International Journal of Management, IT & Engineering
Vol. 8 Issue 9, September 2018,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

STOCHASTIC PROCESS TO ANALYZE BEHAVIOR OF IMPROVED ROUND ROBIN CPU SCHEDULING ALGORITHM

Rupesh Sendre*

RahulSinghai**

Keywords:

CPU Scheduling;
Improved Round Robin Algorithm;
Markov Chain Analysis;
Numerical Data Model;
Transition Probability Matrix.

Abstract

CPU scheduling is one of the root structures for multiprogramming to increase the performance of operating system function that determines which of the process should be executed next when multiple run-able process is waiting in the ready queue. The Round Robin scheduling algorithm is found relatively better than another existing CPU scheduling algorithm. In the proposed paper a data model-based Markov chain analysis of improved round robin algorithm is done in order to determine the transition phenomenon of scheduler. Some specific schemes are performed as its particular cases and generated the results by the mathematical simulation process on the different data sets. The markovian approach of improved round robin scheduling algorithm is transparent and sufficient enough to provide the actual order of shifting various processes and also to determine the order of their execution. These efforts have found very efficient and useful. Further some simulation studies with graphical representation have been done to justify the proposed suggestions.

***Research Scholar, Faculty of Computer Science,PAHER University, Udaipur, Rajasthan, India.**

**** Sr. Assistant Professor, Dept. of International Institute of Professional Studies,Devi Ahilya University, Indore, M.P., India.**

1. Introduction

In Round Robin (RR) principle, processes are assigned to CPU according to their order of arrival. However, unlike FCFS, the processes get only a fixed quantum of CPU time in each iteration. RR therefore avoids a long wait for first CPU response. A process may thus need several rounds for completion. A major drawback in RR policy is that even if a process is near completion, it is still placed at the rear end of first queue, which not only increases the total waiting time but also decreases the throughput. In Improved Round Robin (IRR) scheduling policy, the basic functions of RR are combined with an improvement towards the priority assigned to the processes nearing completion. This algorithm is most suitable and ideal scheduling algorithm for separating processes into categories based on their needs for the processor. It allows scheduler to switch between processes residing in different queues and assign them to processor for execution. The following figure 1.1 depicts the behavior of improved round robin scheduling algorithm.This approach organizes the pending requests in three queues (Q_1, Q_2 and Q_3). The improved round robin scheduling algorithm adopts cycle of four processes (P_i, P_j, P_k and P_w) for the purpose of sequential allocation to scheduler; it starts with two processes from Q_1 followed by one process from $Q_2$and one process from Q_3 (waiting process).

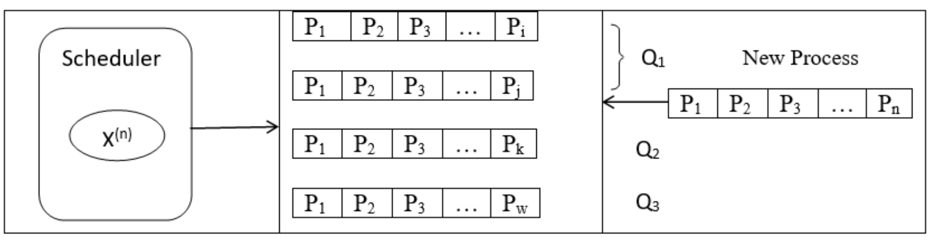


Figure 1.1: Generalized Markov chain models in CPU scheduling [7].

2. Related Work

Designing of an effective and efficient CPU scheduling algorithm is always an area of interest for various researchers. So, many enhancements in traditional round robin CPU scheduling algorithm had been proposed till now and markov chain modeling had also been used to evaluate their performance like Shukla et al. [1] performed a linear data model-based study of improved round robin CPU scheduling algorithm with functions of shortest job first scheduling with varying time quantum. The combined study of FIFO and RR is found efficient in terms of model-based study using Markov chain model and also Shukla et al. [2] performed a general structure of transition scenario for the functioning of CPU scheduler in the presence of deadlock condition and derived an application of Markov chain model for the study of transition probabilities in data model-based approach for multilevel queue scheduling. Simulation study is performed to evaluate with the help of varying values of α and d [3]. Demar et al. [9] proposed an analysis of fair queuing algorithm and derived simulation study on particular algorithm whereas Stankovic [8] presented arrival of a process is random along with their different factors and classifications.

