Cost analysis of heterogeneous two server queue with breakdown.

R. Kalyanaraman
Department of Mathematics, Annamalai University, Annamalainagar-608002, India

Abstract

In this paper the cost analysis of a two heterogeneous server Markovian queue with breakdown has been given. In the queueing model the second server has a threshold for service and if the system breakdown, immediately repair has been carried out.A cost structure is defined and using genetic algorithm the optimum cost has been generated numerically for various values of the parameters.
Keywords: Heterogeneous server - breakdown - slow server - threshold - steady state - cost analysis- genetic algorithm.
2000 Mathematics Subject Classification: 90B22, 60K25 and 60K30.

1 Introduction

Multi server Markoving queueing models are studied by many authors. In the multi server queueing models thr servers are either homogeneous or heterogeneous. The homogeneous case is valid only when the service process is mechanically or electrically controlled. In 1981, Neuts and Takahashi have pointed out that for the queueing system with two heterogeneous servers, analytical results are analytically intractable. Even though, some researchers focused their studies on queue with two heterogeneous servers. The equilibrium analysis for the general input and exponential service time and with n severs was given in Kendall (1953). A non-constructive existence theorem for the stationary distribution of general input and general service time was presented in Kiefer and Wolfowitz (1955). Karlin and Mc.Gregor (1958) obtained the busy period distribution for the $M/M/S$ queue. Krishnamoorthi (1963) consider a Poisson queue with two heterogeneous servers and with vilotion of the First-in-First-out principle.

Heffer(1969) has analyzed waiting time distribution of $M/E_k/S$ queue. A Markovian queuing system with balking and two heterogeneous servers has been considered in Singh (1970). The author determines the capacity of the slower server and obtains the optimal service rates. Singh (1973) discussed a Markovian queue with the number of servers depending upon the queue length. Desmit (1983a, b) presented an approach to identify the distribution of waiting times and queue lengths for the queue $GI/H_2/S$. He reduced the problem to the solution of the Wiener-Hopf-type equations and then used a factorization method to solve the system.

Lin and Kumar (1984) has analyzed the optimal control of a queueing system with two heterogeneous servers. Rubinovitch (1985a, b) studied the problem of a heterogeneous two channels queueing systems. In his first paper he discussed three simple models and gave the condition when to discard to slower server depending on the expected number of customer in the system. In the second paper he studied a queueing model with a stalling concept. In 1999, Abou-Elu-ata and Shawky introduced a simpler approach to find the condition when to discard the slower server in a heterogeneous two channels queue.

Barcelo (2003) has obtained an approximation for the mean waiting time of $M/H_{2b}/S$ queue. Shin and Moon (2009) has carried out an approximate analysis for $M/G/C$ queue. Arkat and Farahani (2014), has used a partial-fraction decomposition approach to the $M/H_2/2$ queue. Zhernovyi (2011) analyzed queue with switching of service modes and threshold blocking of input flow. Kopytko and Zhernovyi (2011) investigated Markovian queue with switching of service mode.

Kalyanaraman and senthilkumar(2018a) analyzed a two heterogeneous server markovian queue with switching service models. In the same year the authors discussed heterogeneous server Markovian queue with restricted admissibility and with reneging. Kalyanaraman and senthilkumar(2018b) analyzed a two heterogeneous server queue with restricted adminissibility. In real life, Models of queueing system with different intensity of service are used for the study of telecommunication process.

In some queueing situations, it can be seen that systems can suddenly break down such as computer system, communication system, and many other. For example, consider a machine that always needs some maintenance otherwise it will be break down and a repair man must be called for repair. this type of queues are called queue with unreliable server. (Avi-Itzhak and Naor(1963), Gaver(1962), White and Christie(1958), Thiruvengadam(1963), Federgrune and Green(1986) and Van Dijk(1988)).

