

Analysis of Multi Level Feedback Queue Scheduling Using Markov Chain Model with Data Model Approach

Shweta Jain

Department of Computer Applications, Shri R.G.P. Gujarati Professional Institute, Indore-10
Email: shwjain@yahoo.com

Dr. Saurabh Jain

Institute of Computer Applications, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore
Email: iamsaurabh_4@yahoo.co.in

ABSTRACT

When a process gets the CPU, the scheduler has no idea of the precise amount of CPU time the process will need. Process scheduling algorithms are used for better utilization of CPU. The number of processes arriving to the CPU at a time comes in mass volume which causes a long waiting queue. In Multilevel feedback queue scheduling, the scheduler moves from one queue to another in order to perform the processing follow the transition mechanism. This paper analysed a general transition scenario for the functioning of CPU scheduler in multilevel queue with feedback mechanism. We proposed a Markov chain model to analyze this transition phenomenon with a general class of scheduling scheme. Simulation study is performed to evaluate the comparative study with the help of varying values of α and d in a mathematical model.

Keywords - Markov chain model, Multi-level feedback queue scheduling, Process queue, Transition probability matrix, Wait State.

Date of Submission: May 31, 2016

Date of Acceptance: Jun 25, 2016

I. INTRODUCTION

Multilevel feedback queue (MLFQ) is most suitable and ideal scheduling algorithm for separating processes into categories based on their need for the CPU. It allows a process to move between queues. If we have not given the relative length of various processes then none of these scheduling: Shortest Process Next, Shortest Remaining Time and Highest Response Ratio Next can be used. Another way of establishing preferences for shorter jobs is to penalize jobs that have been running longer. In other words, if we cannot focus on the time remaining to execute, so we can focus on the time spent in execution so far. MLFQ scheduling is used on preemptive basis with dynamic priority mechanism. When a process first enters a system, it is placed in first queue. When it returns to the ready state after its first execution, it is placed in second queue. After each subsequent execution, it goes to the next lower-priority queue. A shorter process will complete quickly without migrating very far down the hierarchy of ready queues. A longer process can be gradually drifted downward. Thus some processes in queues are used simple FCFS mechanism but few queues are treated as round robin (RR) fashion. One common variation of the MLFQ mechanism is to have a process circulate RR several times through each queue before it moves to the next lower queue. Usually the number of cycles through each queue is increased as the process moves to the next lower queue.

RELATED WORK

MLFQ may be one of most potential scheduling technique for CPU. It is the extension of multi-level queue scheduling algorithm where it is the results of combination of basic scheduling algorithms such as FCFS and RR scheduling algorithm. In MLFQ, where the processes can move from one queue to another queue but in MLQ scheduling processes are assigned to a fixed queue. There are various type of approaches proposed in these papers by different authors to increase the overall performance of the MLFQ and other scheduling algorithms. Jain et al. [1] presented a Linear Data Model based study of Improved Round Robin CPU Scheduling algorithm with features of Shortest Job First scheduling with varying time quantum whereas Chavan and Tikekar [2] derived an Optimum Multilevel Dynamic Round Robin scheduling algorithm, which calculates intelligent time slice and changes after every round of execution.

Suranauwarat [3] used simulator to learn scheduling algorithms in an easier and a more effective way. Sindhu et al. [4] proposed an algorithm which can handle all types of process with optimum scheduling criteria. Li et al. [5] analyzed existing fair scheduling algorithms are either inaccurate or inefficient and non-scalable for multiprocessors to produce larger scale multi-core processors that solves this problem by a new scheduling algorithm called Distributed Weighted Round-Robin (DWRR). Hieh and Lam [6] discussed smart schedulers for multimedia users. Saleem and Javed [7] developed a comprehensive tool which runs a simulation in real time, and generates data to be used for evaluation. Raheja et al. [8] proposed a new scheduling algorithm

called Vague Oriented Highest Response Ratio Next (VHRRN) scheduling algorithm which computes the dynamic priority using vague logic in Fuzzy Systems (FUZZ) and [9] also proposed a 2-layered architecture of multilevel queue scheduler based on vague set theory (VMLQ) that scheduler handles the impreciseness of data as well as improving the starvation problem of lower priority tasks which optimizes the performance metrics and improves the response time of system through simulation.

Shukla and Jain [10] have discussed the use of Markov chain model for multilevel queue scheduler and also proposed a data model based Markov chain model to study the transition phenomenon, designed a scheduling scheme and compared through deadlock-waiting index measure with simulation study[11]. Shukla et al. [12] analyzed round robin scheme using Markov chain model. Helmy and Dekdouk [13] introduced Burst Round Robin, a proportional-share scheduling algorithm as an attempt to combine the low scheduling overhead of round robin algorithms and favor shortest jobs. Maste et al. [14] proposed a new variant of MLFQ algorithm, in which time slice is assigned to each queue such that it changes with each round of execution dynamically i.e used dynamic time quantum and used neural network to adjust this time slice to optimize turnaround time with static time slice for each queue.

Yadav and Upadhyay [15] suggested a novel approach which will improve the performance of MLFQ. Chahar and Raheja [16] analysed basic multilevel queue and multilevel feedback queue scheduling techniques and thereafter discussed a review of techniques proposed by different authors. Rao and Shet [17] articulated the task states of New Multi Level Feedback Queue [NMLFQ] Scheduler and depicted the contingent of task transitions with triggers which leads to a change of state with time instants and intervals are elucidated and [18] also analysed distinguishing problems with existing MLFQ scheduling algorithm to develop a New Multi Level Feedback Queue (NMLFQ) illustrated along with synchronization through semaphore with the relationships of job sets, record of tasks and queues in scheduler involving respective steps describing object oriented code to justify the algorithm. Jain and Jain [19] discussed the various approaches of scheduling algorithm and probability-based Markov chain analysis to determine the performance of these algorithms.

This paper referred different CPU scheduling and their various aspects by Silberschatz and Galvin [20], Stalling [21], Tanenbaum and Woodhull [22], Dhamdhare [23] and Deitel [24] whereas stochastic processes and Markov chain model by Medhi [25]. Motivation from this analysis, this paper proposed a class of scheduling scheme under the assumption of Markov chain model and using a data model approach.

