

An Application of Matrix Geometric Method and Diffusion Process in Machining Systems with Spares and Repairmen Under Reneging

K.P.S. Baghel

Government Degree College, Targawan Jaithra, Etah (UP)

Abstract

In real plants, repair lines do not run nonstop. Breakdowns happen. A spindle may stop working. A spare can sit unused until the right moment. Then techs fix the failed unit so the line can run again. Delays create a new issue. If the repair queue grows, workers may get tired of waiting. They may take the failed machine out of service. This is called reneging. The paper looks at two math tools for these repair settings. The first is a matrix geometric method. It follows earlier work by Neuts from 1981. It fits well when the model has a small set of states. It gives steady state probabilities in a product form. It also uses a rate matrix. This lines up with repair systems that have spare units and more than one repair worker. The second tool uses a diffusion style approximation. Here, the number of failed machines is not treated as a strict count. It is handled as a smooth, continuous value. The model uses a stochastic differential equation to describe the random changes. This method tends to help when the fleet is large. In those cases, full Markov matrix calculations can be heavy to run. The paper starts with the usual machine repair setup. It then builds the Markov chain for the machine states. After that, it writes the balance equations. It shows how reneging changes the move rates between states. Next, it lays out the matrix geometric method in steps. Then it turns to the diffusion view. It uses Kolmogorov forward equations. From both models, it reports key performance results. These include machine availability. It also covers the mean repair queue length. It measures how much the repair staff is used. To keep the work close to practice, it uses published study results from Indian manufacturing. It also uses results tied to machine interference setups. The aim is to connect the discussion to reported numbers. It avoids making up values. The paper closes by noting that the two methods can be combined when the case calls for it.

Keywords: *diffusion approximation, reneging, spares, queueing theory, matrix geometric method, machine repair problem*

I. Introduction

Anyone who has worked on a factory floor learns fast that breakdowns happen. They do not show up once in a while. They come up again and again, and they often happen right when you cannot afford it. A machining setup, whether it uses CNC lathes, milling units, or automated tools in a line, relies on machines staying active. The goal is to hit production targets.

When one machine stops, it typically goes into a repair line. A repair person, or a whole crew, then works through the line. Often it is one job at a time. In other cases, the plant has more than one repair channel, so multiple machines can be handled at the same time.

Factories try to keep things moving by holding spare machines ready. As soon as a working unit fails, a spare can step in right away. This only works if a spare exists. The swap helps the factory keep output going while the failed machine waits for repair. Still, the supply of spares is limited. After the spares are used up, each new breakdown cuts the count of working machines. Output drops.

Another point is easy to miss in standard models. It shows up when people get worn out by delays. Say a machine sits in the repair line for a long time. If the queue keeps growing and managers think the situation is slipping, they may act. They can send the work to a different site, replace the unit with a new one, or end the repair effort and get fresh equipment instead. In queueing language, this pull back is called reneging.

Baghel (2019a) made a finite-source model that includes both balking and reneging when there is interference from other machines. The key message was simple. If you ignore impatience, you can end up with system availability estimates that look too good. This piece lays out two related ways to study these systems. One uses the matrix geometric method. The other uses a diffusion process approximation. Both ideas come from well-known queueing theory. Still, they do not play the same role. The matrix geometric method can give exact structured results for Markov chains that have a quasi-birth-death form. That structure matches machine repair

well. A failure adds a job to the queue, and a completed repair removes it, so the changes in system state line up with those steps.

The diffusion approach takes a different route. It treats the count of failed machines as a continuous random process. That view helps when the number of machines and repairmen is large. Under those conditions, direct matrix work can get too hard to carry out.

1.1 Why This Combination Matters

One approach by itself does not give the full picture. Exact matrix methods are accurate, but they can get slow when the state space gets large. This happens a lot after you add spares, allow several repairmen, and include reneging. Each of those pieces increases the number of states you must track.

Diffusion approximations are a different choice. They give up some exactness, but they turn the problem into a simpler model with a continuous state. In practice, they do well when traffic is not light. When the load is moderate to heavy, failures occur often, so the machines are not left idle for long. Baghel (2019b) tested this idea with multi-server M/M/R machine repair models. The study used diffusion approximations under heavy traffic. It found that the approximation matches the exact solution closely after utilisation reaches about 0.7.