Genetic algorithm (GA) is an important optimization method in evolutionary computational process (Venkataraman, 2009). Several authors including Milton et.al. (2005) and Agrawal (1999) have explored the genetic operators and their applicability into the algorithm improvement. In this study, in addition to the steady state analysis of the models defined and discussed in this thesis, the genetic algorithm (GA) has been successfully applied to solve an optimization problem relate to cost structure of the queueing models. The performance of GA algorithm depends on the genetic operator?s selection, crossover and mutation. How the GA operates is described as follow. Initially, the population is generated randomly. 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 be eleminated. Those results represent the main criteria that GA uses to guide the search. However, the use of these simple but powerful operational concepts allows GA to create intuitively generations of better individuals using the Select function. This optimisation strategy found bases in concepts of the natural evolution process, primarily the Darwinian rule of the survival of the fittest (Poli, 2000). The genetic algorithms are designed to simulate a biological process, much of the relevant terminology is borrowed from biology. However, the entities that this terminology refers to in genetic algorithms are much simpler than their biological counterparts (Mitchell(1995)). The basic components common to almost all genetic algorithms are:

- a fitness function for optimization
- a population of chromosomes
- selection of which chromosomes will reproduce
- crossover to produce next generation of chromosomes
- random mutation of chromosomes in new generation

The fitness function is the function that the algorithm is trying to optimize. The word fitness is taken from evolutionary theory. It is used here because the fitness function tests and quantities how to fit each potential solution is. The fitness function is one of the most pivotal parts of the algorithm. The term chromosome refers to a numerical value or values that represent a candidate solution to the problem that the genetic algorithm is trying to solve (Mitchell, (1995)). Each candidate solution is encoded as an array of parameter values, a process that is also found in other optimization algorithms (Haupt, and Haupt, (1998)) .If a problem has N parameters, then typically each chromosome is encoded as an array of N element. That is, the chromosome = $(p_1, p_2, ..., p_N)$ where each pi is a particular value of the ith parameter (Haupt, and Haupt, (2004)). It is up to the creator of the genetic algorithm to devise how to translate the sample space of candidate solutions into chromosomes. One approach is to convert each parameter value into a bit string (sequence of 1s and 0s), then concatenate the parameters end-to-end like genes in a DNA strand to create the chromosomes (Mitchell, (1996)). Historically, chromosomes were typically encoded this way, and it remains a suitable method for discrete solution spaces. Modern computers allow chromosomes to include permutations, real numbers, and many other objects.

The concept of genetic algorithm begins with a randomly chosen assortment of chromosomes, which serves as the first generation (initial population). Then each chromosome in the population is evaluated by the fitness function to test how well it solves the problem at hand. Now the selection operator chooses some of the chromosomes for reproduction based on a probability distribution defined by the user. The fitter a chromosome is, the more likely it is to be selected. The selection operator chooses chromosomes with replacement, so the same chromosome can be chosen more than once. The crossover operator resembles the biological crossing over and recombination of chromosomes in cell meiosis. This operator swaps a subsequence of two of the chosen chromosomes to create two offspring. The mutation operator randomly flips individual bits in the new chromosomes (turning a 0 into a 1 and vice versa). Typically mutation happens with a very low probability, such as 0.001. Some algorithms implement the mutation operator before the selection and crossover operators; this is a matter of preference. The mutation operator helps protect against this problem by maintaining diversity in the population, but it can also make the algorithm converge more slowly. Typically the selection, crossover, and mutation process continues until the number of offspring is the same as the initial population, so that the second generation is composed entirely of new offspring and the first generation is completely replaced. Now the second generation is tested by the fitness function, and the cycle repeats. It is a common practice to record the chromosome with the highest fitness (along with its fitness value) from each generation, or the best-so-far chromosome (Koza, (1994)). Genetic algorithms are iterated until the fitness value of the best-so-far chromosome stabilizes and does not change for many generations. This means the algorithm has converges to a solution(s).