II. GENERAL CLASS OF MULTI-LEVEL FEEDBACK QUEUE SCHEDULING

Multilevel Feedback Queue algorithm have number of queues and the processes move between these queues, a scheduling algorithm for each queue, a method

used to determine when to upgrade a process, a method used to determine when to demote a process and a method used to determine on which queue a process begins to have each time it returns to the ready state. It is based on the model of Multi-Channel Single server queue. Fundamental advantages possessed by the multilevel feedback queue are having processes of moving from one queue to another queue, for instance with lower priority or higher priority. In this scheme, if the priority of one process is more than another process, then first one is executed and if the priority of one process is same to another process then its running on FSFC (First Come, First Serve) manner. If a job is entered into the system, then the work comes into the category highest priority in the topmost queue. If the job requires extra time when running, its priority is lowest or displacement of the lower queue. The lowermost queue works in FCFS manner. If the processing is completed before time, permanent jobs at the same level.

ASSUMPTION OF THE MODEL

Suppose a multi-level feedback queue scheduling with five active queues Q_1, Q_2, Q_3, Q_4, Q_5 which are treated here as states each having large number of processes $P_j, P_j', P_j'', P_j''', P_j''''$ ($j=1, 2, 3, 4, 5, \dots$) respectively for processing and one waiting states Q_6 . These queues are characterized and organized on the basis of some features like priority, size, or weight. The new arrival of processes are allocated in random manner in these queues. Define Q_i ($i=1, 2, 3, 4, 5$), the states of scheduling system and a specific state Q_6 which is a waiting state. First five states are for arrival and inputting of processes while the last one associate with waiting of the scheduler. A quantum is considered as a small pre-defined slot of time given for processing in different queues to the processes. Symbol n denotes the n^{th} quantum allotted by the scheduler to a process for execution ($n = 1, 2, 3, 4, 5 \dots$). So some assumption steps for the model are:

1. All the first five queues Q_1, Q_2, Q_3, Q_4 and Q_5 are allowed to accept a new process with initial probabilities $pb_1, pb_2, pb_3, pb_4, pb_5$ with the probability condition

$$\sum_{i=1}^5 pb_i = 1$$

2. $Q_6=W$ is used as waiting state in the transition system. Any special conditions over waiting or restricting transition can be considered within this scheduling scheme.
3. The Scheduler has a random movement over all states Q_i ($i=1, 2, 3, 4, 5$) on quantum variation.
4. A new arrival process is picked by the scheduler with a predefined priority of any queue Q_i and allots a quantum of time for processing.
5. While the process remains with the CPU if the allotted quantum is not over. If a process completes within this quantum, then it leaves the queue Q_i and if a process remains incomplete within quantum, scheduler assigns next quantum to the next process of the same queue. The previous incomplete process moves to next queue Q_{i+1} where $(i+1) \leq 5$ and waits

there for next quantum to be allotted for its processing.

- The scheduler can jump from one state to other state at the end of a quantum. The quantum allotment procedure continues by scheduler within Q_i until all Q_i are empty. In that case the scheduler moves towards Q_6 . The scheduler can also move to Q_6 after the completion of any quantum although all Q_i ($i = 1, 2, 3, 4, 5$) are full with processes.

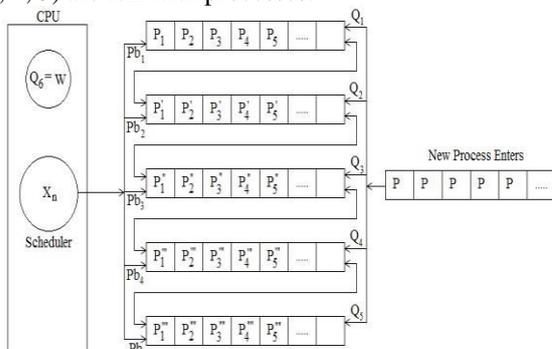


Figure 2.1: The General Multilevel Feedback Queue System

III. MARKOV CHAIN MODEL FOR THE PROPOSED SYSTEM

Define Q_1 as state 1, Q_2 as state 2, Q_3 as state 3, Q_4 as state 4, Q_5 as state 5 and Q_6 as waiting state W. The symbol n denotes the n^{th} quantum of time assigned by scheduler for executing a process ($n=1, 2, 3, 4, \dots$). Assume the movement of scheduler system is random over the different processing states and waiting state.

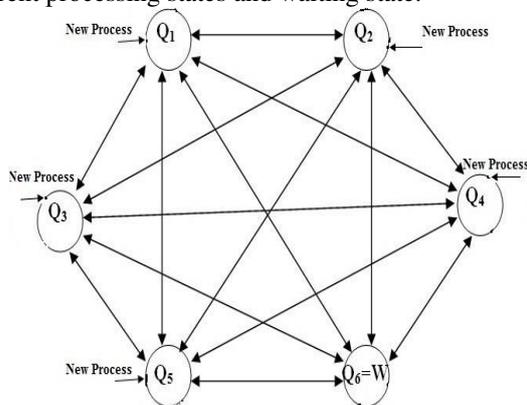


Figure 3.1: Unrestricted Transition Diagram

Figure 3.1 depicts the transition diagram which shows the transition from one state to another state.

Let $\{x^{(n)}, n \geq 1\}$ be a Markov chain where $x^{(n)}$ denotes the state of the scheduler at the quantum of time. The state space for the random variable $x^{(n)}$ is $\{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6\}$ where $Q_6=W$ is waiting state and scheduler X moves stochastically over different processing states and waiting states within different quantum of time. Predefined selections for initial probabilities of states are:

$$\left. \begin{aligned} P[X^{(0)} = Q_1] &= pb_1 \\ P[X^{(0)} = Q_2] &= pb_2 \\ P[X^{(0)} = Q_3] &= pb_3 \\ P[X^{(0)} = Q_4] &= pb_4 \\ P[X^{(0)} = Q_5] &= pb_5 \\ P[X^{(0)} = Q_6] &= pb_6 \end{aligned} \right\} \dots\dots\dots(1)$$

$$\text{With } pb_1 + pb_2 + pb_3 + pb_4 + pb_5 + pb_6 = \sum_{i=1}^6 pb_i = 1, \text{ where } pb_6 = 0.$$

Let S_{ij} ($i, j=1,2,3,4,5,6$) be the unit step transition probabilities of scheduler over six proposed states then transition probability matrix for :

		$X^{(n)}$					
		Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
$X^{(n-1)}$	Q_1	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
	Q_2	S_{21}	S_{22}	S_{23}	S_{24}	S_{25}	S_{26}
	Q_3	S_{31}	S_{32}	S_{33}	S_{34}	S_{35}	S_{36}
	Q_4	S_{41}	S_{42}	S_{43}	S_{44}	S_{45}	S_{46}
	Q_5	S_{51}	S_{52}	S_{53}	S_{54}	S_{55}	S_{56}
	Q_6	S_{61}	S_{62}	S_{63}	S_{64}	S_{65}	S_{66}