A new substitute of RR scheduling algorithm which is suitable for time shared systems. Mohanty et al. [17] performed study to improve the round robin algorithm and proposed a new improved round robin algorithm. Shortest Remaining Burst Round Robin by submitting the scheduler to processes with shortest remaining burst using the dynamic time quantum in the appropriate manner to traditional round robin and using some approaches to increases the performance of these scheduling algorithm. Compared this approach with the different RR scheduling algorithms Behera et al [18] proposed a multi cyclic round robin scheduling algorithm using dynamic time quantum to minimizes the number of context switches, average waiting time and average turnaround time. Some more researchers Goel et al. [16] presented a comparative study between round robin scheduling and Optimum multilevel dynamic round robin scheduling with the help of some mathematical approach. Based on the experimental analysis results are getting better then round robin scheduling algorithm in terms of turnaround time, waiting time and context switch. Few other researchers Gupta et al. [25] analyzed the round robin CPU scheduling algorithm on varying time quantum and presented comparative study on that Whereas Goel & Garg [14] proposed a preemptive scheduling algorithm. calculates dynamic time slice using Optimum multilevel dynamic round robin scheduling algorithm and generate the results after every round of execution. This algorithm process results display robustness and produced improved results as compare to traditional implemented RR scheduling algorithm. Jatav&Singhai [24] did a comparative analysis of various traditional CPU scheduling algorithm and concluded their performance behavior with respect to different scheduling criteria's.

Pandey & Vandana [6] proposed studies on existing round robin scheme to reduce the total waiting time of an any process which is spend in a ready queue and improve the performance of existing round robin algorithm to understand this waiting time phenomenon using mathematical calculation. Few researchers Abdulrahim et al. [11] enhanced a new round robin scheduling algorithm and compared the different round robin algorithms with their different factors and classifications. Proposed priority based new round robin CPU scheduling algorithm reduce the starvation problem. Whereas Mishra & Khan [12] also described an improvement in round robin through preparing a simulator program and tested improved round robin scheduling algorithm. After testing it has been found that the waiting time and turnaround time have been reduced. Li et al. [10] analyzed existing fair scheduling algorithms are either inaccurate or inefficient and non-scalable for multiprocessors to produce large scale multicore processors that solves this problem by a new scheduling algorithm called distributed weighted round robin scheduling.

Medhi [19, 20] proposed an elaborative study of a variety of stochastic processes and their applications in various fields and developed a Markov chain model for the study of uncertain rainfall phenomenon and also presented the use of stochastic process in the management of queues. Naldi [5] developed a Markov chain model for understanding the internet traffic sharing among various operators in competitive areas. Few researchers Shukla & Ojha [13] developed a data set based markov chain model is presented to study the transition states and number of scheduling schemes are designed and treated as its particular cases and are compared under the setup of markov chain model to measure the deadlock index. Performed general class of simulation study to evaluate the comparative merits of specific scheme with their terms, conditions and restrictions. Some more researchers Jain et al. [4] also 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. Jain & Jain [15] proposed work based on data model study of improved RR CPU Scheduling algorithm with features of shortest job first scheduling with varying time quantum by using Markov chain model with different data set and performed some numerical based study. Sendre et al. [7] proposed an improved round robin scheduling algorithm that reduces the average waiting time and increases the throughput and maintains the same level of CPU utilization. Authors also proposed some other ways to assign the scheduler to the next ready process. Further some random probability based numerical studies have been done to justify the proposed suggestions. Deriving a motivation from these, a class of scheduling schemes is designed in this paper for performing an integrated approach of performance comparisons under the assumption of markov chain model and using a data model approach with improved round robin PCU scheduling schemes.

3. Data Model Based General Class of Improved Round Robin Scheduling Analysis

In improved round robin scheduling policy the time requirement for completion of a process P_i after $(r_i- 1)^{th}$round is at the most one-time quantum. Therefore, we consider a priority queue Q_2 in addition to the ready queue Q_1. An additional queue has been used by Pandey et.al [6] for dispatching priority in context of FCFS scheduling. All processes, after being served by the scheduler in penultimate round, are sent to the rear end of Q_2 instead of Q_1. Thus, the processes which need only one quantum or less will be terminated in the first round itself from Q_1, while all others will be terminated on being dispatched from Q_2. Therefore, processes going to scheduler through Q_1, if not terminated, may return back to the rear end of either Q_1 or Q_2.

The scheduling policy can further be improved by adopting some different cycle. Precise idea is to appropriately choose a pair of numbers p and q (p>q) that determine the number of processes from Q_1 and Q_2 for allocation to scheduler in the cycle. An optimal choice may however, depend on the number of processes and the size of their scheduler bursts. In the present work, we shall confine our discussion to p and q. This policy provides better estimates than the conventional round robin scheduling policy in respect of all performance measures, including the throughput, without any significant increase in the overheads [11].