Genetic algorithms are used in a variety of applications. Some prominent examples are automatic programming and machine learning. They are also well suited to modeling phenomena in economics, ecology, the human immune system, population genetics, and social systems. In particular, genetic algorithms have been widely used in various research studies to optimize cost models associated with research problems. For instance, Lin and Ke (2010) applied the genetic algorithm approach to optimize a multi-server infinite capacity queueing system using a triadic policy. Similarly, Haung et al. (2011) investigated the coordination of arrivals and services in a finite capacity queueing system with the triadic policy, and they adopted the genetic algorithm approach to optimize the cost function for this model. The use of genetic algorithms in optimization problems has proven to be effective and efficient. By mimicking natural selection and genetic inheritance, genetic algorithms can find optimal solutions to complex problems. As such, they have become a popular tool in various fields, including engineering, computer science, and economics. Jain and Jain (2022) developed a cost model for a retrial queueing system with an unreliable server and voluntary service. They used the Genetic Algorithm to minimize the system?s cost. In recent studies, numerous researchers have implemented the Genetic algorithm for the purpose of optimization in their respective work. Among them are Khalili, and Khah, (2020), Malik et al. (2021), Meena et al. (2022), Liu et al. (2023), Hasani et al. (2022), Sangha and Antala (2024).

In this paper, a two server queueing system with breakdown and a threshold for second server has been considered. The model hes been completely analysed in steady state by Kalyanaraman and Kalaiselvi(2019). The model definition and the steady results are given in section 2. The corresponding cost structure is defined and the analysis is carried out using genetic algorithm in section 3. A conclusion has been given in section 4.

2 The Queueing Model

Queueing theory continues to be one of the most extensive theories of stochastic process. Queueing theory provides an efficient mathematical framework for the study of several congestion situations arising in different application areas such as telecommunications, production lines, computer networks, etc. Many server queues are widely seen in bank. In a bank customers come for service, then the bank staffs process customers' requests, and after their services are completed, customers leave the bank. This is a classical many server queueing system with arrivals, services, and departures. It can also be observed in other scenarios such as emergency departments of hospitals, call centers, post offices, railway stations, airports and computers. For these systems, servers are usually different from each other. They may possess different types of skills. Even when they have the same skill, their ability on this skill may be different. Such servers are called heterogeneous servers. In manufacturing systems, service systems, telecommunication systems and computer networks breakdown may occur at any moment. This necessitate the study of queues with breakdown. In practice, it can be seen that additional servers are provided to reduce congestion when queue length is too long, for example, in the bank, in the super markets, etc... the decision-makers often provides another server to reduce the long queue. In our model we assume that the number of servers working in the queue can be adjusted based on the number of customers is the system. With all these in mind Kalyanaraman and Kalaiselvi(2019) considered a two server queueing system with breakdown and a threshold for second server with the following assumptions :

- Arrival to the system follows Poisson process with rate λ.
- The two servers are heterogeneous, call servers as Server 1 and Server 2.
- The service provided by the servers are random periods follows negative exponential distribution with rates μ_1 (Server 1) and μ_2 (Server 2). Also $\mu_1 > \mu_2, \mu = \mu_1 + \mu_2$.
- If there are less than K customers in the system the first server works and the second server stays in ideal state. Once the system size reaches K, the second server starts work.
- The system may breakdown during service(Both the servers are busy) and the break downs are assumed to occur according to a Poisson process with rate γ.
- Immediately, the repair takes place, the duration of the repaired period follows negative exponential distribution with rate β.
- Each customer is served by only one server.
- The queue discipline is first come first served.
- The waiting line capacity is infinite.

