Finite source accessible batch service delay queue with single departure and cost analysis using genetic algorithm

R. Kalyanaraman [1] and P. Geetha [2]
Department of Mathematics, Annamalai University, Annamalai Nagar, India
Department of Mathematics, Immaculate College For Women, Cuddalore, India

Abstract

A single server finite source model is analysed in this paper. The inter arrival time and the service times are two different exponential distributions. The service rates are state dependent. The services are given in accessible batches of minimum size 1 and maximum size M. The customers arrive from a source of size N. If the server gives service of maximum size $M(< N)$, the arrivals stay in a queue. After completion of a service of a batch, whatever may be the size of the batch, the customer departs singly. This model is analysed both in the case of time dependent and time independent domain. Some performance measures are derived. Numerical illustrations are provided to show the practical applicability of the model. Cost analysis is carried out using genetic algorithm.

Keywords: Finite Source Queue-Single Departure-Batch Service-Accessible Batch-Transient Analysis-Steady State Analysis-Performance Measures-Cost structure-Genetic algorithm.
AMS 2000 Subject Classifications Number: 60K25, 60K30 and 90B22.

1 Introduction

The Queuing literature contains many works on single server bulk service queues. Bailey(1954)(also Downton 1955)) considered that customers are served in batches of not more than b. If, immediately after the completion of a service, the server finds more than b customers waiting for service, he takes a batch of b customers for service while the others wait on the other hand, if he finds $r\,(0 \leq \mathrm{r} \leq \mathrm{b})$ customers, he takes all the r customers as a batch for service. Bloemena (1960), Jaiswal (1961), Neuts (1967) considered the same rule with the restriction that$(1 \leq \mathrm{r} \leq \mathrm{b})$, and if r=0, the service facility stops until a customer arrives. This rule is called the usual bulk service rule. While Bailey's rule is its modified type, is called, bulk service rule with intermittently available service. Jaiswal(1961) points out the distribution of the queue length for the modified rule can be obtained from that of the usual rule. The rule with a fixed batch size has been considered by Fabens(1961), Tackacs(1962) and others. In their case, the server waits until the queue size become K, and then the server serves all K customers as a batch. If there are more than K customers are waiting at a service completion point, the server takes first arrived K customers (if first in first out queue discipline is followed) for service while others will wait. Bhat(1963) considered the rule that the number taken in a batch is a random variable Y.

Neuts(1967) considered the general bulk service rule. Some more notable works are by Medhi and Borthakur(1972), Medhi(1975, 1979), Chaudhry and Templeton(1983), Chaudhry etal(1984), Briere and Chaudhry (1988) and Chaudhry and Gupta(1992). Markovian systems with accessible batches for service have been studied by Sivasamy(1990).

In this study the genetic algorithm(GA) has been successfully applied to solve an optimization cost analysis that will contribute to the solution of the queueing model defined in this paper. The performance of GA algorithm depends on the genetic operators(such as selection, crossover and mutation) used. The performance of GA using different genetic operators including intuitive recombination process for crossover and interchanging genes for mutation, are used. Genetic algorithm(GA) is an important optimization method in evolutionary computation science(Venkataraman, 2009). A large portion of researchers concurs that GA is a useful direct search process for optimum discovery of solutions. This search method found a baasis in the natural evolution process, primarily in the darwinian rule of the survival of the fittest(Sivanandam and Deepa, 2007). GA is the most suitable technique for analysing the discrete optimisation problem of cost analysis of queues. GA is considered and important method in evolutionary computation(Venkataraman, 2009). Several authors(Milton 2009;Agrawal 1999) have explored the genetic operators and their applicability into the algorithm improvement.

A quick view of how GA operates is depicted as follow. Initially, the population is generated randomly(using the' GeneratePopulation' function). All the members of the population are tested, with the help of a fitness function. A reduction of the population is undertaken with a preference for keeping individuals with higher levels of fitness, letting the rest 'die'(be erased). Those results represent the main criteria that GA uses to guide the search. However, the use of this simple but powerful operational concepts allows GA to create intuitively generations of 'better' individuals(using the 'Select' function). For instance, the key to searching for the shortest distance relies on the fitness value associated with each individual within the population. This optimisation strategy found bases in concepts of the natural evolution process, primarily the Darwinian rule of the survival of the fittest(Poli, 2000). With the poliferation of artificial intelligence, Nature-Inspired Algorithm(NIAs) are gradually getting prominence in the current era. This is because of their learning and adaptation capability from the nature. To address complex problems, scientists study how nature behaves in various contexts. The NIAs are based on physics, biology, and ethology concepts. They employ stochastic components, which include random variables.

Many real-world applications require the optimization of specific goals such as cost reduction, energy consumption reduction, performance, efficiency, and sustainability maximisation. Genetic Algorithm(GA) is one of the various NIAs available. GA is a method of optimization that employs search to solve problems with a large solution space. It implements a natural selection process with the purpose of generating superior solutions. GA maintains a chromosomal population. A comprehensive solution to the problem is represented by a chromosome. Chromosomes provide solutions to problems in the search space, which are rated using a fitness function. Malik et al.(2021) have used genetic algorithm for a Geo/G/I retrial model cost inspection. Jain and Jain(2022) have considered a server based retrial queueing system with breakdown and optimized the cost function using GA. Chahal and Kumar(2024) analysed a queueing models of machining systems with multiple working vacation and generalized triadic policy and carried out optimization using genetic algorithm. Jain and Raychaudhuri(2022) considered customers Intolerance Markovian Model with Working Vacation and Multiple Working Breakdowns and carried out cost optimization using genetic algorithm. Kalyanaraman and Anurathi(2024) has abalysed a heterogeneous two servers queue with restricted admissibility of customers and with hybrid service discipling and carried out cost optimization using genetic algorithm.

Krishnamoorthy and Ushakumari(2000) studied a queuing system with accessible batches for service. But, after completion of service the customers departs individually. As a modification of the above work, in this study we consider a finite source queue of source size N and the maximum number of customers accomadated in the service station is M. This model is analyzed in time independent and time dependent domain. The format of the paper is, the model definition, the analysis, performance measures, the waiting time analysis and numerical study are presented in sections 2, 3, 4, 5 and 6 respectively. Section 6 provides cost and profit analysis. The final section ends with a conclusion.

2 The Model

The model considered in this article is a single server finite source model. The services are given in batches of minimum size 1 and maximum size M. In addition the accessible batch service policy is used. The model has the following statistical characteristics:

1. The arrivals are from a source of size N.
2. The arrival to the queueing system occurs according to a Poisson Process with rate λ.
3. Service times are exponentially distributed, whose parameter depends on the number of customers undergoing service.
4. The maximum number of customers accommodated in the service station is M.

5. At the time of arrival, if the number of customers in the service station is less than M, then the arrival joins the service station until the batch size reaches M, this policy is called batch service with accessible batches.
6. At the time of arrival, if the number of customer in the service station is full, then the arrival joins the queue and waits for service, in the queue literature this model is called delay model.
7. After completion of service the departure takes place singly.