By imposing some restrictions and condition that can produce scheduling algorithm from above mentioned generalized IRR scheduling scheme are as follows:

- A new process can only enter from queue Q_1 and after executing the two processes P_i and P_j from Q1, scheduler immediately picks the next process from Q2 and if queue Q_2 (i.e. process P_k) is found to be empty, then another pair of processes (P_i and P_j) will be dispatched from Q_1. Scheduler comes to Q_3 only if Q_1 and Q_2 are empty.
- Scheduler can't jump to Q_3 from Q_1 without passing Q_2.
- If queue Q_1 is left with a single process, queue Q_2 will have its turn immediately after the dispatch of the single process from Q_1.
- Resting of scheduler on queue Q_3 (process P_w) only if a new process enters in Q_1, otherwise resting continues.
- If Q_1 is left with no process, Q_2 will function as a single ready queue. The transition must occur in sequence from Q_2 to Q_1, Q_1 to Q_2, Q_1 to Q_3 and then Q_2 to Q_3.

3.1Markov Chain Analysis

Let $[X^{(n)}, n \geq 1]$, be a Markov chain where $X^{(n)}$ denotes the state of the scheduling at the n^{th} quantum of time. The state space for the random variable $X^{(n)}$ is $\{Q_1, Q_2, Q_3\}$ where $Q_1 = P_i$, $P_j$are combine process in first queue, $Q_2 = P_k$is second queue and $Q_3 = P_w$ is waiting state and scheduler X move stochastically over different processing states and waiting within different quantum of time. Predefined initial selection probabilities of state are: $P[X^{(0)} = P_i] = Pr_1$; $P[X^{(0)} = P_j] = Pr_2$; $P[X^{(0)} = P_k] = Pr_3$; $P[X^{(0)} = P_w] = Pr_4$; Where $Pr_1 + Pr_2 + Pr_3 + Pr_4 = \sum_{i=1}^{4} Pri = 1$

Generalized transition state Markov chain models:

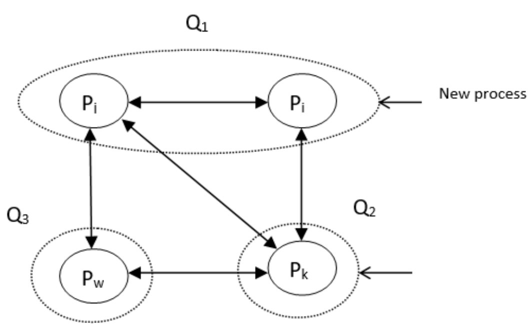


Figure 3.1.1: Generalized transition diagram [7].

Let S_{ij}(i, j = 1, 2, 3,…) be the unit step transition probabilities of scheduler over four states then transition probability depend on subject to condition: $S_{ij} = P[X^{(n)} = P_i / X^{(n-1)} = P_j]$

$$P = X^{(n-1)} \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_i & P_j & P_k & P_w \\ \hline P_i & S_{11} & S_{12} & S_{13} & S_{14} \\ P_j & S_{21} & S_{22} & S_{23} & S_{24} \\ P_k & S_{31} & S_{32} & S_{33} & S_{34} \\ P_w & S_{41} & S_{42} & S_{43} & S_{44} \end{array}$$

Predefined selection for initial probabilities of states are:

$P[X^{(n)} = P_i] = P_{r1}$; $P[X^{(n)} = P_j] = P_{r2}$;

$P[X^{(n)} = P_k] = P_{r3}$; $P[X^{(n)} = P_w] = 0$ ..… eq. 1

Let S_{ij} ($i, j = 1, 2, 3, \ldots$) be the unit step transition probabilities of scheduler over three states then transition probability depend on subject to condition:

$S_{14} = (1 - \sum_{i=1}^{3} S1i)$; $S_{24} = (1 - \sum_{i=1}^{3} S2i)$; $S_{34} = (1 - \sum_{i=1}^{3} S3i)$; $S_{44} = (1 - \sum_{i=1}^{3} S4i)$; & $0 \le S_{ij} \le 1$,

The state probabilities, after the first quantum can be obtained by a simple relationship:

$$P[X^{(1)} = P_i] = P[X^{(0)} = P_i]\, P[X^{(1)} = P_i / X^{(0)} = P_i] + P[X^{(0)} = P_j]\, P[X^{(1)} = P_i / X^{(0)} = P_j] + P[X^{(0)} = P_k]\, P[X^{(1)} = P_i / X^{(0)} = P_k] + P[X^{(0)} = P_w]\, P[X^{(1)} = P_i / X^{(0)} = P_w]$$

$$P[X^{(1)} = P_i] = \sum_{i=1}^{3} Pri\ Si1;$$
$$P[X^{(1)} = P_j] = \sum_{i=1}^{3} Pri\ Si2;$$
$$P[X^{(1)} = P_k] = \sum_{i=1}^{3} Pri\ Si3;$$
$$P[X^{(1)} = P_w] = \sum_{i=1}^{3} Pri\ Si4 \quad \text{.....eq. 2}$$