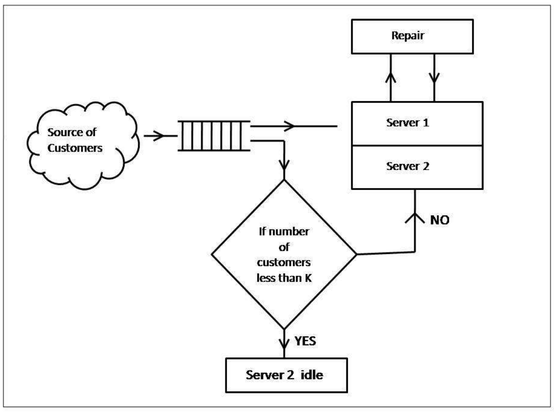

Figure 2.1: Schematic representation of the model

For the mathematical frame work of the above defined model, the following probabilities has been defined:
$p_n(t)$ be the probability that there are n customers in the system at time t, when the system is in busy state. $n \geq 0, t \geq 0$.
$q_n(t)$ be the probability that there are n customers in the system at time t, when the system is in repair state. $n \geq K$
In steady state $\lim_{t\to\infty} p_n(t) = p_n, \lim_{t\to\infty} q_n(t) = q_n$.
Using general birth death arguments the authors obtained the difference equations,using the equations they find the following probabilities: The probability of n customers in the system is

$$p_n = \begin{cases} \rho_1^n p_0, \ 1 \leq n \leq K-1 \\ \rho_1^K \rho^{n-K} p_0 B_{n-K}, \ n \geq K \end{cases} \tag{1}$$

$$q_n = \rho_1^K p_0 C_{n-K}, \ n \geq K \tag{2}$$

$$p_0 = \left[1 + \rho_1 \left[\frac{1-\rho_1^{K-1}}{1-\rho_1}\right] + \rho_1^K \sum_{n=K}^{\infty} \left(\rho^{n-K} B_{n-K} + C_{n-K}\right)\right]^{-1} \tag{3}$$

where

$$\rho_1 = \frac{\lambda}{\mu_1}; \ \rho = \frac{\lambda}{\mu} \tag{4}$$

$$A_1 = \frac{\gamma}{(\lambda+\beta)} \tag{5}$$

$$B_0 = \frac{\mu\mu_1(\lambda+\beta)(\gamma+\lambda+\mu_2)}{\gamma\beta(\mu_2-\mu_1)^2 + (\lambda+\beta)\mu_1\mu_2(2(\gamma+\lambda)+\mu)} \tag{6}$$

$$\psi_1 = \frac{(\gamma+\lambda)(\gamma+\lambda+\mu_2)}{\mu_2(2(\gamma+\lambda)+\mu)} \tag{7}$$

$$\xi_1 = \frac{\gamma\beta(\mu_1(2(\lambda+\gamma)+\mu) - (\gamma+\lambda+\mu_1)(\mu_2-\mu_1))}{(\lambda+\beta)\mu_1\mu_2^2(2(\lambda+\gamma)+\mu)} \tag{8}$$

$$B_1 = \frac{\psi_1 - B_0\xi_1}{\rho} \tag{9}$$

$$B_i = \frac{(\lambda+\gamma+\mu)B_{i-1} - \mu B_{i-2} - \beta C_{i-1}\rho^{-(i-1)}}{\lambda}, \qquad i = 2,3,4,... \tag{10}$$

$$C_0 = \frac{\gamma}{\lambda+\beta} \tag{11}$$

$$C_1 = \frac{\rho B_1 + \lambda A_1 B_0}{(\lambda+\beta)} \tag{12}$$

$$C_i = \frac{\gamma\rho^i B_i + \lambda C_{i-1}}{(\lambda+\beta)}, \qquad i = 2,3,4,... \tag{13}$$

Equation $(1), (2)$ and (3) together represents the probability distribution in steady sate of the model designed in this article and the stability condition is $\frac{\lambda}{\mu} < 1$.