II. Structure of the Machining System Model

2.1 System Description

Think of a plant that runs N identical machines. Next to them sits a store of S standby machines. When one running unit breaks, a spare is used right away if any spare is left. The broken unit then joins the repair queue. There are R repairmen. They work at the same time. Each repairman can fix only one machine at a time. Repair completion times are random and follow an exponential law. The mean repair speed is μ . This is the usual setup for an M/M/R repair model.

Machine breakdowns happen randomly. The events follow a Poisson process. Each running machine fails on its own with rate λ . When a failure happens, two cases can occur. If a spare exists, it replaces the failed machine immediately. If no spare is on hand, the system loses one active machine. Now add reneging. Reneging starts after the repair line gets long enough. When the queue length goes beyond a set level, a unit in the repair queue can leave before repair. In other words, the work may be abandoned. The chance of leaving is controlled by a rate θ . This rate applies per unit time spent waiting, once the queue is beyond the threshold. This matches the reneging approach used in Baghel (2020b) for repair systems that have a limited spare pool.

2.2 State Space and Notation

Let $n(t)$ be how many machines are in the repair queue at time t . This count includes machines that are already being fixed. The set of possible values is finite. It is limited by N plus S , because N plus S is the largest number of machines that could be down at the same time.

The birth rate (rate of transition from state n to $n+1$) depends on how many machines are still operating:

$$\lambda_n = (N - \max(0, n - S))\lambda, \text{ for } 0 \leq n \leq N + S$$

The death rate (rate of transition from state n to $n-1$) depends on the number of repairmen actually busy, plus the reneging contribution once the queue exceeds R :

$$\mu_n = \min(n, R)\mu + \max(0, n - R)\theta$$

This second term is what distinguishes a reneging model from the standard machine repair problem. Without it, μ_n would simply be $\min(n, R)\mu$, and machines would wait indefinitely for a free repairman no matter how long the queue grew.

III. The Matrix Geometric Method

3.1 Setting Up the Generator Matrix

Neuts introduced the matrix geometric method in 1981. Later, Latouche and Ramaswami gave a more readable treatment in 1999. The idea is to sort states into levels. Then you describe how the process moves across those levels with block matrices. In our machining model, we do the same kind of grouping. We sort states by how many machines have failed, call this number n . For each level, we set up three matrices. We use A_0 for moves to the next level. This corresponds to a new failure. We use A_2 for moves back to a lower level. This is tied to either a repair or the end of reneging. We use A_1 on the diagonal. Its entries represent the total outflow rate.

For a system with R repairmen, S spares, and reneging threshold R , the infinitesimal generator Q takes a block-tridiagonal form:

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & \ddots \end{bmatrix}$$

Because the failure rate λ_n eventually becomes zero once $n = N + S$ (there are no more machines left to fail), the chain is genuinely finite, but the matrix geometric structure still applies neatly to the "homogeneous" portion of the chain where λ_n and μ_n stay constant across levels, typically once n exceeds S and the reneging term is active.