The Schematic representation of the model is given in the figure 2.1

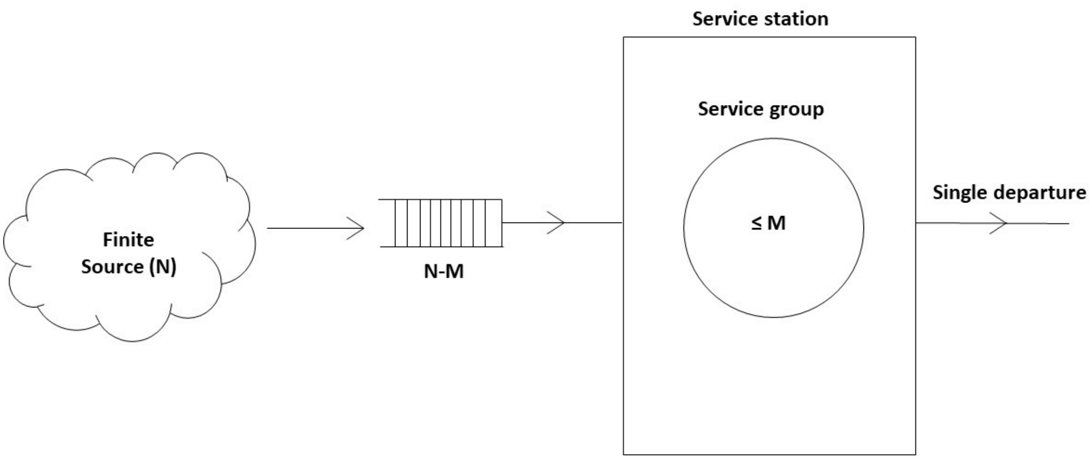

Figure 2.1: The Queueing System

3 The Analysis

For the analysis, the following notations have been introduced:

Let, $X(t)=$ Number in the Queue at time t, $X(t) \in \{0,1,2,...,N-M\}$ and $Y(t)=$ Number in the service station at time t, $Y(t) \in \{0,1,2,...,M\}$. The stochastic process $\{(X(t),Y(t)) : t \geq 0\}$ is a Markov Process with state space $S = \{0,1,2,...,N-1\} \times \{0,1,2,...,M\}$. Let $p(n,m,t) = Pr\{X(t)=n, Y(t)=m\}$ be the probability distributions and $p(n,m) = \lim_{t\to\infty} p(n,m;t)$ state probability distribution.

3.1 The Transient Analysis

The forward Kolmogorov equations of the process are obtained as

$$p'(0,0;t) = -N\lambda p(0,0;t) + \mu_1 p(0,1;t) \tag{1}$$

$$p'(0,n;t) = -[(N-n)\lambda + n\mu_n]p(0,n;t) + (N-n+1)\lambda p(0,n-1;t) \\ +(n+1)\mu_{n+1}p(0,n+1;t); \qquad 1 \le n \le M-1 \tag{2}$$

$$p'(0,M;t) = -[(N-M)\lambda + M\mu_M]p(0,M;t) + (N-M+1)\lambda p(0,M-1;t) \\ +M\mu_M p(1,M;t) \tag{3}$$

$$p'(n,M;t) = -[(N-M-n)\lambda + M\mu_M]p(n,M;t) + (N-M-n+1)\lambda \\ p(n-1,M;t) + M\mu_M p(n+1,M;t); \qquad 1 \le n \le N-M-1 \tag{4}$$

$$p'(N-M,M;t) = -M\mu_M p(N-M,M;t) + \lambda p(N-M-1,M;t) \tag{5}$$

The corresponding matrix form for equations (1) to (5) is

$$p'(t) = Ap(t) \tag{6}$$

where,

$$A = \begin{bmatrix} a_0 & \mu_1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ b_0 & a_1 & 2\mu_2 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & b_1 & a_2 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & b_2 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{M-1} & M\mu_M & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{M-1} & a_M & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & M\mu_M & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & a_{M+(N-M-1)} & M\mu_M \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & b_{M+(N-M-1)} & a_{M+(N-M)} \end{bmatrix}$$

where,
$a_0 = -N\lambda$
$a_1 = -[(N-1)\lambda + \mu_1]$
$a_2 = -[(N-2)\lambda + 2\mu_2]$
$a_{M-1} = -[(N-M+1)\lambda + (M-1)\mu_{M-1})]$
$a_M = -[(N-M)\lambda + M\mu_M]$
$a_{M+(N-M-1)} = -(\lambda + M\mu_M)$
$a_{M+(N-M)} = -M\mu_M$
$b_0 = N\lambda$

$b_1 = (N-1)\lambda$
$b_2 = (N-2)\lambda$
$b_{M-1} = (N-M+1)\lambda$
$b_{M+(N-M-1)} = \lambda$

$$p(t) = (p(0,0;t), p(0,1;t), ...p(0,M-1;t), p(0,M;t), p(1,M;t) \\ , ...p(N-M-1,M;t), p(N-M,M;t))^T \tag{7}$$

Integrating the equation (6) and $p'(t)$ is $\frac{d}{dt}p(t)$ get,

$$\frac{p'(t)}{p(t)} = A \tag{8}$$

$$\log p(t) = At + c \tag{9}$$

$$p(t) = e^{At}.e^c \tag{10}$$

$$p(t) = e^{At}.C \tag{11}$$

If $t = 0$,

$$p(0) = e^0.C \tag{12}$$

$$C = p(0) \tag{13}$$

Therefore,

$$p(t) = e^{At}.p(0) \tag{14}$$

where $p(0)$ is the initial probability vector.

For finding matrix exponential, Python provides sophisticated method powered by the SciPy library, we use the coding in Python, and find the values of e^{At} for various values of t and fixing the parameters $N = 20, M = 15, \mu_i(i = 1, 2, ..., 15) = 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4$. The probability vector $p(t)$ is obtained using $p(t) = e^{At} . p(0)$, by taking the initial probability vector $p(0) = [1, 0, ..., 0]'$

3.2 The Steady State Analysis

In steady state, the following steady state equations are obtained from (1) to (5),

$$N\lambda p(0,0) = \mu_1 p(0,1) \tag{15}$$

$$\begin{aligned}[(N-n)\lambda + n\mu_n]p(0,n) = (N-n+1)\lambda p(0,n-1)\\ +(n+1)\mu_{n+1}p(0,n+1); 1 \leq n \leq M-1\end{aligned} \tag{16}$$

$$[(N-M)\lambda + M\mu_M]p(0,M) = (N-M+1)\lambda p(0,M-1) + M\mu_M p(1,M) \tag{17}$$

$$\begin{aligned}[(N-M-n)\lambda + M\mu_M]p(n,M) = (N-M-n+1)\lambda p(n-1,M)\\ +M\mu_M p(n+1,M); \quad 1 \leq n \leq N-M-1\end{aligned} \tag{18}$$

$$M\mu_M p(N-M,M) = \lambda p(N-M-1,M) \tag{19}$$

and the normalization condition is,

$$p(0,0) + \sum_{m=1}^{M} p(0,m) + \sum_{m=1}^{N-M} p(m,M) = 1 \tag{20}$$

From (15) and (16) ,

$$p(0,M-1) = \frac{N(N-1)(N-2)...(N-(M-2))\lambda^{M-1}}{(M-1)!\mu_1\mu_2...\mu_{M-1}}p(0,0) \tag{21}$$

$$p(0,M) = \frac{N(N-1)(N-2)...(N-(M-1))\lambda^{M}}{M!\mu_1\mu_2...\mu_{M}}p(0,0) \tag{22}$$

From (17) ,

$$p(1,M) = \frac{N(N-1)(N-2)...(N-M)\lambda^{M+1}}{M!(M\mu_M)\mu_1\mu_2...\mu_{M}}p(0,0) \tag{23}$$

