034	784.038
$(15, 4.0, 2.5, 0.5, 0.4)$	2	3.144711	6.241353	663.198746	1.0000001230	1.0000003051	666.465
$(15, 4.0, 3.5, 0.5, 0.4)$	4	1.663650	6.385354	648.224219	1.0000002715	1.0000006383	614.731
$(15, 4.0, 2.5, 1.0, 0.4)$	3	2.714446	6.261816	656.009030	1.0000003573	1.0000006589	676.592
$(15, 4.0, 2.5, 1.3, 0.4)$	4	2.434002	6.271396	651.542521	1.0000049835	1.0000068298	565.251
$(15, 4.0, 2.5, 1.0, 0.5)$	3	2.566887	6.264654	652.861800	1.0000000089	1.0000000157	624.372
$(15, 4.0, 2.5, 1.0, 0.3)$	3	2.879377	6.257603	659.340484	1.0000000642	1.0000001287	641.933

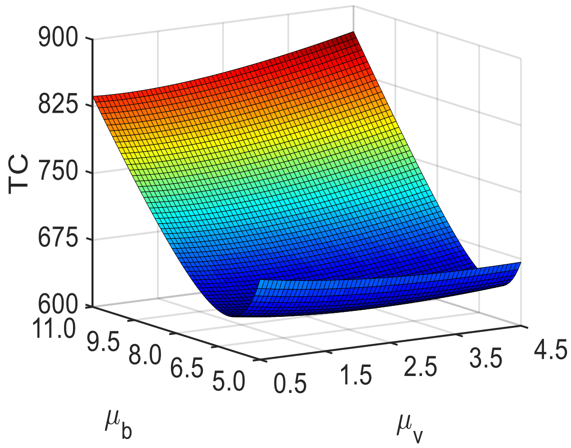

Figure 4.9: Surface plot for TC wrt μ_v, and μ_b

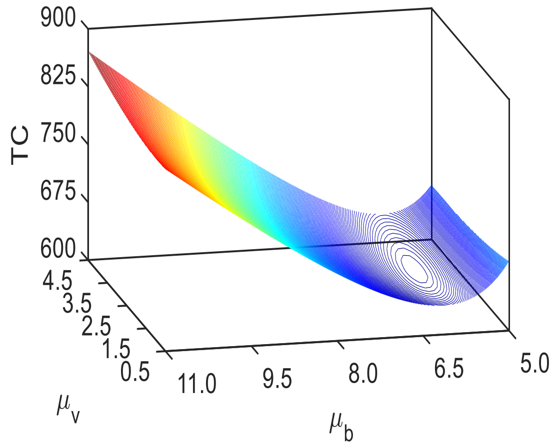

Figure 4.10: Three dimension plot for TC wrt μ_v, and μ_b

To examine the economic performance of the CS algorithm for the cost minimization problem for the service system, we assess the results tabulated in Table 4.3. The CS algorithm is coded in MATLAB and all the numerical experiments for different data sets of system parameters are summarized in Table 4.3. For each of the tested examples, we set the system capacity $K = 15$, unit cost elements as appeared in Figs. 4.6 & 4.7 and the default parameters of CS algorithm as $\omega_1 = 1.5$, $p_a = 0.25$, $\Theta_1 = 0.01$. The lower and upper bound of the decision variable S are 2 and 15 respectively. Similarly, the ranges for the system design parameters μ_v and μ_b are fixed as [0.5 4.5] and [5 11] respectively. Table 4.3 shows the numerical results of 100 independent experiments for each example by the CS algorithm. The corresponding associated optimal values for decision variables S^*, μ_v^*, and μ_b^* are mentioned in the table along with the optimal cost of the system. Note that, for the convenience and better understanding, the statistical parameters mean and maximum ratios are also computed. The ratio of the solution produced by the CS algorithm is calculated by $\left(\frac{TC}{TC^*}\right)$, where TC is the solution generated by the CS algorithm and TC^* is the minimum solution among 100 independent experiments. From Table 4.3, we observe that (1) the mean values $\left(\frac{TC}{TC^*}\right)$ of the CS algorithm vary from 1.0000000021 to 1.0000049835. It reveals the strength of the CS algorithm for all the test instances. (2) the max values $\left(\frac{TC}{TC^*}\right)$ of the CS algorithm vary from 1.0000000034 to 1.0000068298, which implies that the searching quality of the CS algorithm is very well. It is also observed that the CS algorithm is capable of solving the test instances within a reasonable time. Table 4.3 illustrates the optimal value of decision parameters S, μ_v and μ_b in combination. From the optimal tableau, we infer for the optimal service rates and the threshold value for vacation interruption for the governing set problem parameters of the studied service system. For any type of increased failure likelihood, higher service rates are require with additional expected total cost to maintain expected waiting time in the system upto some extent.