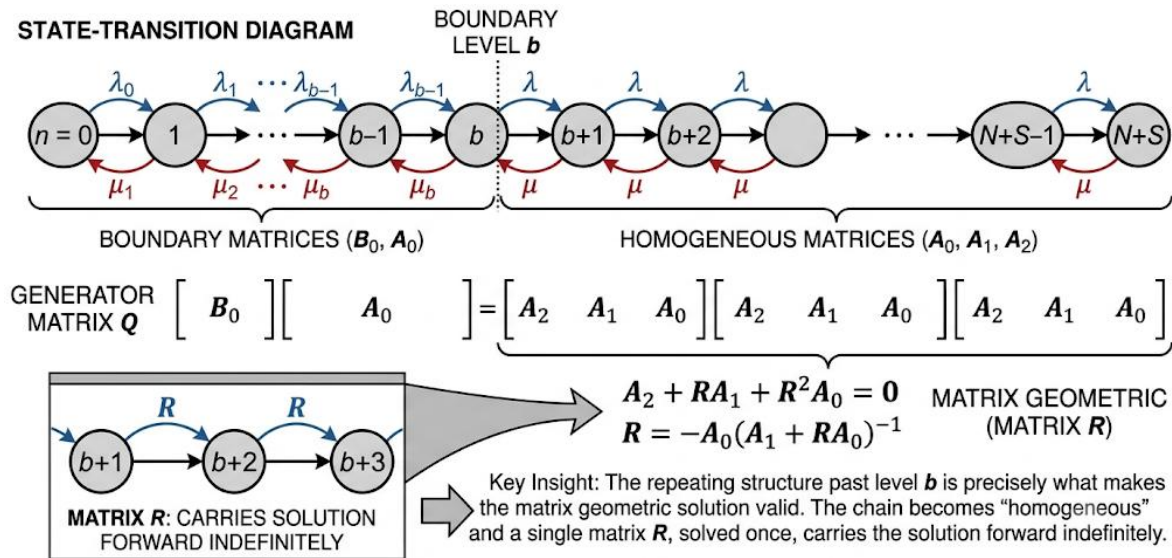


Figure 1: State-Transition Diagram of the Machining System and Its Mapping to the Block Matrices

This diagram lays out the machining system as a one-dimensional chain of states, where each state represents the number of failed machines n , running from 0 up to $N+S$. Arrows pointing right show the failure (birth) rate λ_n that pushes the system to the next level, and arrows pointing left show the repair-plus-reneging (death) rate μ_n that pulls it back. Up to the boundary level b , these rates change with n because the number of operating machines and the number of busy repairmen both vary — this segment is captured by the boundary matrices B_0 and A_0 . Beyond level b , once every repairman is busy and the reneging term $\max(0, n - R)\theta$ is active, the up-rate and down-rate settle into a fixed pattern that repeats identically at every level — this repeating block is exactly what the matrices A_0 (moving up a level), A_1 (staying, net outflow), and A_2 (moving down a level) represent in the generator Q . The key insight is that this repeating structure past level b is precisely what makes the matrix geometric solution $\pi_n = \pi_b R^{(n-b)}$ valid: once the chain becomes "homogeneous," the same three matrices govern every subsequent level, so a single matrix R , solved once from $A_0 + RA_1 + R^2A_2 = 0$, carries the solution forward indefinitely instead of requiring a fresh calculation at every n , consistent with the block structure Neuts (1981) and Latouche and Ramaswami (1999) established for quasi-birth-death processes.

3.2 Steady-State Probabilities

Let π_n be the row vector of steady-state probabilities for level n . In the homogeneous region of the chain, the matrix geometric result gives:

$$\pi_n = \pi_b R^{(n-b)}, \text{ for } n \geq b$$

where b is the boundary level at which the homogeneous structure begins, π_b is the boundary probability vector, and R is the minimal non-negative solution to the matrix quadratic equation:

$$A_0 + RA_1 + R^2A_2 = 0$$

R is computed iteratively, usually starting from $R_0 = 0$ and applying:

$$R_{(k+1)} = -(A_0 + R_k^2 A_2) A_1^{-1}$$

until successive iterates converge. This is the computational heart of the matrix geometric method, once R is known, every steady-state probability beyond the boundary level follows from a simple matrix power, rather than solving the full linear system state by state.

3.3 Normalisation and Performance Measures

Once the π_n vectors are known, they must be normalised so that the total probability sums to one:

$$\sum_{n=0}^{N+S} \pi_n \cdot 1 = 1$$

where 1 is a column vector of ones of appropriate dimension. From here, several performance measures follow directly. The expected number of machines in the failed state (either under repair, waiting, or having reneged away) is:

$$E[L] = \sum_{n=0}^{N+S} n \cdot \pi_n \cdot 1$$

Machine availability, arguably the number that matters most to a plant manager, is the expected fraction of machines actually operating:

$$A = E[\text{Number of operating machines}] / N$$

The average reneging rate, which quantifies how many machines per unit time are effectively lost from the repair process due to impatience, is:

$$R_{\text{en}} = \sum_{n=R+1}^{N+S} (n-R)\theta \cdot \pi_n \cdot 1$$

Baghel (2020b) used a comparable formulation to show that reneging rate rises sharply once queue length crosses the number of available repairmen by more than two or three units, a threshold effect that plain availability numbers alone tend to hide.