From (18) and (19) ,

$$p(N-M,M) = \frac{G}{M!(M\mu_M)^{N-M}\mu_1\mu_2...\mu_{M}}p(0,0) \tag{24}$$

where,

$$G = N(N-1)(N-2)...(N-(M+(N-M-1)))\lambda^{M+(N-M)}$$

$$p(0,0) = \left\{1 + \sum_{m=1}^{M} \frac{N(N-1)...(N-(m-1)))\lambda^m}{m!\mu_1\mu_2...\mu_m} + \sum_{m=1}^{N-M} \frac{N(N-1)...(N-(M+(m-1))\lambda^{M+m}}{M!(M\mu_M)^m\mu_1\mu_2...\mu_M}\right\}^{-1} \tag{25}$$

$$p(0,j) = \binom{N}{j}\prod_{i=1}^{j}\rho_i p(0,0), \qquad j = 1,2,...,M \tag{26}$$

$$p(j,M) = \binom{N}{M+j}\prod_{i=1}^{M}\rho_i(\rho_{M+1})^j p(0,0), \qquad j = 1,2,...,N-M \tag{27}$$

where,

$$\rho_i = \frac{\lambda}{\mu_i},\ \rho_{M+1} = \frac{\lambda}{M\mu_M}, \qquad i = 1,2,...,M$$

Therefore,

$$p(0,0) = \left\{1 + \sum_{j=1}^{M}\binom{N}{j}\prod_{i=1}^{j}\rho_i + \sum_{j=1}^{N-M}\binom{N}{M+j}\prod_{i=1}^{M}\rho_i(\rho_{M+1})^j\right\}^{-1} \tag{28}$$

3.3 SOME PERFORMANCE MEASURES:

In this section some performance measures like mean number of customers in the queue, in the system, in the service station and in the source, the idle probability are derived both in the case of time dependent domain(transient case) and time independent domain(stationary case) using statistical formulas.

3.3.1 Transient Case

1. Mean number of customers in the queue at time t

$$L_1(t) = \sum_{n=0}^{N-M} np(n,M;t) \tag{29}$$

2. Mean number of customers in the system at time t

$$L_2(t) = \sum_{n=0}^{M} np(0,n;t) + \sum_{n=1}^{N-M} np(n,M;t) \tag{30}$$

3. Mean number of customers in the service station at time t

$$L_3(t) = \sum_{n=0}^{M} np(0,n;t) + M \sum_{n=1}^{N-M} p(n,M;t) \tag{31}$$

4. Mean number of customers in the source at time t

$$L_4(t) = \sum_{n=0}^{M} (N-n)p(0,n;t) + \sum_{n=0}^{N-M} (N-M-n)p(n,M;t) \tag{32}$$

5. Idle Probability at time t

$$p(0,0) = \left\{ 1 + \sum_{j=1}^{M} \binom{N}{j} \prod_{i=1}^{j} \rho_i + \sum_{j=1}^{N-M} \binom{N}{M+j} \prod_{i=1}^{M} \rho_i (\rho_{M+1})^j \right\}^{-1} \tag{33}$$

3.3.2 Stationary Case

1. Mean number of customers in the queue

$$L_1 = \sum_{n=0}^{N-M} np(n,M) = \sum_{n=1}^{N-M} n\binom{N}{M+n} \prod_{i=1}^{M} \rho_i (\rho_{M+1})^n p(0,0) \tag{34}$$

2. Mean number of customers in the system

$$\begin{aligned} L_2 &= \sum_{n=0}^{M} np(0,n) + \sum_{n=1}^{N-M} np(n,M) \\ &= \sum_{n=1}^{M} n\binom{N}{n} \prod_{i=1}^{n} \rho_i p(0,0) + \sum_{n=1}^{N-M} n\binom{N}{M+n} \prod_{i=1}^{M} \rho_i (\rho_{M+1})^n p(0,0) \end{aligned} \tag{35}$$

3. Mean number of customers in the service station

$$\begin{aligned} L_3 &= \sum_{n=0}^{M} np(0,n) + M \sum_{n=1}^{N-M} p(n,M) \\ &= \sum_{n=1}^{M} n\binom{N}{n} \prod_{i=1}^{n} \rho_i p(0,0) + M \sum_{n=1}^{N-M} \binom{N}{M+n} \prod_{i=1}^{M} \rho_i (\rho_{M+1})^n p(0,0) \end{aligned} \tag{36}$$

4. Mean number of customers in the source

$$L_4 = \sum_{n=0}^{M} (N-n)p(0,n) + \sum_{n=0}^{N-M} (N-M-n)p(n,M)$$

$$= \sum_{n=0}^{M} (N-n)\binom{N}{n} \prod_{i=1}^{n} \rho_i p(0,0) + \sum_{n=0}^{N-M} (N-M-n)\binom{N}{M+n} \prod_{i=1}^{M} \rho_i (\rho_{M+1})^n p(0,0) \quad (37)$$

5. Idle Probability

$$p(0,0) = \left\{ 1 + \sum_{j=1}^{M} \binom{N}{j} \prod_{i=1}^{j} \rho_i + \sum_{j=1}^{N-M} \binom{N}{M+j} \prod_{i=1}^{M} \rho_i (\rho_{M+1})^j \right\}^{-1} \quad (38)$$

3.4 Particular Case:

If $M = 1$, the model coincides with the model $M|M|1||N$.

4 Waiting Time Analysis

Let W represents the time spent by an arriving customer(Test Customer) in the queue and $W(t)$ be its Cummulative Distributive Functions. There are two cases (i)If the Test Customer finds no one in the system, its waiting time is the service time in the system. In this case $W = 0$. (ii)If the Test Customer finds the service station is full then the waiting time in the queue $W > 0$. Using simple probabilistic arguments the distribution of W is obtained as

(i) If $W = 0$,

$$W(0) = \Pr\{W = 0\} \quad (39)$$

$$W(0) = \Pr\{M-1 \text{ (or) less number of customers in the service station}\}$$

$$W(0) = \sum_{n=0}^{M-1} p(0,n) \quad (40)$$

$$W(0) = \sum_{n=0}^{M-1} \binom{N}{n} \prod_{i=1}^{n} \rho_i p(0,0) \quad (41)$$

(ii) If $W > 0$,

$$W(t) = \Pr\{0 < W \leq t\} \quad (42)$$

$$W(t) = \sum_{n=M}^{N-M} Pr\{(n-M+1)\,\text{service completions} \leq t\,/\text{the customer finds } n \text{ in the system}\} \times \; p(n-M,M) \tag{43}$$

$$W(t) = \sum_{n=M}^{N-M} p(n-M,M) \int_0^t e^{-M\mu_M x}.M\mu_M x.\frac{(M\mu_M)^{n-M}}{(n-M)!}dx \tag{44}$$

By using integration by parts method we get,

$$\int_0^t e^{-M\mu_M x}.M\mu_M x.\frac{(M\mu_M)^{n-M}}{(n-M)!}dx = 1 - \sum_{i=0}^{n-M}(M\mu_M t)^i.\frac{e^{-M\mu_M t}}{i!} \tag{45}$$

The Cummulative distribution function for waiting time Random variable W is,

$$W(t) = \sum_{n=M}^{N-M} p(n-M,M)\left\{1 - \sum_{i=0}^{n-M}(M\mu_M t)^i.\frac{e^{-M\mu_M t}}{i!}\right\} \tag{46}$$

Now differentiating $W(t)$ with respect to t we get,

$$\frac{d}{dt}(W(t)) = \sum_{n=M}^{N-M} p(n-M,M)\left\{ -\sum_{i=0}^{n-M}\frac{(M\mu_M)^i}{i!}(t^i.(-M\mu_M)e^-M\mu_M t + e^-M\mu_M t.it^{i-1})\right\} \tag{47}$$

$$\frac{d}{dt}(W(t)) = \sum_{n=M}^{N-M} p(n-M,M)\times \left\{\sum_{i=0}^{n-M}\frac{(M\mu_M)^{i+1}.e^{-M\mu_M t}.t^i}{i!} - \sum_{i=1}^{n-M}\frac{(M\mu_M)^i.e^{-M\mu_M t}.t^{i-1}}{(i-1)!}\right\} \tag{48}$$