Figure 3.2: Transition Matrix

Unit-step Transition Probabilities for the wait state W are as follows:

$$\left. \begin{aligned} S_{16} &= (1 - \sum_{i=1}^5 S_{1i}), & S_{26} &= (1 - \sum_{i=1}^5 S_{2i}) \\ S_{36} &= (1 - \sum_{i=1}^5 S_{3i}), & S_{46} &= (1 - \sum_{i=1}^5 S_{4i}) \\ S_{56} &= (1 - \sum_{i=1}^5 S_{5i}), & S_{66} &= (1 - \sum_{i=1}^5 S_{6i}) \end{aligned} \right\} \dots\dots\dots(2)$$

$$0 \leq S_{ij} \leq 1$$

If S_{ij} ($i, j=1,2,3,4,5$) be the unit-step transition probabilities of scheduler over six proposed states then transition probability matrix for $X^{(n)}$ will be

$$S_{ij} = P [X^{(n)} = Q_i / X^{(n-1)} = Q_j]$$

After first quantum, the state probabilities can be determined by the following expressions:

$$\begin{aligned}
 P[X^{(1)}=Q_1] &= P[X^{(0)}=Q_1] \cdot P[X^{(1)}=Q_1 / X^{(0)}=Q_1] + \\
 &P[X^{(0)}=Q_2] \cdot P[X^{(1)}=Q_1 / X^{(0)}=Q_2] + \\
 &P[X^{(0)}=Q_3] \cdot P[X^{(1)}=Q_1 / X^{(0)}=Q_3] + \\
 &P[X^{(0)}=Q_4] \cdot P[X^{(1)}=Q_1 / X^{(0)}=Q_4] + \\
 &P[X^{(0)}=Q_5] \cdot P[X^{(1)}=Q_1 / X^{(0)}=Q_5] + \\
 &P[X^{(0)}=W] \cdot P[X^{(1)}=Q_1 / X^{(0)}=W] \\
 &= \sum_{i=1}^6 pb_i S_{i1} \quad \dots\dots\dots(3) \\
 P[X^{(1)}=Q_2] &= \sum_{i=1}^6 pb_i S_{i2} \\
 P[X^{(1)}=Q_3] &= \sum_{i=1}^6 pb_i S_{i3} \\
 P[X^{(1)}=Q_4] &= \sum_{i=1}^6 pb_i S_{i4} \\
 P[X^{(1)}=Q_5] &= \sum_{i=1}^6 pb_i S_{i5} \\
 P[X^{(1)}=W] &= \sum_{i=1}^6 pb_i S_{i6}
 \end{aligned}$$

Similarly, after second quantum, the state probabilities can be determined by the following expressions

$$\begin{aligned}
 P[X^{(2)}=Q_1] &= \sum_{j=1}^6 \left\{ \sum_{i=1}^6 (pb_i S_{ij}) \right\} S_{j1} \\
 P[X^{(2)}=Q_2] &= \sum_{j=1}^6 \left\{ \sum_{i=1}^6 (pb_i S_{ij}) \right\} S_{j2} \\
 P[X^{(2)}=Q_3] &= \sum_{j=1}^6 \left\{ \sum_{i=1}^6 (pb_i S_{ij}) \right\} S_{j3} \\
 P[X^{(2)}=Q_4] &= \sum_{j=1}^6 \left\{ \sum_{i=1}^6 (pb_i S_{ij}) \right\} S_{j4} \\
 P[X^{(2)}=Q_5] &= \sum_{j=1}^6 \left\{ \sum_{i=1}^6 (pb_i S_{ij}) \right\} S_{j5} \\
 P[X^{(2)}=W] &= \sum_{j=1}^6 \left\{ \sum_{i=1}^6 (pb_i S_{ij}) \right\} S_{j6} \quad \dots\dots\dots(4)
 \end{aligned}$$

In a similar way, the expression for the nth quantum:

$$\begin{aligned}
 P[X^{(n)}=Q_1] &= \sum_{m=1}^6 \dots \sum_{k=1}^6 \left\{ \sum_{j=1}^6 \left(\sum_{i=1}^6 pb_i S_{ij} \right) S_{jk} \right\} S_{k1} \dots S_{m1} \\
 P[X^{(n)}=Q_2] &= \sum_{m=1}^6 \dots \sum_{k=1}^6 \left\{ \sum_{j=1}^6 \left(\sum_{i=1}^6 pb_i S_{ij} \right) S_{jk} \right\} S_{k1} \dots S_{m2} \\
 P[X^{(n)}=Q_3] &= \sum_{m=1}^6 \dots \sum_{k=1}^6 \left\{ \sum_{j=1}^6 \left(\sum_{i=1}^6 pb_i S_{ij} \right) S_{jk} \right\} S_{k1} \dots S_{m3} \\
 P[X^{(n)}=Q_4] &= \sum_{m=1}^6 \dots \sum_{k=1}^6 \left\{ \sum_{j=1}^6 \left(\sum_{i=1}^6 pb_i S_{ij} \right) S_{jk} \right\} S_{k1} \dots S_{m4} \\
 P[X^{(n)}=Q_5] &= \sum_{m=1}^6 \dots \sum_{k=1}^6 \left\{ \sum_{j=1}^6 \left(\sum_{i=1}^6 pb_i S_{ij} \right) S_{jk} \right\} S_{k1} \dots S_{m5} \\
 P[X^{(n)}=W] &= \sum_{m=1}^6 \dots \sum_{k=1}^6 \left\{ \sum_{j=1}^6 \left(\sum_{i=1}^6 pb_i S_{ij} \right) S_{jk} \right\} S_{k1} \dots S_{m6} \quad \dots\dots\dots(5)
 \end{aligned}$$

IV. SIMULATION STUDY WITH GRAPHICAL ANALYSIS ON MATHEMATICAL DATA MODEL

The basic and scientific approach for graphical data analysis related to state transition probabilities is managed by a mathematical data model with two parameters α and d where value of i stands number of queues.