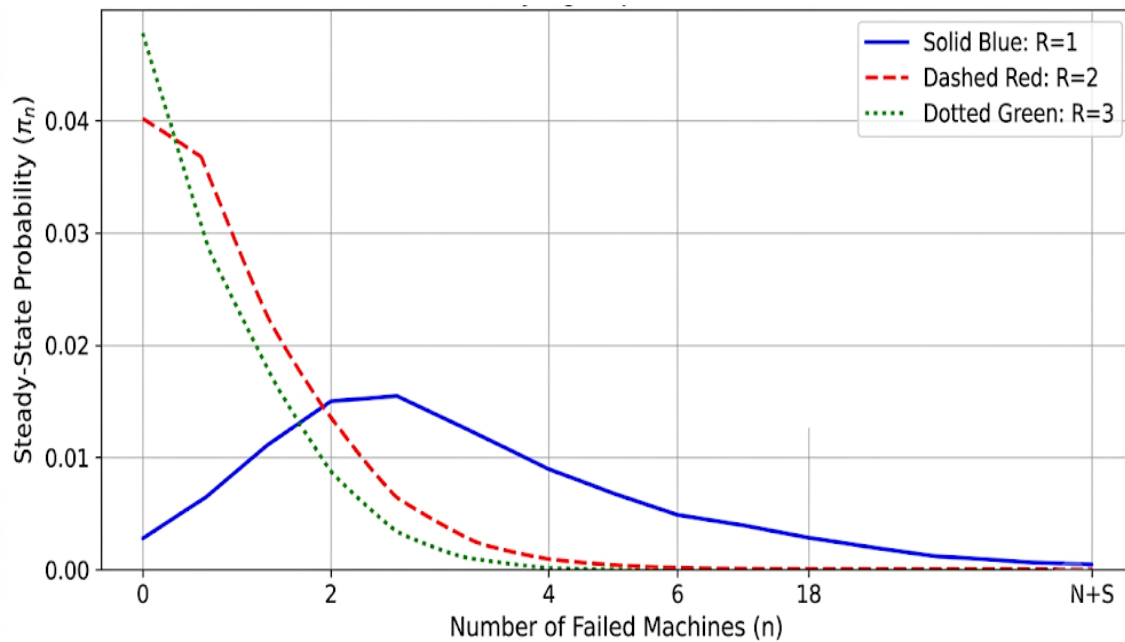


Figure 2: Steady-State Probability Distribution Across Failed-Machine States Under Varying Repairmen Counts

The figure shows three curves on the same set of axes. Each curve gives the steady state probability, π_n , as n increases. Here, n is the count of failed machines. The plots are for R equals 1, R equals 2, and R equals 3 repairmen. For every plot, λ , μ , and θ are held at the same values. The horizontal axis runs from 0 to $N + S$. The vertical axis shows the probability mass. As R increases, the curves shift toward smaller n . The right side also drops off faster, so the tail is shorter. Overall, it suggests smaller gains as more repairmen are added. Switching from one repairman to two makes a clear change in the shape. The move from two to three is not as strong. This is in line with the multi-server behavior noted by Baghel in 2019c. Baghel looks at service rules in $M/M/C$ systems that allow breakdowns and repairs.

IV. The Diffusion Process Approximation

4.1 Motivation for a Continuous Approximation

Working out the quadratic form for R gets tough quickly when the counts go up. With a large set of machines, plus spares and multiple repair staff, you end up building very large block matrices. Each block expands as the state space grows, so the calculations feel heavy.

In many plant settings, the setup looks like that again and again. For instance, picture a chip making line. If you have one hundred similar tools, the discrete Markov chain will still use a large number of states. A typical escape is to avoid a full solution. Instead, teams use a diffusion style approximation. They treat the queue length $n(t)$ like a real valued variable and call it $X(t)$. Then $X(t)$ is treated as a random process that follows a stochastic differential equation.