4.1 Expected Mean Waiting Time

$$E(W) = \int_0^\infty t dW(t) \tag{49}$$

Consider,

$$\int_0^\infty t dW(t) = \sum_{n=M}^{N-M} p(n-M,M)\sum_{i=0}^{n-M}\frac{(M\mu_M)^{i+1}}{i!}\int_0^\infty e^{-M\mu_M t}.t^{i+1}dt - \sum_{n=M}^{N-M} p(n-M,M)\sum_{i=1}^{n-M}\frac{(M\mu_M)^i}{(i-1)!}\int_0^\infty e^{-M\mu_M t}.t^i dt \tag{50}$$

Since,

$$\int_0^\infty e^{-M\mu_M t}.t^{i+1}dt = \frac{(i+1)!}{(M\mu_M)^{i+2}} \text{ and } \int_0^\infty e^{-M\mu_M t}.t^{i}dt = \frac{i!}{(M\mu_M)^{i+1}}$$

Substituting above values in equation (42) we get,

$$E(W) = \sum_{n=M}^{N-M} p(n-M,M) \sum_{i=0}^{n-M} \frac{(M\mu_M)^{i+1}}{i!} \times \frac{(i+1)!}{(M\mu_M)^{i+2}} - \sum_{n=M}^{N-M} p(n-M,M) \sum_{i=1}^{n-M} \frac{(M\mu_M)^{i}}{(i-1)!} \times \frac{i!}{(M\mu_M)^{i+1}} \tag{51}$$

$$E(W) = \sum_{n=M}^{N-M} p(n-M,M) \left\{ \sum_{i=0}^{n-M} \frac{(i+1)}{M\mu_M} - \sum_{i=1}^{n-M} \frac{i}{M\mu_M} \right\} \tag{52}$$

$$E(W) = \sum_{n=M}^{N-M} p(n-M,M) \sum_{i=0}^{n-M} \frac{1}{M\mu_M} \tag{53}$$

$$E(W) = \sum_{n=M}^{N-M} p(n-M,M) \frac{(n-M+1)}{M\mu_M} \tag{54}$$

5 The Numerical Study

In this section, we presents some numerical illustrations to show the effect of the parameters on the model, both transient case and steady state case in this section. By taking particular values to the parameters, λ, μ_i, M and N, the probabilities and performance measures are calculated and are presented in the following subsections.

5.1 Transient Case

For finding matrix exponential, Python provides sophisticated method powered by the SciPy library. We use the coding in the Python, and we find the value of e^{At} for various values of t and fixing the parameters $N = 20, M = 15, \lambda = 5, \mu_i(i = 1, 2, ...15) = 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4$ .The probability is obtained using $p(t) = e^{At}$, by taking the initial probability vector $p(t)$ $P(0) = [1, 0, 0, ...0]'$. The corresponding performance measures are calculated using the formulas in the subsection 3.3.1. The transient probabilities of various values of t are presented in table 5.1 and 5.2 and the performance measures are presented in table 5.3. The first row of table 5.1 and 5.2 show the idle probability.

Table 5.1: The Transient State Probabilities

M=15,N=20,λ=5, μ_i(i=1, 2, ... 15)=1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4					
$p_{(i,j;t)}$	t=0.1	t=0.2	t=0.3	t=0.4	t=0.5
$p_{(0,0;t)}$	8.09636×10^{-05}	3.18357×10^{-08}	1.30422×10^{-10}	6.50613×10^{-12}	1.67108×10^{-12}
$p_{(0,1;t)}$	1.01094×10^{-03}	1.02241×10^{-06}	7.60888×10^{-09}	5.20380×10^{-10}	1.52667×10^{-10}
$p_{(0,2;t)}$	5.93189×10^{-03}	1.50810×10^{-05}	1.97522×10^{-07}	1.81563×10^{-08}	6.04230×10^{-09}
$p_{(0,3;t)}$	2.17513×10^{-02}	1.36038×10^{-04}	3.04864×10^{-06}	3.70167×10^{-07}	1.38870×10^{-07}
$p_{(0,4;t)}$	5.59062×10^{-02}	8.42714×10^{-04}	3.15081×10^{-05}	4.97661×10^{-06}	2.09272×10^{-06}
$p_{(0,5;t)}$	1.07076×10^{-01}	3.81511×10^{-03}	2.32624×10^{-04}	4.71458×10^{-05}	2.21066×10^{-05}
$p_{(0,6;t)}$	1.58584×10^{-01}	1.31107×10^{-02}	1.27702×10^{-03}	3.28032×10^{-04}	1.70689×10^{-04}
$p_{(0,7;t)}$	1.85997×10^{-01}	3.50533×10^{-02}	5.35240×10^{-03}	1.72324×10^{-03}	9.90644×10^{-04}
$p_{(0,8;t)}$	1.75475×10^{-01}	7.41041×10^{-02}	1.74384×10^{-02}	6.96538×10^{-03}	4.40556×10^{-03}
$p_{(0,9;t)}$	1.34491×10^{-01}	1.25092×10^{-01}	4.46966×10^{-02}	2.19412×10^{-02}	1.52100×10^{-02}
$p_{(0,10;t)}$	8.42079×10^{-02}	1.69197×10^{-01}	9.07929×10^{-02}	5.42974×10^{-02}	4.11058×10^{-02}
$p_{(0,11;t)}$	4.31516×10^{-02}	1.82129×10^{-01}	1.46661×10^{-01}	1.05975×10^{-01}	8.73276×10^{-02}
$p_{(0,12;t)}$	1.80679×10^{-02}	1.49256×10^{-01}	1.88306×10^{-01}	1.63128×10^{-01}	1.45885×10^{-01}
$p_{(0,13;t)}$	6.14827×10^{-03}	8.05254×10^{-02}	1.91251×10^{-01}	1.97177×10^{-01}	1.90902×10^{-01}
$p_{(0,14;t)}$	1.68365×10^{-03}	3.22318×10^{-02}	1.52005×10^{-01}	1.85312×10^{-01}	1.93947×10^{-01}
$p_{(0,15;t)}$	3.64210×10^{-04}	9.59482×10^{-03}	9.25503×10^{-02}	1.32977×10^{-01}	1.50556×10^{-01}
$p_{(1,15;t)}$	6.24785×10^{-05}	2.18183×10^{-03}	4.57395×10^{-02}	7.82387×10^{-02}	9.65210×10^{-02}
$p_{(2,15;t)}$	8.17870×10^{-06}	3.67206×10^{-04}	1.76557×10^{-02}	3.62939×10^{-02}	4.91064×10^{-02}
$p_{(3,15;t)}$	7.67518×10^{-07}	4.31900×10^{-05}	4.99909×10^{-03}	1.24601×10^{-02}	1.86010×10^{-02}
$p_{(4,15;t)}$	4.59835×10^{-08}	3.17389×10^{-06}	9.24136×10^{-04}	2.81676×10^{-03}	4.66559×10^{-03}
$p_{(5,15;t)}$	1.32135×10^{-09}	1.09817×10^{-07}	8.37407×10^{-05}	3.14703×10^{-04}	5.81447×10^{-04}
Total Probability	0.9999993	0.999999	0.999999	0.999999	0.999999