	$X^{(n)}$					
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
Q_1	a	$a+d.i$	$a+2d.i$	$a+3d.i$	$a+4d.i$	$1-(5a+10d.i)$
Q_2	$a+d.i$	$a+2d.i$	$a+3d.i$	$a+4d.i$	$a+5d.i$	$1-(5a+15d.i)$
Q_3	$a+2d.i$	$a+3d.i$	$a+4d.i$	$a+5d.i$	$a+6d.i$	$1-(5a+20d.i)$
Q_4	$a+3d.i$	$a+4d.i$	$a+5d.i$	$a+6d.i$	$a+7d.i$	$1-(5a+25d.i)$
Q_5	$a+4d.i$	$a+5d.i$	$a+6d.i$	$a+7d.i$	$a+8d.i$	$1-(5a+30d.i)$
Q_6	$a+5d.i$	$a+6d.i$	$a+7d.i$	$a+8d.i$	$a+9d.i$	$1-(5a+35d.i)$

Figure 4.1: Data Model Matrix

CASE 1: When $\alpha=0.1$ and

1. $d = 0.002$

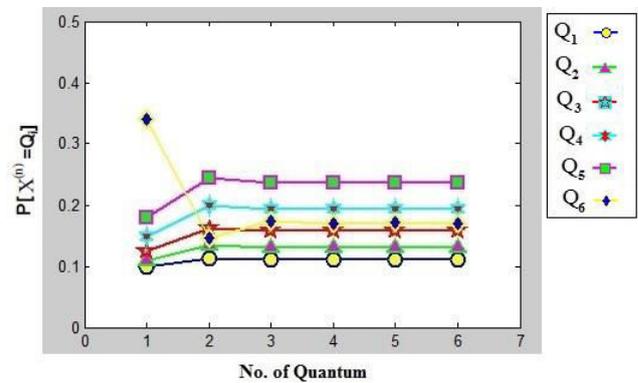


Figure 4.1.1: $\alpha=0.1$ and $d = 0.002$

2. $d = 0.004$

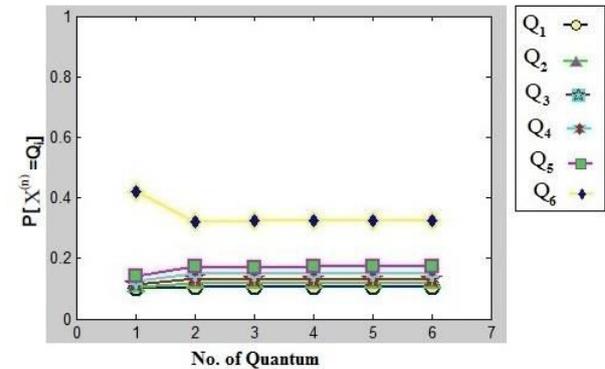


Figure 4.1.2: $\alpha=0.1$ and $d = 0.004$

3. $d = 0.006$

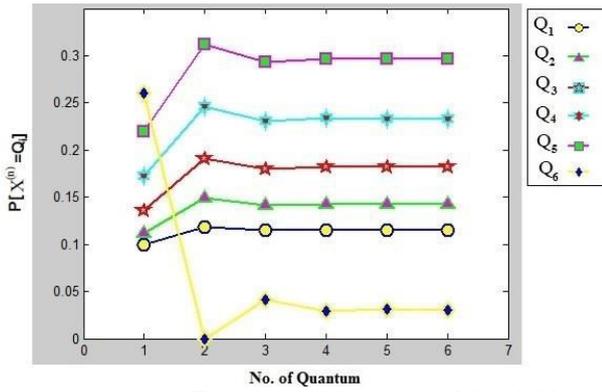


Figure 4.1.3: $\alpha=0.1$ and $d = 0.006$

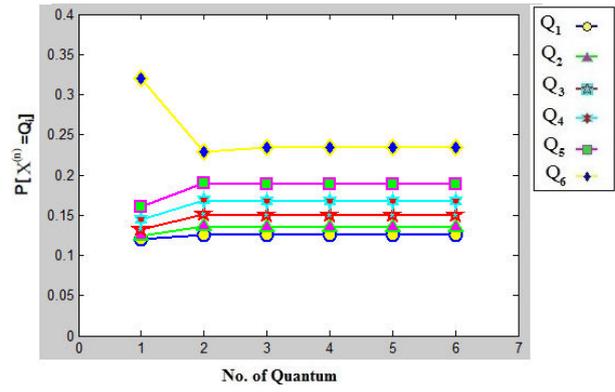


Figure 4.2.1: $\alpha=0.12$ and $d = 0.002$

4. $d = 0.008$

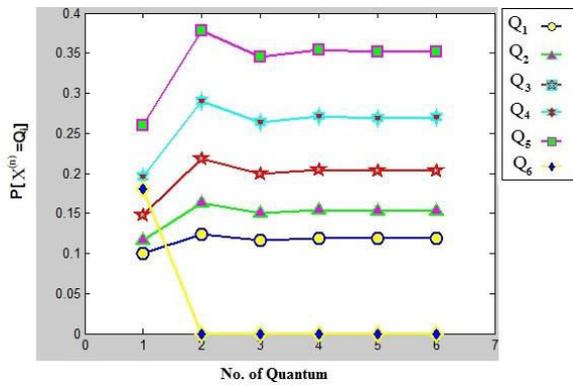


Figure 4.1.4: $\alpha=0.1$ and $d = 0.008$

2. $d = 0.004$

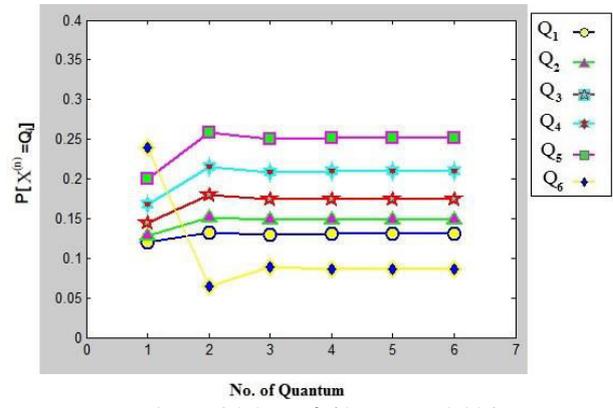


Figure 4.2.2: $\alpha=0.12$ and $d = 0.004$

5. $d = 0.010$

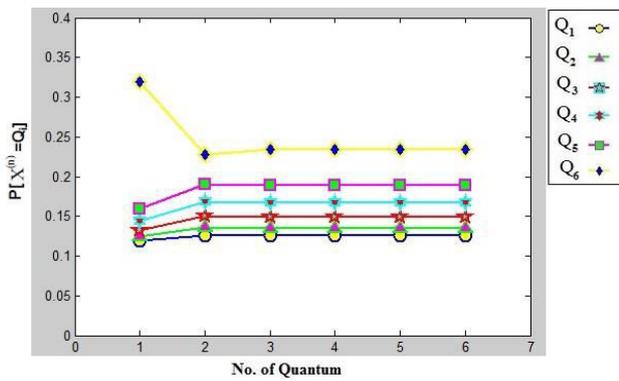


Figure 4.1.5: $\alpha=0.1$ and $d = 0.010$

3. $d = 0.006$

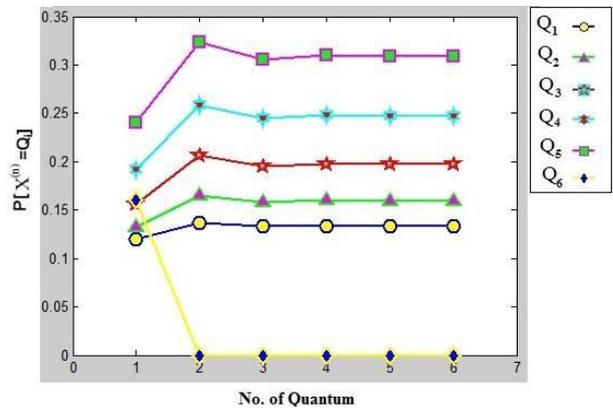


Figure 4.2.3: $\alpha=0.12$ and $d = 0.006$

4. $d = 0.008$

CASE 2: When $\alpha=0.12$ and

1. $d = 0.002$

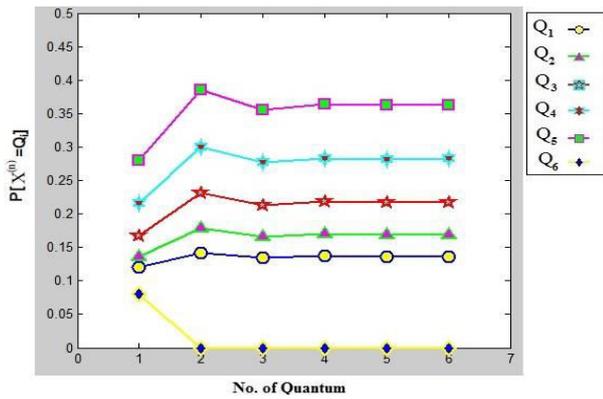


Figure 4.2.4: $\alpha=0.12$ and $d = 0.008$

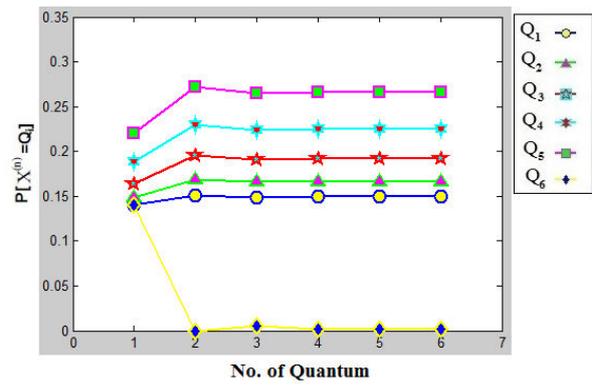