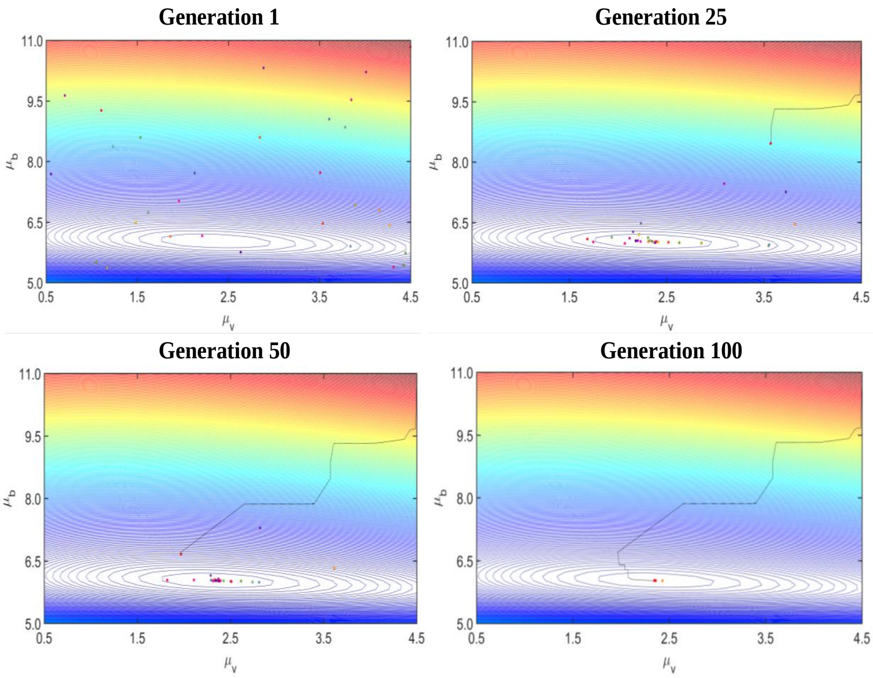

Figure 4.11: Generations of the CS algorithm wrt the optimal pair (μ_v, μ_b)

Next, with the help of Fig. 4.11, some selected generations of CS algorithm in the feasible domain are provided for the illustrative purpose. Fig. 4.11 demonstrates the convex nature of the expected cost function (4.22) for the combined optimal values of decision parameters μ_v and μ_b along with the optimal anticipated cost of the system. In addition, to emphasize the convex nature of the expected cost function for the combined optimal values of design parameters μ_v and μ_b, a three dimensional surface plot and contour plot are depicted in Figs. 4.9 & 4.10, respectively. Because the CS algorithm is an agent-based stochastic optimization technique, one can easily observe from the generation 1 that initially all the agents are randomly scattered in the entire feasible region and come closer and closer very quickly by exploring the untouched area as in generations 25, 50, and 100. To perform the optimal analysis, we fix the default value of system parameters and associated cost elements as: $K = 15$, $\lambda = 3.8$, $\theta = 2.5$, $\zeta = 1.0$, $p = 0.5$, $C_h = 100$, $C_i = 20$, $C_b = 80$, $C_r = 300$, $C_1 = 60$, and $C_2 = 10$. The generations of the CS algorithm demonstrate its ability to function strongly and to approach to the optimality within a justifiable time interval. The coordinates of the best agent with the best position using CS algorithm are obtained as $[S^*, \mu_v^*, \mu_b^*] = [4, 2.345192, 6.023757]$ along with the minimal expected cost of the system $TC^* = 628.645873$.