Table 5.2: The Transient State Probabilities

M=15,N=20,λ=5, μ_i(i=1, 2, ... 15)=1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4					
$p_{(i,j;t)}$	t=0.6	t=0.7	t=0.8	t=0.9	t=1
$p_{(0,0;t)}$	9.34367×10^{-13}	7.27120×10^{-13}	6.50513×10^{-13}	6.18307×10^{-13}	6.03889×10^{-13}
$p_{(0,1;t)}$	8.99865×10^{-11}	7.15356×10^{-11}	6.45763×10^{-11}	6.16224×10^{-11}	6.02936×10^{-11}
$p_{(0,2;t)}$	3.74635×10^{-09}	3.04023×10^{-09}	2.76856×10^{-09}	2.65216×10^{-09}	2.59955×10^{-09}
$p_{(0,3;t)}$	9.03894×10^{-08}	7.48311×10^{-08}	6.87279×10^{-08}	6.60879×10^{-08}	6.48893×10^{-08}
$p_{(0,4;t)}$	1.42728×10^{-06}	1.20469×10^{-06}	1.11568×10^{-06}	1.07682×10^{-06}	1.05909×10^{-06}
$p_{(0,5;t)}$	1.57707×10^{-05}	1.35636×10^{-05}	1.26640×10^{-05}	1.22675×10^{-05}	1.20859×10^{-05}
$p_{(0,6;t)}$	1.27164×10^{-04}	1.11382×10^{-04}	1.04826×10^{-04}	1.01910×10^{-04}	1.00567×10^{-04}
$p_{(0,7;t)}$	7.69561×10^{-04}	6.86143×10^{-04}	6.50822×10^{-04}	6.34962×10^{-04}	6.27626×10^{-04}
$p_{(0,8;t)}$	3.56360×10^{-03}	3.23291×10^{-03}	3.09013×10^{-03}	3.02541×10^{-03}	2.99532×10^{-03}
$p_{(0,9;t)}$	1.27945×10^{-02}	1.18058×10^{-02}	1.13703×10^{-02}	1.11709×10^{-02}	1.10777×10^{-02}
$p_{(0,10;t)}$	3.59173×10^{-02}	3.36985×10^{-02}	3.26998×10^{-02}	3.22375×10^{-02}	3.20204×10^{-02}
$p_{(0,11;t)}$	7.91823×10^{-02}	7.55210×10^{-02}	7.38322×10^{-02}	7.30410×10^{-02}	7.26672×10^{-02}
$p_{(0,12;t)}$	1.37160×10^{-01}	1.32969×10^{-01}	1.30972×10^{-01}	1.30022×10^{-01}	1.29570×10^{-01}
$p_{(0,13;t)}$	1.86026×10^{-01}	1.83315×10^{-01}	1.81940×10^{-01}	1.81266×10^{-01}	1.80940×10^{-01}
$p_{(0,14;t)}$	1.95906×10^{-01}	1.96305×10^{-01}	1.96368×10^{-01}	1.96367×10^{-01}	1.96360×10^{-01}
$p_{(0,15;t)}$	1.57850×10^{-01}	1.60989×10^{-01}	1.62397×10^{-01}	1.63046×10^{-01}	1.63351×10^{-01}
$p_{(1,15;t)}$	1.05494×10^{-01}	1.09766×10^{-01}	1.11792×10^{-01}	1.12754×10^{-01}	1.13212×10^{-01}
$p_{(2,15;t)}$	5.61673×10^{-02}	5.97467×10^{-02}	6.15011×10^{-02}	6.23484×10^{-02}	6.27549×10^{-02}
$p_{(3,15;t)}$	2.23433×10^{-02}	2.43442×10^{-02}	2.53519×10^{-02}	2.58454×10^{-02}	2.60838×10^{-02}
$p_{(4,15;t)}$	5.90473×10^{-03}	6.60115×10^{-03}	6.96099×10^{-03}	7.13948×10^{-03}	7.22626×10^{-03}
$p_{(5,15;t)}$	7.77692×10^{-04}	8.93524×10^{-04}	9.54890×10^{-04}	9.85715×10^{-04}	1.00080×10^{-03}
Total Probability	0.99999	0.999999	0.999999	0.999999	0.99999

Table 5.3: The system performance measures

$M = 15, N = 20, \mu_i(i = 1, 2, ...15) = 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4$					
t	0.1	0.2	0.3	0.4	0.5
$L_1(t)$	8.13290×10^{-05}	3.05910×10^{-03}	1.00163×10^{-01}	2.01047×10^{-01}	2.72106×10^{-01}
$L_2(t)$	7.34744	9.19710	11.47183	11.28709	11.02992
$L_3(t)$	7.34843	9.23297	12.41269	13.03790	13.2995
$L_4(t)$	12.65147	8.31796	7.48714	6.76107	6.42795
$E(W(t))$	1.62658×10^{-05}	6.11820×10^{-04}	2.00326×10^{-02}	4.02094×10^{-02}	5.44212×10^{-02}
t	0.6	0.7	0.8	0.9	1.0
$L_1(t)$	3.12366×10^{-01}	3.33164×10^{-01}	3.43468×10^{-01}	3.48474×10^{-01}	3.50882×10^{-01}
$L_2(t)$	10.86713	10.78037	10.73692	10.71571	10.70551
$L_3(t)$	13.41507	13.46747	13.49186	13.50334	13.50880
$L_4(t)$	6.27258	6.19936	6.16467	6.14817	6.14034
$E(W(t))$	6.24732×10^{-02}	6.66328×10^{-02}	6.86936×10^{-02}	6.96948×10^{-02}	7.01764×10^{-02}

In the figure 5.4, for varying values of t, the mean length $L_1(t)$, $L_2(t)$, $L_3(t)$, $L_4(t)$ are drawn as graphs. In the figure 5.5, the graph of expected waiting time using Little's law are drawn.