Figure 4.3.2: $\alpha=0.14$ and $d = 0.004$

5. $d = 0.010$

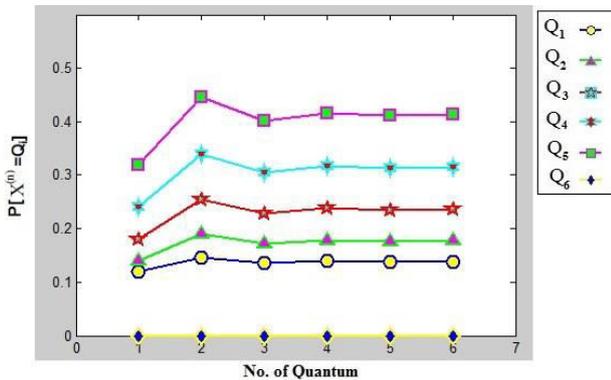


Figure 4.2.5: $\alpha=0.12$ and $d = 0.010$

3. $d = 0.006$

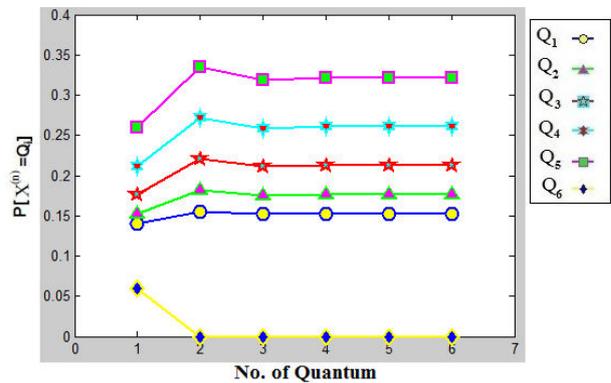


Figure 4.3.3: $\alpha=0.14$ and $d = 0.006$

CASE 3: When $\alpha=0.14$ and

1. $d = 0.002$

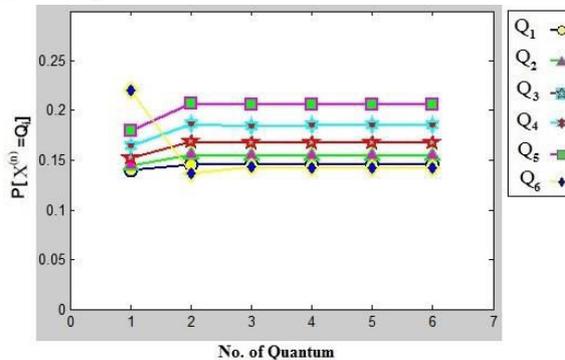


Figure 4.3.1: $\alpha=0.14$ and $d = 0.002$

4. $d = 0.008$

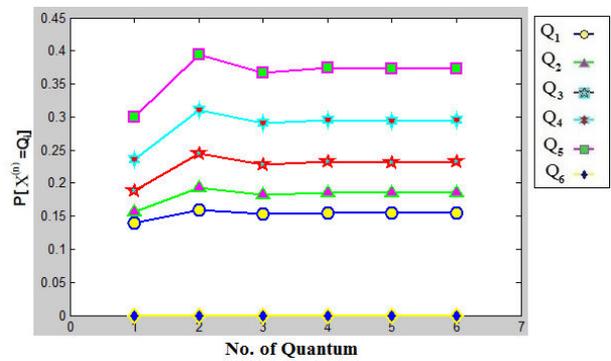


Figure 4.3.4: $\alpha=0.14$ and $d = 0.008$

2. $d = 0.004$

5. $d = 0.010$

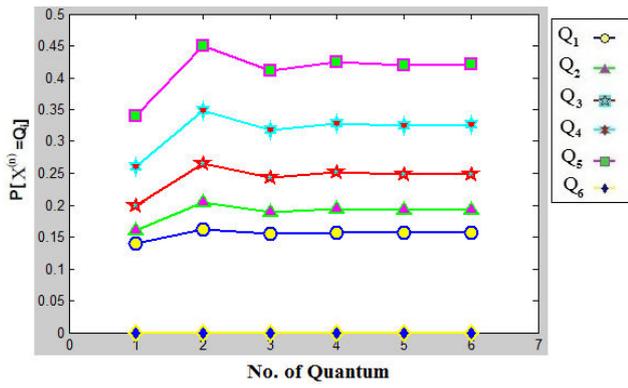


Figure 4.3.5: $\alpha=0.14$ and $d = 0.010$

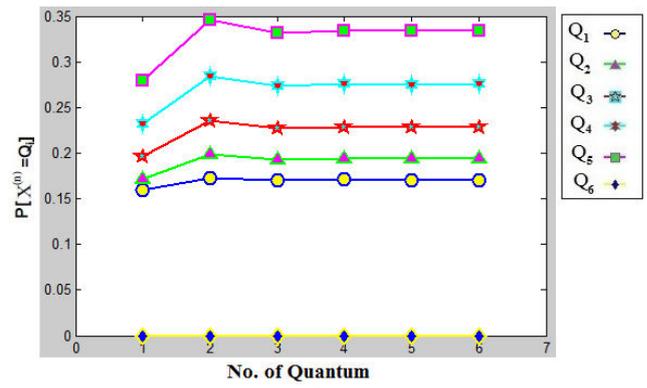


Figure 4.4.3: $\alpha=0.16$ and $d = 0.006$

CASE 4: When $\alpha=0.16$ and
1. $d = 0.002$

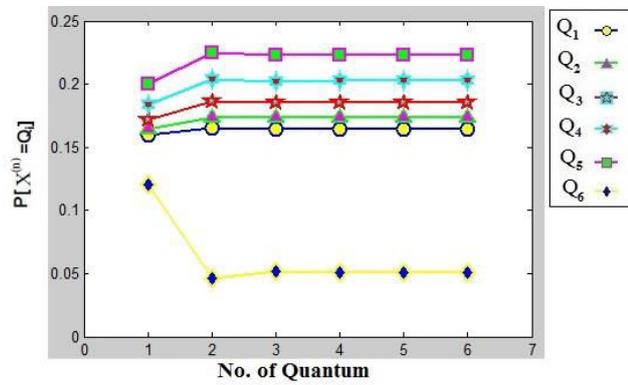


Figure 4.4.1: $\alpha=0.16$ and $d = 0.002$

4. $d = 0.008$

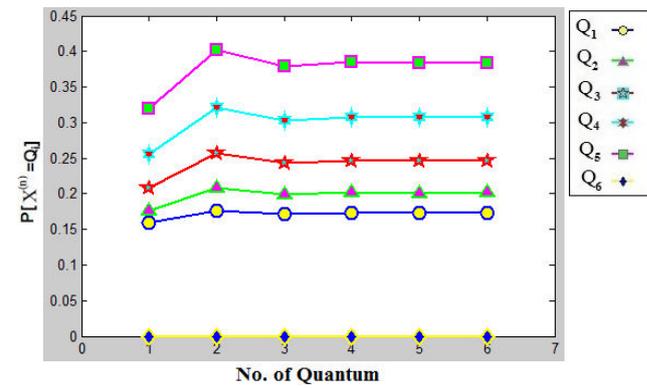


Figure 4.4.4: $\alpha=0.16$ and $d = 0.008$

2. $d = 0.004$

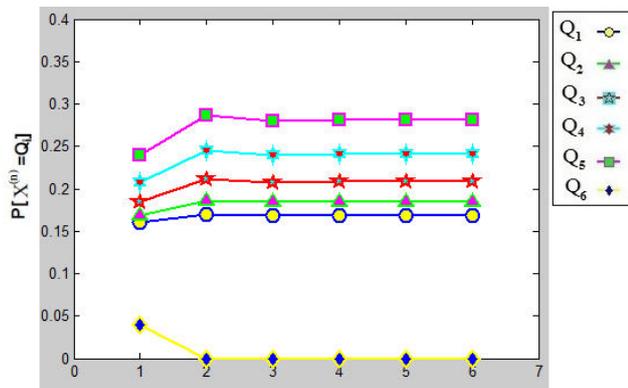


Figure 4.4.2: $\alpha=0.16$ and $d = 0.004$

5. $d = 0.010$

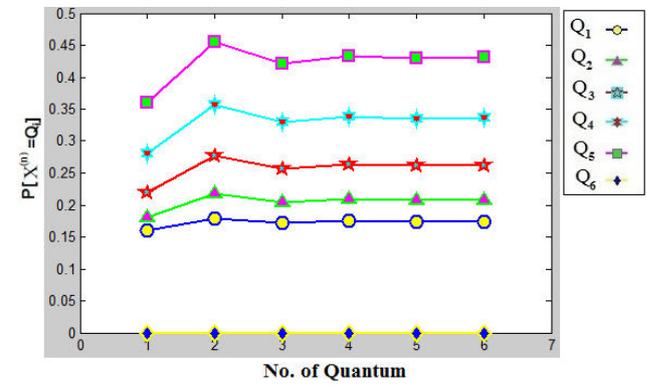


Figure 4.4.5: $\alpha=0.16$ and $d = 0.010$

CASE 5: When $\alpha=0.18$ and

1. $d = 0.002$

3. $d = 0.006$

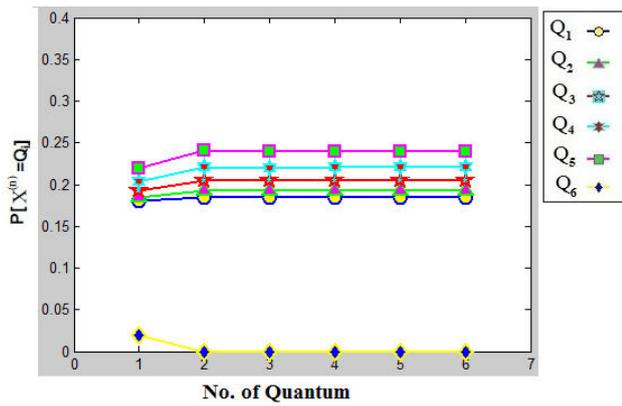


Figure 4.5.1: $\alpha=0.18$ and $d = 0.002$

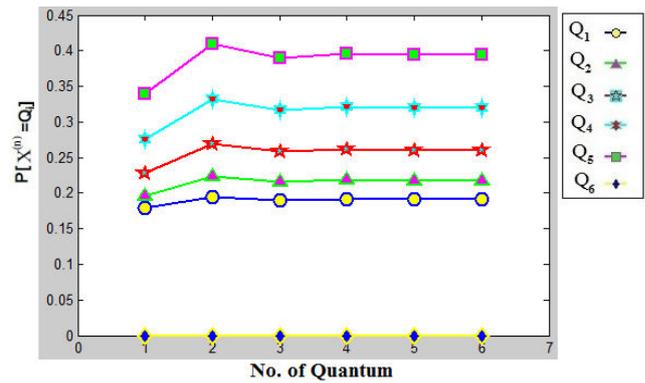


Figure 4.5.4: $\alpha=0.18$ and $d = 0.008$

2. $d = 0.004$

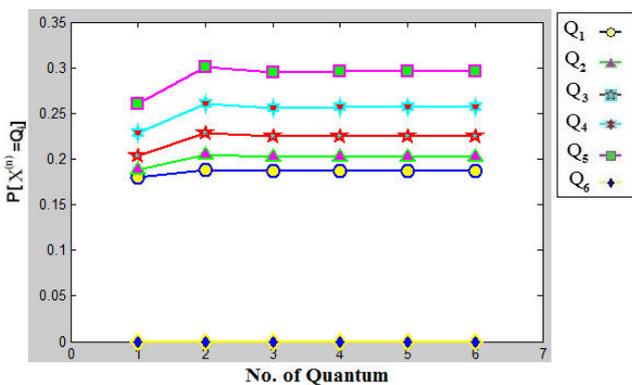


Figure 4.5.2: $\alpha=0.18$ and $d = 0.004$

5. $d = 0.010$

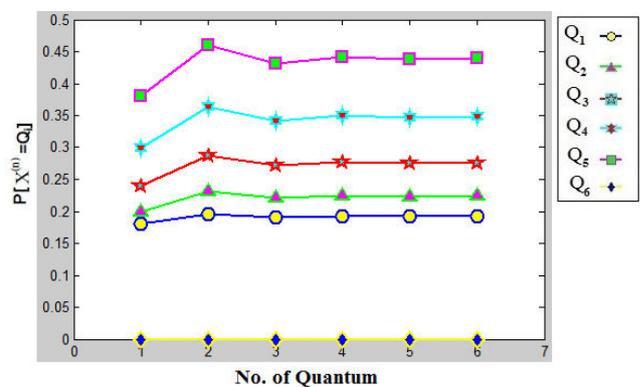