Moreover, a comparative analysis of the findings of CS algorithm with the swarm intelligence based optimization technique: PSO algorithm, and semi-classical optimizers: QN method and DS method is executed. To perform the comparative analysis, we use the different data sets of system parameters, and the results are compared in Tables 4.3–4.8. In many of the engineering problems, in general, the computation time and optimal result are used to compare the quality performance and the efficiency of any algorithm. Henceforth, the results in Tables 4.3–4.4 are compared based on optimal operating policies, the value of statistical parameters, and the CPU time (in seconds). It is observed that the approximate optimal values for both discrete and continuous system design parameters along with the minimal expected cost obtained by the CS algorithm, PSO algorithm, and QN method are very close to each other. The CPU time for CS algorithm is a little bit higher than the PSO algorithm, but the corresponding minimal expected cost obtained by the CS algorithm converges more significantly for all test instances in comparison to PSO algorithm. It represents that the searching quality in the feasible domain, and the efficiency of the CS algorithm is higher than the PSO algorithm. The results of CS algorithm are also better than semi-classical optimizers: QN method and DS method for each numerical experiment, which reveals the robustness of the CS algorithm in comparison to the other optimizers.

From overall studies, we conclude that the cuckoo search is also a better alternative for computing the optimal cost of the studied problems and can be further used in many of the real-life engineering and management issues based.

Table 4.4: Optimal design parameters S^*, μ_v^*, and μ_b^* with minimal expected cost using PSO algorithm

$(K, \lambda, p, \zeta, \theta)$	S^*	${\mu_v}^*$	${\mu_b}^*$	TC^*	$Mean\left\{\frac{TC}{TC^*}\right\}$	$Max\left\{\frac{TC}{TC^*}\right\}$	CPU time
$(15, 4.0, 3.0, 0.5, 0.4)$	3	2.454718	6.312615	655.930864	1.0000895403	1.0002686209	369.912
$(20, 4.0, 3.0, 0.5, 0.4)$	3	2.461271	6.364605	657.206007	1.0000273809	1.0000813628	839.129
$(25, 4.0, 3.0, 0.5, 0.4)$	3	2.458064	6.372377	657.363025	1.0000956269	1.0002868744	907.391
$(15, 3.8, 2.5, 1.0, 0.5)$	4	2.342742	6.024270	628.645887	1.0000139740	1.0000209621	384.359
$(15, 4.2, 2.5, 1.0, 0.4)$	3	2.856182	6.508092	679.760662	1.0000000285	1.0000000625	394.634
$(15, 4.0, 2.5, 0.5, 0.4)$	2	3.144914	6.241691	663.198752	1.0000000761	1.0000000924	346.265
$(15, 4.0, 3.5, 0.5, 0.4)$	4	1.667457	6.385047	648.224252	1.0000318314	1.0000482396	242.219
$(15, 4.0, 2.5, 1.0, 0.4)$	3	2.716391	6.261328	656.009030	1.0000000543	1.0000000932	243.253
$(15, 4.0, 2.5, 1.3, 0.4)$	4	2.434305	6.271358	651.542522	1.0000744696	1.0002334043	370.656
$(15, 4.0, 2.5, 1.0, 0.5)$	3	2.567094	6.265535	652.861802	1.0000014423	1.0000043107	344.374
$(15, 4.0, 2.5, 1.0, 0.3)$	3	2.873351	6.256605	659.340484	1.0000009172	1.0000011513	359.874

Table 4.5: Optimal values of (μ_v^*, μ_b^*) with corresponding minimum cost TC^* for different S.

S	Initial value	μ_v^*	μ_b^*	TC^*	Iterations
$S=2$	[4.0, 8.0]	2.591212	5.996891	630.295601	7
$S=3$	[4.0, 8.0]	2.419055	6.017466	628.802729	9
$S=4$	[4.0, 8.0]	**2.345122**	**6.023837**	**628.645873**	**8**
$S=5$	[4.0, 8.0]	2.328528	6.023562	628.732661	9
$S=6$	[4.0, 8.0]	2.335526	6.021151	628.801052	8
$S=7$	[4.0, 8.0]	2.346266	6.018843	628.823950	8
$S=8$	[4.0, 8.0]	2.353866	6.017280	628.821152	8
$S=9$	[4.0, 8.0]	2.357856	6.016391	628.809845	8
$S=10$	[4.0, 8.0]	2.359441	6.015944	628.798418	8
$S=11$	[4.0, 8.0]	2.359819	6.015743	628.789613	8
$S=12$	[4.0, 8.0]	2.359653	6.015671	628.783636	8
$S=13$	[4.0, 8.0]	2.359367	6.015652	628.779879	8
$S=14$	[4.0, 8.0]	2.359070	6.015657	628.777643	8
$S=15$	[4.0, 8.0]	2.358671	6.015675	628.775383	8