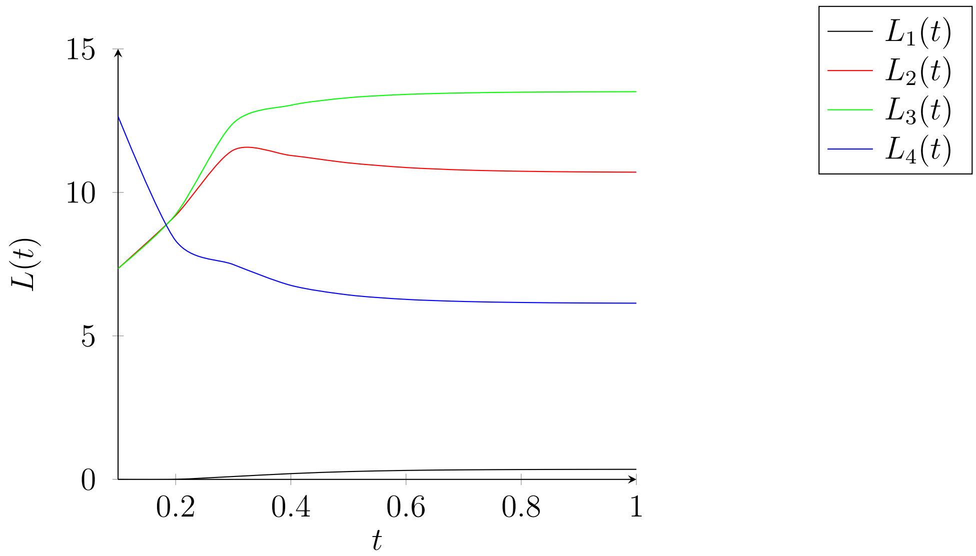

Figure: 5.4

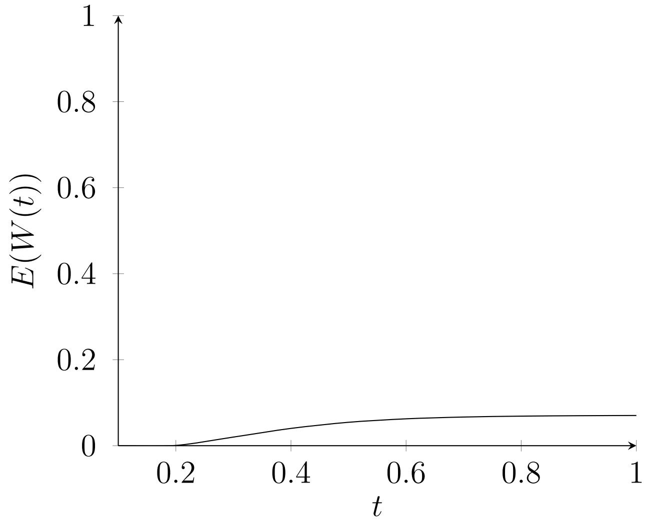

Figure: 5.5

5.2 Steady State Case

we calculated the stationary probabilities and the performance measures obtained in subsection 3.3.2 For the analysis, we vary the arrival rate λ from 1 to 10. The steady state probabilites are presented in tables 5.6 and 5.7. The corresponding system performance measures are presented in tables 5.8. In the figure 5.9, the system performance measures L_1 , L_2 , L_3 and L_4 are shown as graphs and the mean waiting time is shown in the graph 5.10.

Table 5.6: The Steady State Probabilities

$M = 15, N = 20, \mu_i(i = 1, 2, ...15) = 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4$					
p_i	λ=1	λ=2	λ=3	λ=4	λ=5
$p_{0,0}$	1.51213×10^{-05}	3.49466×10^{-08}	4.12803×10^{-10}	1.17846×10^{-11}	5.91284×10^{-13}
$p_{0,1}$	3.02420×10^{-04}	1.39787×10^{-06}	2.47682×10^{-08}	9.42772×10^{-10}	5.91284×10^{-11}
$p_{0,2}$	2.61181×10^{-03}	2.41449×10^{-05}	6.41721×10^{-07}	3.25685×10^{-08}	2.55327×10^{-09}
$p_{0,3}$	1.30590×10^{-02}	2.41449×10^{-04}	9.625682×10^{-06}	6.51369×10^{-07}	6.38318×10^{-08}
$p_{0,4}$	4.26930×10^{-02}	1.57871×10^{-03}	9.44071×10^{-05}	8.51791×10^{-06}	1.04340×10^{-06}
$p_{0,5}$	9.75864×10^{-02}	7.21695×10^{-03}	6.47363×10^{-04}	7.78780×10^{-05}	1.19246×10^{-05}
$p_{0,6}$	1.62640×10^{-01}	2.40565×10^{-02}	3.23681×10^{-03}	5.19187×10^{-04}	9.93719×10^{-05}
$p_{0,7}$	2.03300×10^{-01}	6.01413×10^{-02}	1.21381×10^{-02}	2.59593×10^{-03}	6.21074×10^{-04}
$p_{0,8}$	1.94331×10^{-01}	1.14976×10^{-01}	3.48077×10^{-02}	9.92563×10^{-03}	2.96837×10^{-03}
$p_{0,9}$	1.43949×10^{-01}	1.70335×10^{-01}	7.73503×10^{-02}	2.94093×10^{-02}	1.09940×10^{-02}
$p_{0,10}$	8.33338×10^{-02}	1.97230×10^{-01}	1.34345×10^{-01}	6.81057×10^{-02}	3.18246×10^{-02}
$p_{0,11}$	3.78813×10^{-02}	1.79300×10^{-01}	1.83198×10^{-01}	1.23829×10^{-01}	7.23287×10^{-02}
$p_{0,12}$	1.35290×10^{-02}	1.28071×10^{-01}	1.96284×10^{-01}	1.76898×10^{-01}	1.29158×10^{-01}
$p_{0,13}$	3.78434×10^{-03}	7.16482×10^{-02}	1.64714×10^{-01}	1.97928×10^{-01}	1.80641×10^{-01}
$p_{0,14}$	8.22683×10^{-04}	3.11514×10^{-02}	1.107422×10^{-01}	1.72111×10^{-01}	1.96349×10^{-01}
$p_{0,15}$	1.37114×10^{-04}	1.03838×10^{-02}	5.37110×10^{-02}	1.14741×10^{-01}	1.63624×10^{-01}
$p_{1,15}$	1.19044×10^{-05}	2.88439×10^{-02}	2.23796×10^{-02}	6.37448×10^{-02}	1.13628×10^{-01}
$p_{2,15}$	2.11596×10^{-06}	6.40975×10^{-04}	7.45986×10^{-03}	2.83310×10^{-02}	6.31266×10^{-02}
$p_{3,15}$	1.76330×10^{-07}	1.06829×10^{-04}	1.86496×10^{-03}	9.44368×10^{-03}	2.63027×10^{-02}
$p_{4,15}$	9.79609×10^{-09}	1.18699×10^{-05}	3.10827×10^{-04}	2.09860×10^{-03}	7.30632×10^{-03}
$p_{5,15}$	2.72114×10^{-10}	6.59440×10^{-07}	2.59023×10^{-05}	2.33177×10^{-04}	1.01477×10^{-03}
Total Probability	0.999985	1	1	1	1

Table 5.7: The Steady State Probabilities

p_i	λ=6	λ=7	λ=8	λ=9	λ=10
$p_{0,0}$	4.41037×10^{-14}	4.42908×10^{-15}	5.61753×10^{-16}	8.61009×10^{-17}	1.54442×10^{-17}
$p_{0,1}$	5.29244×10^{-12}	6.20071×10^{-13}	8.98804×10^{-14}	1.54982×10^{-14}	3.08883×10^{-15}
$p_{0,2}$	2.74245×10^{-10}	3.74861×10^{-11}	6.20992×10^{-12}	1.20463×10^{-12}	2.66763×10^{-13}
$p_{0,3}$	8.22735×10^{-09}	1.31201×10^{-09}	2.48397×10^{-10}	5.42083×10^{-11}	1.33381×10^{-11}
$p_{0,4}$	1.61383×10^{-07}	3.00249×10^{-08}	6.49653×10^{-09}	1.59498×10^{-09}	4.36055×10^{-10}
$p_{0,5}$	2.21325×10^{-06}	4.80399×10^{-07}	1.18794×10^{-07}	3.28109×10^{-08}	9.96696×10^{-09}
$p_{0,6}$	2.21325×10^{-05}	5.60465×10^{-06}	1.58392×10^{-06}	4.92164×10^{-07}	1.66116×10^{-07}
$p_{0,7}$	1.65993×10^{-03}	4.90407×10^{-05}	1.58392×10^{-05}	5.53685×10^{-06}	2.07645×10^{-06}
$p_{0,8}$	9.52021×10^{-03}	3.28140×10^{-04}	1.21123×10^{-04}	4.76332×10^{-05}	1.98484×10^{-05}
$p_{0,9}$	4.23121×10^{-02}	1.70147×10^{-03}	7.17766×10^{-04}	3.17554×10^{-04}	1.47025×10^{-04}
$p_{0,10}$	1.46979×10^{-02}	6.89542×10^{-02}	3.32439×10^{-03}	1.65463×10^{-03}	8.51199×10^{-04}
$p_{0,11}$	4.00851×10^{-02}	2.19400×10^{-02}	1.20887×10^{-02}	6.76892×10^{-03}	3.86909×10^{-03}
$p_{0,12}$	8.58967×10^{-02}	5.48499×10^{-02}	3.45391×10^{-02}	2.17573×10^{-02}	1.38182×10^{-02}
$p_{0,13}$	1.44162×10^{-01}	1.07398×10^{-01}	7.72903×10^{-02}	5.47735×10^{-02}	3.86522×10^{-02}
$p_{0,14}$	1.88038×10^{-01}	1.63432×10^{-01}	1.34418×10^{-01}	1.07166×10^{-01}	8.40266×10^{-02}
$p_{0,15}$	1.88038×10^{-01}	1.90671×10^{-01}	1.79224×10^{-01}	1.60748×10^{-01}	1.40044×10^{-01}
$p_{1,15}$	1.56698×10^{-01}	1.85375×10^{-01}	1.99138×10^{-01}	2.00935×10^{-01}	1.94506×10^{-01}
$p_{2.15}$	1.04465×10^{-01}	1.44180×10^{-01}	1.77011×10^{-01}	2.00935×10^{-01}	2.16118×10^{-01}
$p_{3,15}$	5.22327×10^{-02}	8.41052×10^{-02}	1.18007×10^{-01}	1.50701×10^{-01}	1.80098×10^{-01}
$p_{4,15}$	1.74109×10^{-02}	3.27076×10^{-02}	5.24478×10^{-02}	7.53507×10^{-02}	1.00055×10^{-01}
$p_{5,15}$	2.90182×10^{-03}	6.35981×10^{-03}	1.16551×10^{-02}	1.88377×10^{-02}	2.77929×10^{-02}
Total Probability	1	1	1	1	1