Using these probabilities we have the following performance measure such as expected number of customers in the system, second moment of number of customers in the system and variance of number of customers in the system.
(i)Expected number of customers in the system

$$L = \left[\frac{\rho_1^2\left[(1-\rho_1^K) - K\rho_1^{K-1}(1-\rho_1)\right]}{(1-\rho_1)^2} + \sum_{n=K+1}^{\infty} n\rho_1^K \rho^{n-K} B_{n-K}\right] p_0 \tag{14}$$

(ii) Second moment of number of customers in the system

$$L_1 = \mu_1 \sum_{n=1}^{K-1} n^2 p_n + \sum_{n=K}^{\infty} n^2 p_n \tag{15}$$

(iii) Variance of number of customers in the system

$$L_2 = L_1 - L^2 \tag{16}$$

(iv) A customer's mean waiting time

$$W = \frac{L}{\lambda} \tag{17}$$

(v) Probability that both the servers are busy

$$P_B = 1 - p_0\left[\frac{\mu_1 + \lambda}{\mu_1}\right] \tag{18}$$

3 Cost analysis

In queueing theory, cost analysis is essential for evaluating the efficiency and financial performance of service systems, such as customer support lines, production processes, or transportation services. The cost analysis in queueing theory involves balancing service costs (e.g., staffing, resources) and waiting costs (e.g., lost productivity, customer dissatisfaction). This analysis helps in determining optimal levels of service capacity and balancing costs to improve overall efficiency. This involves analyzing how sensitive the total cost is to changes in service level, arrival rates, or service rates. Sensitivity analysis can help in making decisions when demand is variable or when costs fluctuate. A total expected cost function related to the model discussed i this paper has been formed wih K as the decision variable. The aim of developing such a function is to determine the optimal threshold K, say K*,so as to minimize the cost function. On the basis of the definition of each cost element and the corresponding performance measures.

$$T_c = C_1\mu_1 + C_2\mu_2 + C_hL + C_Ip_0 + C_BP_B + C_WW \tag{19}$$

Where C_1 - The rate at which first service is provided per unit time
C_2 - The rate at which second service is provided per unit time
C_h - Holding cost per customer per unit time
C_I - Cost per unit time for idle period
C_B - unit cost incurred during period of server activity
C_W - waiting cost of a customer per unit time

The genetic algorithm is applied to find minimum total cost. The general steps of the GA implemented to our models are 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 Step3.

The results are generated using R-programming.

Table 3.1: Expected cost ($\beta = 7, \gamma = 6, C_h = 5, C_b = 7, C_1 = 2, C_2 = 3$)

N	K	λ	μ_1	μ_2	L	M	Minimum total expected cost
500	10	6.373002	15.00298	6	11.11171	0.1612287	437.4315
	30	14.28571	15.0147		11.11171	0.1612287	437.5194
	100	7.346971	15.01247		11.11171	0.1612287	437.5027
1000	10	10.33501	15.02052	6	11.11171	0.1612287	437.563
	30	19.39297	15.0034		11.11171	0.1612287	437.4347
	100	9.710176	15.01525		11.11171	0.1612287	437.5236
10000	10	6.026744	15.00236	6	11.11171	0.1612287	437.4269
	30	11.64008	15.00032		11.11171	0.1612287	437.4116
	100	7.7867246	15.01092		11.11171	0.1612287	437.4911
500	10	14.12562	6	15.01962	11.11171	0.1612287	437.5563
	30	6.438494		15.00101	11.11171	0.1612287	437.4168
	100	17.608		15.00524	11.11171	0.1612287	437.4485
1000	10	7.426545	6	15.00551	11.11171	0.1612287	437.4505
	30	14.54522		15.0012	11.11171	0.1612287	437.4182
	100	13.37625		15.00562	11.11171	0.1612287	437.4513
10000	10	9.014066	6	15.01183	11.11171	0.1612287	437.4979
	30	18.53312		15.00775	11.11171	0.1612287	437.4673
	100	8.244244		15.00723	11.11171	0.1612287	437.4634

Fot the given values of cost coefficients, γ and β the minimum total cost function is calculted, shown in the last column of the table 3.1 using genetic algorithm. In the caculation process, N (500,1000,10000), K (10,30,100) $|mu_2 = 6$ the corresponding optimum arrival rate, optimum service rate mu_1 are shown in $3^r d$ and $4^t h$ column of the table 3.1. Also by fixing $mu_1 = 6$, the optimum service rate mu_2 is calculated and is shown in the fifth column.