Table 4.6: The illustration of the iterative process of Quasi-Newton method with $K=15$, $\lambda=3.8$, $\theta=2.5$, $\zeta=1.0$, & $p=0.5$ and initial value $(S,\mu_v,\mu_b)=(4,4.0,8.0)$

Iterations	0	1	2	3	4
μ_v	4.0	3.748015	3.456909	3.126884	2.760937
μ_b	8.0	7.000001	5.785820	6.061863	6.041227
$TC(S,\mu_v,\mu_b)$	701.658710	655.963302	631.743611	630.116894	629.063117

Iterations	5	6	7	**8**
μ_v	2.311734	2.346547	2.345115	**2.345122**
μ_b	6.025696	6.023865	6.023842	**6.023837**
$TC(S,\mu_v,\mu_b)$	628.648240	628.645878	628.645874	**628.645873**

4.7 Conclusion and Future Scope

In this chapter, we have studied the finite capacity service system with vacation interruption and synchronized reneging. The probabilistic interpretations of various performance measures and manifold numerical results with the aid of computer software MATLAB (2018b) have been explored. Next, we have formulated the expected cost function to determine the optimal joint values of several design parameters, namely, S, μ_b, and μ_v at minimum cost. Also, we have employed the CS algorithm to search for the global minimum of the expected total cost of the service system along with optimal design parameters. The comparative analysis with the swarm intelligence based optimization technique: PSO algorithm and semi-classical techniques: QN

Table 4.7: Optimal values of (S^*, μ_v^*, μ_b^*) with corresponding minimum expected cost TC^* for several combinations of $(\lambda, \theta, \zeta, p)$ via Quasi-Newton method.

$(\lambda, \theta, \zeta, p)$	$(4.0, 3.0, 0.5, 0.4)$	$(3.8, 2.5, 1.0, 0.5)$	$(4.2, 2.5, 1.0, 0.4)$	$(4.0, 2.5, 0.5, 0.4)$	$(4.0, 3.5, 0.5, 0.4)$
S^*	3	**4**	3	2	4
(μ_v^0, μ_b^0)	[4.0, 8.0]	[4.0, 8.0]	[4.0, 8.0]	[4.0, 8.0]	[4.0, 8.0]
Total Iteration	8	**8**	9	7	8
μ_v^*	2.457793	**2.345122**	2.855791	3.144495	1.666797
μ_b^*	6.312697	**6.023837**	6.508098	6.241254	6.384594
$TC(S^*, \mu_v^*, \mu_b^*)$	655.930866	**628.645873**	679.760661	663.198746	648.224220

Table 4.8: Optimal values of (S^*, μ_v^*, μ_b^*) with corresponding minimum expected cost TC^* for several combinations of $(\lambda, \theta, \zeta, p)$ via Quasi-Newton method.

$(\lambda, \theta, \zeta, p)$	$(4.0, 2.5, 1.0, 0.5)$	$(4.0, 2.5, 1.0, 0.3)$	$(4.0, 2.5, 0.8, 0.4)$	$(4.0, 2.5, 1.0, 0.4)$	$(4.0, 2.5, 1.3, 0.4)$
S^*	3	3	3	3	4
(μ_v^0, μ_b^0)	[4.0, 8.0]	[4.0, 8.0]	[4.0, 8.0]	[4.0, 8.0]	[4.0, 8.0]
Total Iteration	10	8	8	7	9
μ_v^*	2.565124	2.874571	2.853014	2.716693	2.434955
μ_b^*	6.266219	6.256481	6.258797	6.261617	6.271687
$TC(S^*, \mu_v^*, \mu_b^*)$	652.861803	659.340484	658.914493	656.009030	651.542521

method & DS method has been executed to justify the convergence of the obtained results. The numerical simulation between them infers that the CS algorithm has superior search characteristics and achieve the optimal cost value within a reasonable time.

Furthermore, the results in this chapter would be useful to system designers and queueing or supply chain managers in practice. The state-of-the-art of the studied chapter is its modeling, methodology, and analysis that would be highly suitable for just-in-time (JIT) services. The present study can also be extended for finite population, balking behavior of customers, unreliable servers, N-policy, etc.