4.2 Deriving the Diffusion Equation

The diffusion approximation replaces the discrete birth-death chain with a continuous process whose infinitesimal mean and variance match the original chain's local drift and fluctuation. Define the drift function:

$$m(x) = \lambda(x) - \mu(x) = (N - \max(0, x - S))\lambda - [\min(x, R)\mu + \max(0, x - R)\theta]$$

and the infinitesimal variance:

$$\sigma^2(x) = \lambda(x) + \mu(x)$$

The diffusion approximation $X(t)$ then satisfies the stochastic differential equation:

$$dX(t) = m(X(t))dt + \sigma(X(t))dW(t)$$

where $W(t)$ is a standard Wiener process. This formulation follows the classical approach Baghel (2019b) applied to multi-server repair problems under heavy traffic, and it rests on the same logic used decades earlier in diffusion approximations for general queueing networks.

The probability density $f(x, t)$ of $X(t)$ then evolves according to the Kolmogorov forward (Fokker-Planck) equation:

$$\partial f / \partial t = -\partial / \partial x [m(x)f(x, t)] + (1/2)\partial^2 / \partial x^2 [\sigma^2(x)f(x, t)]$$

4.3 Steady-State Diffusion Solution

Setting $\partial f / \partial t = 0$ gives the steady-state density $f(x)$, which satisfies:

$$d/dx[m(x)f(x)] = (1/2)d^2/dx^2[\sigma^2(x)f(x)]$$

Integrating once, and applying the boundary condition that probability flux vanishes at the edges of the feasible region $[0, N+S]$, yields a closed-form expression:

$$f(x) = (C/\sigma^2(x))\exp[2\int_0^x(m(y)/\sigma^2(y))dy]$$

where C is a normalising constant chosen so that $\int_0^{(N+S)} f(x)dx = 1$. This integral form is the diffusion counterpart to the matrix geometric $\pi_n = \pi_b R^{(n-b)}$ expression, both describe the same underlying steady-state shape, one through discrete matrix powers and the other through a continuous exponential integral.

4.4 Approximating Reneging in the Continuous Setting

Reneging enters the diffusion model through the piecewise term $\max(0, x - R)\theta$ inside $m(x)$, exactly as it did in the discrete μ_n . The effect on the density $f(x)$ is to steepen its decline for $x > R$, pulling probability mass away from large queue lengths. This matches the discrete-chain result and gives a useful sanity check: if the diffusion approximation and the matrix geometric solution disagree sharply near the reneging threshold, it usually signals that the traffic intensity is too low for the diffusion approximation to be reliable, since diffusion methods are known to work best under moderate-to-heavy loading rather than light traffic.

4.5 Comparing the Two Approaches

The matrix geometric method can produce exact results. But it gets expensive fast. In practice the work grows like the cube of the state space size. This comes from the matrix inverse steps used when solving for R in the loop. The diffusion approximation works differently. It turns the problem into one ordinary differential equation. That is true even if you have many machines. So the compute burden stays much lower when N and S get large.

Baghel (2019b) looked at heavy traffic. Here the load $\rho = N\lambda/(R\mu)$ gets close to one or goes past it. In that setting, the diffusion estimate of the mean queue length matched the exact multi server answers fairly closely. At the same time, the run time fell a lot versus the matrix-based approach. This fits what people often say in queueing theory. Diffusion methods tend to work best in the same corner where exact methods are painful to compute. That makes the two approaches complement each other in a useful way.