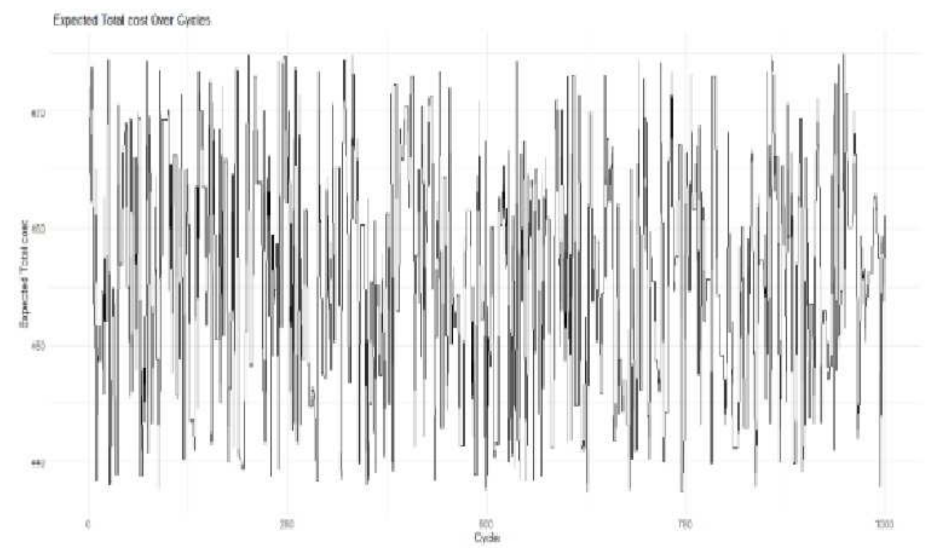

Figure 3.2: Expected total cost

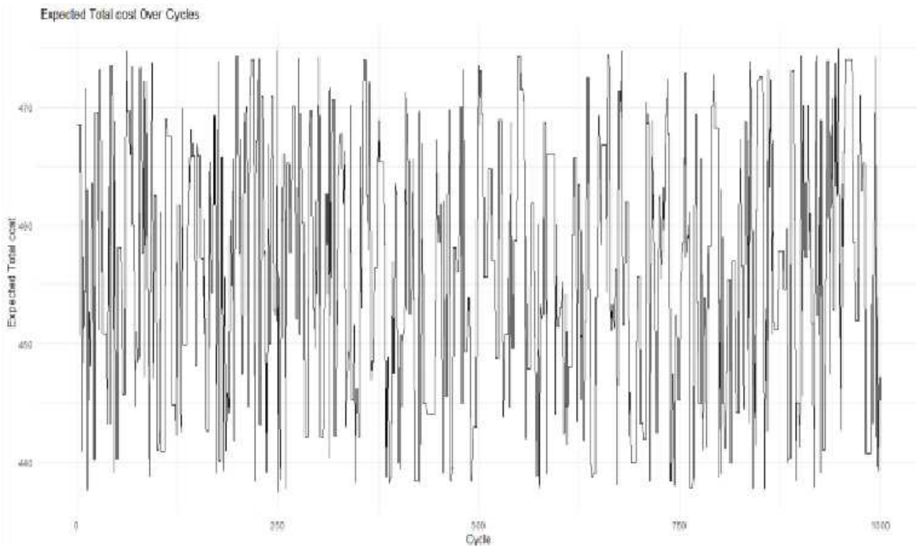

Figure 3.3: Expected Total cost

Figures 3.2 and 3.3 presents the curve of total cost againt number of cycles.

4 Conclusion

In this paper, cost analsis has been obtained for total cost of the system of the given queueing model using genetic algorithm. Numerical illustrations are provided. Our computational experience shows that, the minimum total cost is obtainable.

References

[1] **Abou-El-Ata, M.O. and Shawky, A.I.,** A simple approach for the slower server problem. *commun.Fa. Sci.Univ.* 48, 1-6, 1999.