Figure 4.5.5: $\alpha=0.18$ and $d = 0.010$

3. $d = 0.006$

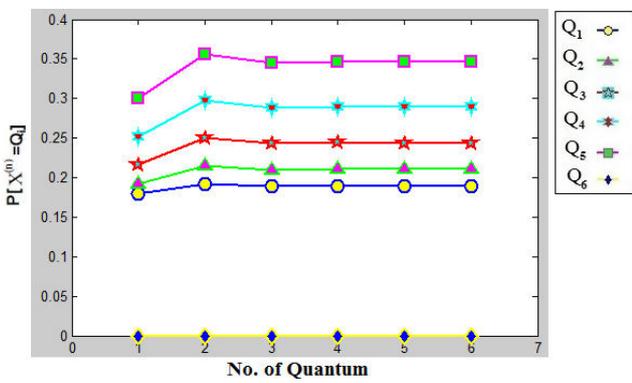


Figure 4.5.3: $\alpha=0.18$ and $d = 0.006$

4. $d = 0.008$

V. DISCUSSION ON GRAPHS

Case 1: When $\alpha=0.1$ and increasing value of d from 0.002 to 0.010, we observed pattern of the transition probabilities of Q_1 to Q_5 are similar in varying quantum, But few are very close (fig.4.1.1 and fig.4.1.5) and few are comparatively far (fig.4.1.2, fig.4.1.3 and fig.4.1.4) to each other. Now we find the waiting state Q_6 is moving high, going to down up to second quantum after that nearly straight according to quantum increases. As d increases this probability being reduces that can be seen in fig.4.1.3 and fig.4.1.4.

Case 2: At $\alpha=0.12$, $d=0.002$ (fig.4.2.1) waiting state $w=Q_6$ the system is more likely to shift over to high. This is not a good sign for scheduling procedure. With increasing values of d , the scheduler of processing all queues from Q_1 to Q_5 get consolidated in fig 4.2.1 and 4.2.2 but in fig.4.2.3-- fig 4.2.5 queues get fared to each other. Along with this, a remarkable drop in the wait state could also be observed near to zero from fig.4.2.1 to fig.4.2.5

Case 3: With $\alpha=0.14$ and varying values of d , we find that almost all the pattern in fig.4.3.1- fig.4.3.5 queues from Q_1 to Q_5 remains constant over little high as increasing of d . But waiting state Q_6 is going to drop continuously to zero especially in fig. 4.3.2- fig.4.3.5.