Table 5.8: The system performance measures

$M = 15, N = 20, \mu_i(i = 1, 2, ...15) = 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4$					
λ	1	2	3	4	5
L_1	2.38450×10^{-05}	4.53760×10^{-03}	4.42670×10^{-02}	1.58298×10^{-01}	3.53088×10^{-01}
L_2	7.42775	10.0706	11.2918	11.3938	10.6961
L_3	7.42804	10.1207	11.7282	12.7932	13.5137
L_4	12.5719	9.87472	8.22758	7.04846	6.13319
$E(W)$	3.87558×10^{-06}	2.99689×10^{-04}	1.62165×10^{-03}	3.79619×10^{-03}	6.44964×10^{-03}
λ	6	7	8	9	10
L_1	6.06480×10^{-01}	8.88681×10^{-01}	1.17525	1.4505	1.70622
L_2	9.59936	8.4198	7.33847	6.42947	5.70412
L_3	13.9985	14.322	14.5371	14.6804	14.7764
L_4	5.39500	4.78929	4.28764	3.86912	3.51734
$E(W)$	9.00045×10^{-03}	1.14117×10^{-02}	1.25904×10^{-02}	1.42621×10^{-02}	1.57460×10^{-02}

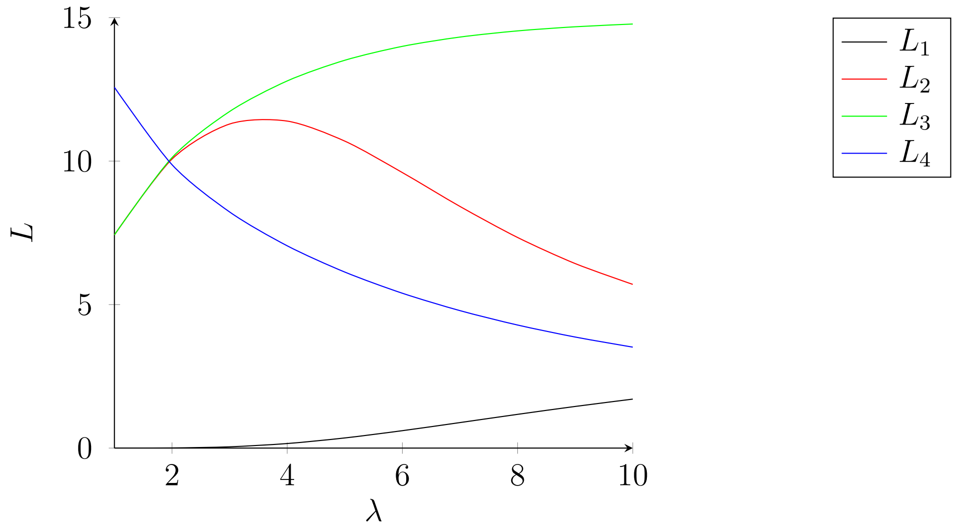

Figure: 5.9

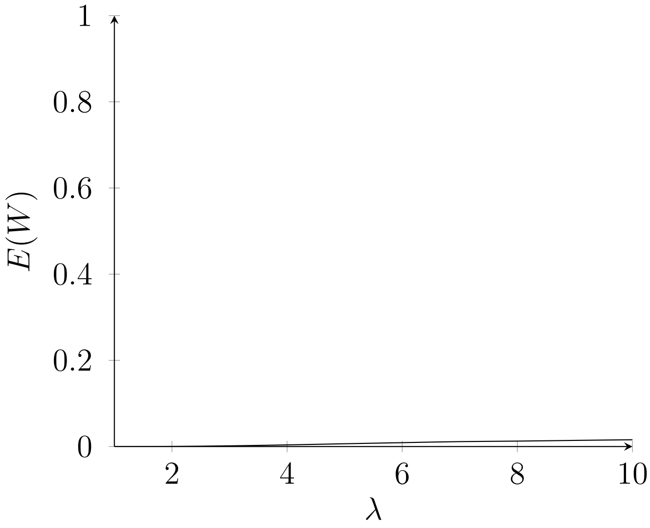

Figure: 5.10

6 Cost and Profit analysis

Cost analysis in a queueing model involves determining the costs associated with operating a queueing system and balancing them to achieve optimal performance. The goal is to minimize total costs which typically include service costs and waiting costs. For example, In call centers, minimizing operational costs while ensuring acceptable waiting times. In health care, balancing staff costs with patient waiting times. In manufacturing, reducing down time costs while maintaining efficiency.

Profit analysis in queue management involves examining how queueing model impact profitability in a bussiness setting. This includes evaluating the trade offs between between costs benefits. Effective queue management balances costs and benefits to optimize profitability. Implementing targeted strategies such as reducing waiting time, using technology, etc..,

Revenue analysis in a queue system involves assessing how revenue is generated and affacted by customer arrivals, service rates, waiting times and over all system performance. This type of analysis is particularly relevent in industries like retail, telecommunications, or service-oriented business where queues are a significant operational aspect.

We define suitable cost, revenue and profit structure to the model discussed in this article. To define the structure of the cost and profit related elements

- C_s -The rate at which service is provided per unit time
- C_h -The holding cost per customer per unit time
- C_i -Cost per unit time for idle period
- C_b -cost incurred during period of service activity

- C_w -Waiting cost of a customer per unit time
- R_e -Cost related to revenue

are introduced.
Total expected cost per unit time of the system,
$T_c = (\mu_1 C_{s_1} + \mu_2 C_{s_2} + ... + \mu_M C_{s_M}) + C_h L_2 + C_i p(0,0) + C_b(1 - p(0,0)) + C_w E(W)$
Total expected revenue of the system,
$T_r = (\mu_1 R_{e_1} + \mu_2 R_{e_2} + ... + \mu_M R_{e_M})(1 - p(0,0))$
Total expected profit of the system,
$T_p = T_r - T_c$.

The total cost function is minimized using genetic algorithm. In this analysis, a population of our choice of chromosomes with six decision factors, which are genes, has been considered. In the given population each individual is referred as chromosomes and each chromosome contains six genes, which are randomly initialized. The expected total cost function is the fitness function. R-program is used to calculate the value. The chromosome with lowest total cost value is the best fittest. 1000 cycles are used for the generation. The relevant values are presented in table 6.1.