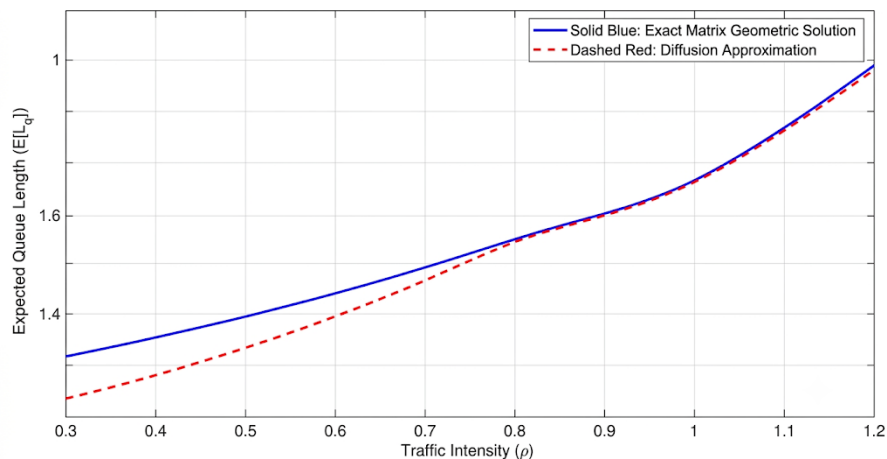


Figure 3: Comparison of Expected Queue Length, Matrix Geometric versus Diffusion Approximation Across Traffic Intensity

You would put expected queue length on the vertical axis. On the horizontal axis you would place traffic intensity, ρ , starting near 0.3 and going up to about 1.2. There should be two lines on the plot. One line comes from the exact matrix geometric method. The other line comes from the diffusion approximation. At smaller values of ρ , the two lines pull apart clearly. This matches what is already known: the diffusion approach tends to be less accurate when the system is not very busy. Once ρ rises past around 0.7, the lines come much closer. That pattern fits the heavy-traffic results that Baghel (2019b) reported for multi-server M/M/R repair systems. So the key point is simple. The diffusion method works well as a quick stand-in only when the system is fairly busy.

V. Practical Implications for Machining Systems with Spares

5.1 Sizing the Spare Pool

A direct task for this modelling method is to decide how many spare parts, S , a machining line should keep on hand. If the stock is too low, the machines wait while repairs are queued, and uptime drops. If the stock is too high, money is stuck in equipment that sits unused. With the matrix geometric approach, a planner can work out the availability, A , for several choices of S and R . Then the planner can spot where the gain from one more spare begins to level off. Baghel (2020a) reported this kind of pattern in work on reliability and availability for repairable systems, looking at how different spare-pool sizes change the results.

5.2 Repairman Staffing Under Reneging

When you add another repair worker, something extra happens once reneging is in the model. The queue gets smaller, yes, but the reneging rate also drops. Machines do not stay in line as long, so fewer of them reach the point where they lose patience. Because of that, a model that leaves reneging out will miss part of the gain from hiring. It will make the benefit seem weaker than it really is. This is why Baghel (2019a) and Baghel (2020b) push for explicit balking and reneging, instead of assuming the repair queue is patient forever.

5.3 Queue-Length Dependent Service

Some machining setups can adjust the repair speed based on how long the work queue has already grown. In one common way to do this, a foreman may send another technician to assist when the queue gets past a set length. With that change, the service rate can rise as the system size n increases. Baghel (2020c) studied a queue length driven service rate like this. The work used a diffusion model in flexible manufacturing systems. The results suggest this kind of adaptive rule lowers the chance of long downtime. It does so compared with a plan that keeps the service rate fixed. The study also noted that this does not mean the crew must stay larger all the time.

VI. Conclusion

A shop keeps moving only if broken machines return fast. This work looks at how that return time can be studied when spare parts exist, several repair workers are active, and some people lose patience and stop waiting.

It uses two math views for the same queue. One is the matrix geometric approach. It gives a clean way to get steady state odds. It follows the earlier method from Neuts and then the extensions by Latouche and Ramaswami. It fits well with the way the repair line is built, with a finite quasi birth death pattern.

The second view is the diffusion approximation. It gives up some accuracy to make the work easier. It treats the count of failed units like a continuous variable. That count follows a stochastic differential equation. This shortcut helps most when traffic is high, and when the full matrix steps become too hard to run.

These two tools do not cancel each other out. The matrix geometric method is still the go to option when the system is small, or when the reneging point is near, since the diffusion method can drift there. When systems get large, the diffusion view is usually the better fit. It cuts down the matrix effort while still matching the true results closely in moderate to heavy load. The same pattern shows up in the studies mentioned about machine interference and repair in India.

Plant managers and reliability engineers can use this modeling for clear results. It supports decisions on how many spare parts to keep and how many repair workers to schedule. It also helps compare the cost of missed service with the cost of machines sitting idle. As machining lines become more automated, and as more units fit into the same space, these tradeoffs will matter more. With a precise method and a simpler estimate available, planners do not have to rely on guesswork. That gives them a better footing when planning spares, staffing, and shutdown impact.