Case 4: At $\alpha=0.16$ and d from 0.002 to 0.010, we analyzed system transition probabilities of Q_1 to Q_5 has moved

higher chance of receiving the scheduler. With increment of d values these going to lower in fig.4.4.1 and nullifies to zero in fig 4.4.2- fig. 4.4.5.

Case 5: At $\alpha=0.18$ and for $d=0.004-0.010$, we find equal chances of receiving the scheduler for all the queues Q_1 to Q_5 moving higher except $d=0.002$ (comparatively closer). With the increment of d , we find that the chance of scheduler going on to Q_6 (the waiting state) nullifies to its minimum in fig.4.5.1-fig.4.5.5.

VI. CONCLUSION

This paper proposed Markov chain analysis for multi-level feedback queue scheduling scheme finding the effects of wait state with the overall throughput and performance of the system. With the varying values of α and d , we observe that the system determines some of the interesting combination of it which provides some clue regarding better choice of queues over others for high priority processes. This analysis also highlights a path for designing of system with chances of graceful degradation with wait state constantly lower over the increasing quantum. In each and every graph, with successive value of d in the different specified conditions, an interesting trend of waiting probability can be determined. Therefore this paper also suggests this fact that the initial combinations of α and d are the better choice as they are showing less chance of system going on waiting state then their higher counterparts.