[2] **Agrawal, S. and Deb, K.** , 'Understanding Interactions among genetic algorithms parameters', Foundations of Genetic Algorithms, vol. 5,1999.

[3] **Arkat, J. and Farahani, M.H.,** Partial fraction recomposition approach to the $M/H_2/2$ queue. *Franian, Jr. of. Operation Research* 5(1), 55-63, 2014.

[4] **Avi-Itzhak, B. and Naor, P.,** Some queueing problems with the service station subject to breakdowns. *Operation Research* 11, 303-320, 1963.

[5] **Barcelo, F.,** An engineering approximation for the mean waiting time in $M/H_{2b}/S$ queue. *In proceeding of the Int. Network Optimization Conference, Paris, France.* Oct. 19-27, 2003.

[6] **Desmit, J.H.A.,** A numerical solution for the multi-server queue with hyper exponetial servce times, *Int. Jr. of Oper. Res. Lett,* 2, 217-224, 1983a.

[7] **Desmit, J.H.A.,** The queue $GI/M/S$ with customers of differnt types or the queue $GI/HM/S$, *adv. Appl. Prob.,* 15, 392-419, 1983b.

[8] **Hasani Goodarzi,A., Diabat,E., Jabbarzadeh,A. and Paquet,M.** An M/M/c queue model for vehicle routing problem in multi-door cross-docking environments, *Comput. Oper. Res.* 138 , 105513, 2022.

[9] **Haung,H.I ,. Hsu,P.C, and Ke,J.C.** Optimal operation of an $M/M/2$ queue with removable servers, *Expert Syst. Appl.* 38 (8) 1005410059,2011.

[10] **Haupt, R. L., and Haupt, S. E.** Practical Genetic Algorithms. New York: Wiley Interscience,1998.

[11] **Haupt, R. L., and Haupt, S. E.** Practical Genetic Algorithms (2nd ed.). Hobo ken: Wiley,2004.

[12] **Heffer, J.C.,** Steady state solution of the $M/Ek/C$ (FIFO) queuing system,*CORSJ.* 7, 16-30, 1969.

[13] **Federgruen, A. and Green, L.,** Queueing system with service interruptions. *Operation Research* 34, 752-768, 1986.

[14] **Gaver, D.P.,** A waiting line with interrupted service including priorities.*Franian, Jr. R. Stat. Soc.,* ser-B 24, 73-90, 1962.

[15] **Jain,M and Jain,A.** Genetic algorithm in retrial queueing system with server breakdown and caller intolerance with voluntary service, *Int. J. Sys. Assur. Eng. Manag.* 13 , 582598, 2022.

[16] **Kalyanaraman, R., and Kalaiselvi, S.,** Heterogeneous server queue with breakdown and a threshold on slow server, *AIP conference proceedings,* 2177, 1-8, 2019.

[17] **Kalyanaraman, R., and Senthilkumar, R.,** Heterogeneous server Markovian queue with switching of service modes, *Annamalai University Science Journal,* 51(1), 1-8, 2018a.

[18] **Kalyanaraman, R., and Senthilkumar, R.,** Heterogeneous server Markovian queue with Restricted admissibility and with Reneging, *Mathematical sciences international research journal,* 7(1), 309-315, 2018b.

[19] **Karlin, S. and McGregor. J.,** Many server queuing process with Poisson input and exponential service times,*pacific J.math.* 8, 87-118, 1958.

[20] **Kendall, D.G.,** Stochastic process in the theory of queues, *Ann.math.stat.* 24, 333-354, 1953.

[21] **Khalili, S. and Khah,M.M.** A new queuing-based mathematical model for hotel capacity, *J. Appl. Res. Ind.* 7 (3) ,203220, 2020.

[22] **Kiefer, J. and Wolfowitz, J.,** On the theory of queues with many servers, *trans Amer.math.soc.* 78, 1-18, 1955.