The general steps of the GA implemented to our model is

```
Input: Fitness function, Decision variables
Output: Best fitness(optimal)value, Best(optimal)solution
Step 1: Initialize population size
Step 2: Generate initial solution
Step 3: Evaluate the fitness value
Step 4: Select parameters based on fitness
Step 5: If
        criteria satisfied then get the optimal solution
        else
        cross over
        mutation
        generate next generation
        goto Step 3.
```

Table 6.1: Expected total cost, Total expected revenue and Total expected profit

S.No.	Arrival rate	μ_M	Mean number of customers in the system	Mean waiting time in the queue	Expected total cost	Total expected revenue	Total expected profit
1	1.621497	10.00024	3.591204	0.09136619	324.7059	27043.13	26718.4241
2	2.901962	0.1321958	15.29464	0.3253712	367.7299	399.4087	31.6788
3	3.199791	5.002808	16.16119	3.108343	412.925	13550.06	13137.135
4	3.403904	10.10006	16.16119	3.108343	451.1544	27312.64	26861.4856
5	4.144202	10.03059	7.752687	0.174476	366.5484	27125.07	26758.5216
6	4.645974	5.061178	16.16119	3.108343	413.3628	13707.66	13294.2972
7	5.127787	10.00093	1.609141	0.05206374	304.8905	27044.99	26740.0995
8	7.279139	10.02532	1.609141	0.05206374	305.0734	27110.84	26805.7666
9	12.75144	10.01591	28.97339	1.19863	578.6452	27085.44	26506.7948
10	12.8668	5.001275	17.67703	0.3730363	428.072	13545.92	13545.92
11	14.42059	15.0218	11.87284	0.1715143	445.184	40601.34	40156.156
12	14.51486	10.0135	1.609738	0.0521755	304.9907	27078.93	26773.9393
13	14.6837	5.021765	17.47488	0.3689985	426.2041	27078.93	26652.7259
14	15.23037	15.0092	21.31344	0.4462054	539.4954	40567.32	40027.8246
15	17.85752	10.00142	68.55063	136.9258	974.309	27046.31	26072.001

The above table demonstrate the expected total cost, using GA and are given in sixth column. Using R code the total expected revenue has been calculated and are given in the seventh column. The eighth column presents the total expected profit. Columns two, three, four and five shows the optimum arrival rate, service rate(μ_M), mean number of customers in the system and mean waiting time in the queue.

7 Conclusion

In this paper, we have analysed a single server finite source model, in addition a services are given in batches of size r($1 \leq r \leq N$)with accessible batch service policy. If the server is busy with a batch of M customers, the customer wait in a queue. Eventhough the services are given in batches, after completion of service the customers departure singly. The model analysed both in time independent domain and time dependent domain. In the case of time dependent domain the formula for probability vector p(t) is given. In the case of time dependent domain the probabilities are obtained using recurrence concept. Both the cases performance measures like mean number of customers, idle probability and mean waiting time are calculated. Waiting time Distribution is also derived. To show the practical applicability of the model we present some numerical illustrations. Cost and profit analysis is also carried out.

References

[1] Agarwal, S., and Deb, K., ’Understanding Interactions among genetic algorithms parameters’, Foundations of Genetic Algorithms, vol. 5, 1999.

[2] Bailey, N.T.J., on queueing processes with bulk service, Journal of Royal Statistical Society, Series B, Vol 16, 80-87,1954.

[3] Bhat, U.N., Imbedded Markov Chain analysis of bulk queues, Journal of the Australian Mathematical Society, Vol 4, 224-263, 1963.

[4] Bloemena, A.R. On a queuing process with a certain type of bulk service, Bull. Inst. Statist., 37, 219-227, 1960.

[5] Briere, G., and Chaudhry, M.L., Computional analysis of single server bulk-arrival queues $M^X|G|1$, Computers and Operations Research, Vol 15(3), 283-292, 1988.

[6] Chahal, P.K., and Kumar, K., A genetic algorithm for cost optimization in queueing models of machining systems with multiple working vacation and generatized triadic policy, Decision Analytics Journal 10 100395, 1-14, 2024.

[7] Chaudhry, M.L., Easton, G.D., and Posner, M.J.M., Some Correction results for the queue $GI^X|M|1$, European Journal of Operational Research, Vol 18, 131-132, 1984.

[8] Chaudhry, M.L., and Gupta, U.C., Exact Computational analysis of waiting time distributions of single server bulk arrival queues: $M^X|G|1$, European Journal of Operational Research, Vol 63(3), 445-462, 1992.

[9] Chaudhry, M.L., and Templeton J.G.C., A first course in bulk queues, John Wiley and Sons, New York, 1983.

[10] Downton, F., waiting time in bulk service queues, Journal of Royal Statistical Society, Series B, Vol 17, 256-261, 1955.

[11] Fabens, A.T., The Solution of queueing and inventory models by Semi-Markov Processes, Journal of Royal Statistical Society, Series B, Vol 23, 113-127, 1961.

[12] Jain, A., and Raychaudhuri, C., Cost Optimization using Genetic Algorithm in Customers Intolerance Markovian Model with Working Vacation and Multiple Working Breakdowns, International Journal of Mathematical, Engineering and Management Sciences Vol. 7, 656-669, 2022.

[13] Jain, M., and Jain, A., Genetic algorithm in retrial queueing system with server breakdown and caller intolerance with voluntary service. International Journal System Assurance Engineering and Management, 13, 582-598, 2022.

[14] Jaiswal, N.K., Pre-emptive resume priority queue, Operations Research, Vol 9, 732-742, 1961.

[15] Kalyanaraman, R., and Anurathi, B., Cost optimization using genetic algorithm of a heterogeneous two servers queue with restricted admissibility of customers and with hybrid service discipline, Journal of Systems Engineering and Electronics, Volume 34 ISSUE 9, 273-300, 2024.

[16] Krishnamoorthy, A. and Ushakumari P.V., A queueing system with single arrival bulk service and single departure, Mathematical and Computer Modelling, 31, 99-108, 2000.

[17] Malik, G., Upadhyaya, S., and Sharma, R., Cost inspection of a Geo/G/1 retrial model using particle swarm optimization and genetic algorithm, Ain Shams Eng.J.12(2)(2021)2241-2254, 2021.

[18] Medhi, J. and Borthakur A., On a two server Markovian queue witha a general bulk service rule, Cashiers due Centre d'etudes de recherche operationalle, Vol 14, 151-158, 1972.

[19] Medhi J., Further results on waiting time distribution in a Poisson queue under a general bulk service rule, Cashiers due Centre d' etudes de Recherche Operationalle, Vol 21(1), 183-189, 1979.

[20] Medhi, J., Waiting time distribution in a Poission queue with a general bulk service rule, Management sciences, Vol 21(7), 777-782, 1975.

[21] Milton, J., 'Analysis and improvement of genetic algorithms using concepts from information theory', University of Technology Sydney, 2009.

[22] Neuts, M.F., A General Class of Bulk Queues with Poission Input, The Analysis of Mathematical statistics, Vol 38(3), 759-770, 1967.

[23] Poli, R., 'Exact Schema Theorem and Effective Fitness for GP with One-Point Crossover', GECCO, pp.469-76, 2000.

[24] Sivanandam, S., and Deepa, S., Introduction to genetic algorithms, Springer Science and Business Media, 2007.

[25] Sivasamy, R., A bulk service queue with accessible and non-accessible batches, Operations Research 27(1), 46-54, 1990.

[26] Tackacs, L., Introduction to the theory of queues, Oxford University Press, New York, 1962.

[27] Venkataraman, P., 2009, Applied optimization with MATLAB programming, John Wiley and sons.