REFERENCES

- [1] Jain, S., Shukla, D. and Jain, R. Linear Data Model Based Study of Improved Round Robin CPU Scheduling algorithm, *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6), June 2015, 562-564.
- [2] Chavan, S. R. and Tikekar, P. C., An Improved Optimum Multilevel Dynamic Round Robin Scheduling Algorithm, *International Journal of Scientific & Engineering Research*, 4(12), 2013, 298-301.
- [3] Suranauwarat, S., A CPU Scheduling Algorithm Simulator, *IEEE Proceedings (Frontiers in Education Conference - Global Engineering: Knowledge without Borders, Opportunities without Passports, 2007. FIE '07. 37th Annual)*, 19-24.
- [4] Sindhu, M., Rajkamal, R. and Vigneshwaran, P., An Optimum Multilevel CPU Scheduling Algorithm, *IEEE (International Conference on Advances in Computer Engineering (ACE))*, 2010, 90-94.
- [5] Li, T., Baumberger, D. and Hahn, S., Efficient and Scalable Multiprocessor Fair Scheduling using Distributed Weighted Round-Robin, *ACM*, 2009.
- [6] Hieh, J. and Lam L. S., A SMART Scheduler for Multimedia Application, *ACM Transactions on Computer System (TOCS)*, 2(21), 2003, 117-163.
- [7] Saleem, U. and Javed, M. Y., Simulation of CPU Scheduling Algorithms, *IEEE (TENCON 2000 Proceedings)*, 2, 2000, pp.562-567.
- [8] Raheja, S., Dadhich, R. and Rajpal, S., Vague Oriented Highest Response Ratio Next (VHRRN) Scheduling Algorithm in Fuzzy Systems (FUZZ), *2013-IEEE International Conference (IEEE)*, 2013,1-7.
- [9] Raheja, S., Dadhich, R. and Rajpal, S., 2-Layered Architecture of Vague Logic Based Multilevel Queue Scheduler, *Applied Computational Intelligence and Soft Computing*, Hindawi Publishing Corporation, Article ID 341957, 2014, 1-12.
- [10] Shukla, D. and Jain, S., A Markov Chain Model for Multilevel Queue Scheduler in Operating System, *Proceedings of International Conference on Mathematics and Computer Science*, ICMCS-07, 2007(a), 522-526.
- [11] Shukla, D. and Jain, S., Deadlock State Study in Security Based Multilevel Queue Scheduling Scheme in Operating System, *Proceedings of National Conference on Network Security and Management*, NCNSM-07, 2007(b), 166-175.
- [12] Shukla, D., Jain, S., Singhai, R. and Agarwal, R. K., A Markov Chain Model for the Analysis of Round-Robin Scheduling Scheme, *International Journal of Advanced Networking and Applications*, 1(1), 2009, 1-7.
- [13] Helmy T. and Dekdouk A., Burst Round Robin: As a Proportional-Share Scheduling Algorithm, *IEEE Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations*, 14(11), November 2007, 424-428.
- [14] Maste, D., Ragha, L. and Marathe, N., Intelligent Dynamic Time Quantum Allocation in MLFQ Scheduling, *International Journal of Information and Computation Technology* ©International Research Publications House, 3(4), 2013, 311-322.

- [15] Yadav, R. K. and Upadhayay, A., A fresh loom for Multilevel Feedback Queue Scheduling Algorithm, *International Journal of Advances in Engineering Sciences* © RG Education Society (INDIA), 2(3), July 2012, 21-23.
- [16] Chahar, V. and Raheja, S., A Review of Multilevel Queue and Multilevel Feedback Queue Scheduling Techniques, *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(1), January 2013, 110-113.
- [17] Rao, M. V. P. and Shet, K. C., Analysis of New Multi Level Feedback Queue Scheduler for Real Time Kernel, *International Journal Of Computational Cognition*, 2010 Yang's Scientific Research Institute, 8(3), September 2010, 5-16.
- [18] Rao, M. V. P. and Shet, K. C., Task States, Triggers and Timeline of New Multi-Level Feedback Queue Scheduler, *International Journal of Computer Science and Mobile Computing*, 3(7), July 2014, 807-816.
- [19] Jain, S. and Jain, S., A Research Survey and Analysis for CPU Scheduling Algorithms using Probability-Based Study, *International Journal of Engineering and Management Research*, 5(6), December 2015, 628-633.
- [20] A. Silberschatz, and P. Galvin, Operating system concept, (Ed.8, John Wiley and Sons (Asia), 2010).
- [21] W. Stalling, Operating systems, (Ed.5, Pearson Education, Singapore, Indian Edition, New Delhi,2004).
- [22] A. Tanenbaum, and A.S. Woodhull, Operating System, (Ed. 8, Prentice Hall of India, New Delhi, 2000).
- [23] D. M. Dhamdhare, System Programming and Operating Systems, (Revised Ed. 8, Tata McGraw-Hill Education Pvt. Ltd.(New Delhi), 2009).
- [24] H. M. Deitel, An Introduction to Operating System, (Second edition, Pearson Education Pvt. Ltd., Singapore, Indian Ed, New Delhi, 1999).
- [25] J. Medhi, Stochastic processes, (Ed. 4, Wiley Limited (Fourth Reprint), New Delhi, 1991).

Author's Biography



Mrs. Shweta Jain received her M.C.A. degree from Barakatullah University, Bhopal, in 1999. She worked as Software Engineer since 1999 to 2004 in various organizations. She served as Associate Professor in Computer Science and Application Department in Shri R.G.P. Gujarati Professional Institute, Indore, for 10 years, since 2006. Now she is pursuing her Ph.D. in Computer Science. Her areas of interest include Operating systems and Artificial Intelligence. She has published research papers in International and National Conferences and Journals.



Dr Saurabh Jain has completed M.C.A. degree in 2005 and Ph.D. (CS) in 2009 from Dr. H.S. Gour Central University, Sagar. He worked as Lecturer in the department of Comp. Science & Applications in the same University since 2007 to 2010. Currently, He is working as an Associate Professor and Coordinator in institute of Computer Applications in Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore since 2010. He did his research in the field of Operating system. In this field, he authored and co-authored 25 research papers in National/International Journals and Conference Proceedings. His current research interest is to analyze the scheduler's performance under various algorithms.