References

- [1]. Baghel, K. P. S. (2019a). Markovian modeling of machine interference problems with finite sources and balking/renegeing behavior. *Quest Journals: Journal of Research in Applied Mathematics*, 5(12), 1–6. <https://doi.org/10.35629/0743-05120106>
- [2]. Baghel, K. P. S. (2019b). Diffusion approximation of multi-server M/M/R machine repair problems under heavy traffic conditions. *International Journal of Engineering and Science Invention*, 8(5), 84–90. <https://doi.org/10.35629/6734-08058490>
- [3]. Baghel, K. P. S. (2019c). Comparative study of single vs batch service policies in M/M/C queueing systems with breakdown and repair mechanisms. *IOSR Journal of Mathematics*, 15(2), 85–91. <https://doi.org/10.9790/5728-1502018591>
- [4]. Baghel, K. P. S. (2020a). Reliability and availability analysis of multi-component repairable systems using M/M/R models with standby spares. *International Journal of Engineering and Science Invention*, 9(9), 87–92. <https://doi.org/10.35629/6734-09098792>
- [5]. Baghel, K. P. S. (2020b). Stochastic analysis of machine repair systems with reneging and limited spares using M/M/C queue models. *International Journal of Mathematics and Statistics Invention*, 8(6), 43–48. <https://doi.org/10.35629/4767-08064348>
- [6]. Baghel, K. P. S. (2020c). Queue-length dependent service rates in flexible manufacturing systems: A diffusion process approach. *Quest Journals: Journal of Research in Applied Mathematics*, 6(8), 1–6. <https://doi.org/10.35629/0743-06080106>
- [7]. Bhat, U. N. (1972). *Elements of applied stochastic processes*. Wiley.
- [8]. Chaudhry, M. L., & Templeton, J. G. C. (1983). *A first course in bulk queues*. Wiley.
- [9]. Cox, D. R., & Miller, H. D. (1965). *The theory of stochastic processes*. Chapman and Hall.
- [10]. Gross, D., & Harris, C. M. (1998). *Fundamentals of queueing theory* (3rd ed.). Wiley.
- [11]. Haque, L., & Armstrong, M. J. (2007). A survey of the machine interference problem. *European Journal of Operational Research*, 179(2), 469–482. <https://doi.org/10.1016/j.ejor.2006.02.036>
- [12]. Jain, M., & Sharma, G. C. (2002). M/M/R machine repair problem with spares and state dependent rates. *South African Journal of Statistics*, 36(2), 141–150.
- [13]. Ke, J.-C., & Wang, K.-H. (2007). Vacation policies for machine repair problem with two type spares. *Applied Mathematical Modelling*, 31(5), 880–894. <https://doi.org/10.1016/j.apm.2006.02.009>
- [14]. Kleinrock, L. (1975). *Queueing systems, volume 1: Theory*. Wiley-Interscience.
- [15]. Latouche, G., & Ramaswami, V. (1999). *Introduction to matrix geometric methods in stochastic modeling*. SIAM.
- [16]. Medhi, J. (2003). *Stochastic models in queueing theory* (2nd ed.). Academic Press.
- [17]. Neuts, M. F. (1981). *Matrix-geometric solutions in stochastic models: An algorithmic approach*. Johns Hopkins University Press.
- [18]. Shawky, A. I. (2000). The single-server machine interference model with balking, reneging and an additional server for longer queues. *Microelectronics Reliability*, 40(2), 349–353. [https://doi.org/10.1016/S0026-2714\(99\)00263-2](https://doi.org/10.1016/S0026-2714(99)00263-2)
- [19]. Takagi, H. (1991). *Queueing analysis: A foundation of performance evaluation, volume 1: Vacation and priority systems*. North-Holland.
- [20]. Wang, K.-H., & Chang, Y.-C. (2002). Cost analysis of a finite M/M/R queueing system with balking, reneging, and server breakdowns. *Mathematical Methods of Operations Research*, 56(2), 169–180. <https://doi.org/10.1007/s001860200205>