[23] **Kopytko, B., and Zhernovyi, K.,** Stationary characteristics of $M^{\theta}/M/1$ queue with switching of service modes, *Scientific Res. of the Ins. math and comp.sci.,* 2(10), 117-128, 2011.

[24] **Koza, J. R.** Introduction to Genetic Programming. In K. E. Kinnear (Ed.), Advances in Genetic Programming (pp. 21-41). Cambridge: MIT Press,1994.

[25] **Krishnamoorthi, B.,** On Poisson queue with two heterogeneous servers, *oper.Res,* 11(3) 321-330, 1963.

[26] **Lin, C.H. and Ke,J.C.** Genetic algorithm for optimal thresholds of an infinite capacity multi-server system with triadic policy, *Expert Syst. Appl.* 37 (6) ,42764282,2010.

[27] **Lin, W. and Kumar, P.,** Optimal control of c queuing system with two heterogeneous servers, *Automatic control, IEEE trans.on.* 29(8), 696-703, 1984.

[28] **Liu,H., Niu,Z., Du, J. and Lin,X.** Genetic algorithm for delay efficient computation offloading in dispersed computing, *Ad Hoc Netw.* 142 , 17, 2023.

[29] **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) 22412254, 2021.

[30] **Meena,R.K., Jain, M., Assad, A.,Sethi, R. and Garg, D.** Performance and cost comparative analysis for M/G/1 repairable machining system with N policy vacation, *Math. Comput. Simulation* 200 315328, 2022.

[31] **Milton, J., Kennedy, P. and Mitchell, H.,** 'The effect of mutation on the accumulation of information in a genetic algorithm', *AI 2005: Advances in Artificial Intelligence, Springer,* pp. 360-8, 2005.

[32] **Mitchell, M.** Genetic Algorithms: An Overview. *Complexity*, 1(1), 31-39. 30 ,1995.

[33] **Mitchell, M.** An Introduction to Genetic Algorithms. Cambridge: MIT Press,1996.

[34] **Neuts, M.F. and Takahashi, T.,** Asymptotic behavior of the Stationary distribution in the $GI/PH/C$ queue with heterogeneous servers, *Z. Wahrscheinlickeirsth* 57, 441-452, 1981.

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

[36] **Rubinovitch, M.,** The slow server problem, *J.Appl.prob,* 22, 205-213, 1985a.

[37] **Rubinovitch, M.,** The slow server problem; A queue with stalling, *J.Appl.prob.* 22, 879-892, 1985b.

[38] **Sanga, S.S,and Antala, K.S.**Performance analysis and cost investigations for state dependent single unreliable server finite queue under F-policy using GA and QNM, *J. Comput. Appl. Math.* 441 ,115679,2024.

[39] **Shin, Y.W. and Moon, D.H.,** Sensitivity and approximation of $M/G/C$ queue: *Numerical experiments proceeding of it 8th international symposium on operations Research and its applications,* 140-147 ,2009.

[40] **Singh, V.P.,** Two servers Markovian queues with balking, *Heterogeneous Vs Homogeneous servers, oper. Res.* 18(1), 145 -159 ,1970.

[41] **Singh, V.P.,** Queue dependence-servers, *Jr of Eng.maths,* 7(2), 123-126, 1973.

[42] **Thiruvengadam, K.,** Queueing with breakdowns. *Operation Research* 11, 303-320, 1963.

[43] **Van Dijk, N.,** Simple bounds for queueing systems with breakdowns. *Performance Evaluation* 8(2), 117-128, 1988.

[44] **Venkataraman, P.** , Applied optimization with MATLAB programming, John Wiley and Sons, 2009.

[45] **White, H. and Christie, L.,** Queueing with pre-emptive priorities or with breakdown. *Operation Research* 6, 79-95, 1958.

[46] **Zhernovyi, K. Yu.,** Investigation of $M^{\theta}/G/1/m$ queue with switching of service modes and threshold blocking of input flow, *Information process,* 10(2), 159-180, 2011.