Similarly, state probabilities after second quantum can be obtained by simple relationship:

$$P[X^{(2)} = P_i] = P[X^{(1)} = P_i]\, P[X^{(2)} = P_i / X^{(1)} = P_i] + P[X^{(1)} = P_j]\, P[X^{(2)} = P_i / X^{(1)} = P_j] + P[X^{(1)} = P_k]\, P[X^{(2)} = P_i / X^{(1)} = P_k] + P[X^{(1)} = P_w]\, P[X^{(2)} = P_i / X^{(1)} = P_w]$$

$$P[X^{(2)} = P_i] = \sum_{i=1}^{4} \left(\sum_{j=1}^{3} Prj\ Sji\right) S_{i1};$$
$$P[X^{(2)} = P_j] = \sum_{i=1}^{4} \left(\sum_{j=1}^{3} Prj\ Sji\right) S_{i2};$$
$$P[X^{(2)} = P_k] = \sum_{i=1}^{4} \left(\sum_{j=1}^{3} Prj\ Sji\right) S_{i3};$$
$$P[X^{(2)} = P_w] = \sum_{i=1}^{4} \left(\sum_{j=1}^{3} Prj\ Sji\right) S_{i4} \quad \text{.....eq. 3}$$

The generalized expressions for n quantum are:

$$P[X^{(n)} = P_i] = \sum_{m=1}^{4} \ldots\ldots\ldots \sum_{l=1}^{4} \sum_{k=1}^{4} \sum_{i=1}^{4} \sum_{j=1}^{3} Prj\ Sji\ \ Sik\ \ Skl\ \ldots\ldots Sm1;$$
$$P[X^{(n)} = P_j] = \sum_{m=1}^{4} \ldots\ldots\ldots \sum_{l=1}^{4} \sum_{k=1}^{4} \sum_{i=1}^{4} \sum_{j=1}^{3} Prj\ Sji\ \ Sik\ \ Skl\ \ldots\ldots Sm2;$$
$$P[X^{(n)} = P_k] = \sum_{m=1}^{4} \ldots\ldots\ldots \sum_{l=1}^{4} \sum_{k=1}^{4} \sum_{i=1}^{4} \sum_{j=1}^{3} Prj\ Sji\ \ Sik\ \ Skl\ \ldots\ldots Sm3;$$
$$P[X^{(n)} = P_w] = \sum_{m=1}^{4} \ldots\ldots\ldots \sum_{l=1}^{4} \sum_{k=1}^{4} \sum_{i=1}^{4} \sum_{j=1}^{3} Prj\ Sji\ \ Sik\ \ Skl\ \ldots\ldots Sm4 \quad \text{.....eq. 4}$$

4. Simulation Study of Proposed Mathematical Data Model

The generalized mathematical data model is described below, by using two parameters *x* and *y*, where *i* is stands for number of queues.

$$\longleftarrow X^{(n)} \longrightarrow$$

$$P = X^{(n-1)} \updownarrow \begin{array}{c|cccc} & P_i & P_j & P_k & P_w \\ \hline P_i & x & x+y.i & x+2y.i & 1-(3x+3y.i) \\ P_j & x+y.i & x+2y.i & x+3y.i & 1-(3x+6y.i) \\ P_k & x+2y.i & x+3y.i & x+4y.i & 1-(3x+9y.i) \\ P_w & x+3y.i & x+4y.i & x+5y.i & 1-(3x+12y.i) \end{array}$$

Figure 4.1: Mathematical data model-based matrix.

Graphical analysis is performed under above mentioned generalized improved round robin scheduling scheme with different data sets. It is considered that in the probability model-based matrix various quantum values are increasing gradually. This analytical discussion on graphs about the variation $P[X^{(n)} = P_i]$ over six cases are as follows:

Case – I:

(a) when $x = 0.1$ and $y = 0.002$

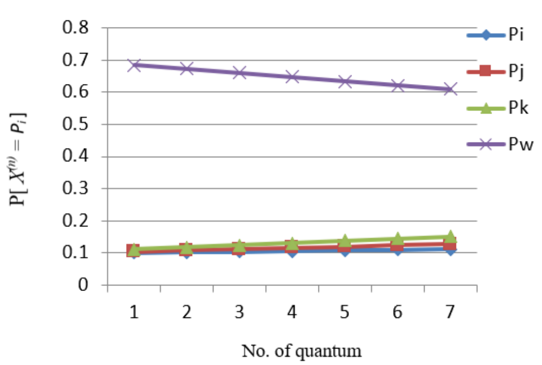


Figure 4.1.1

(b) when $x = 0.1$ and $y = 0.004$

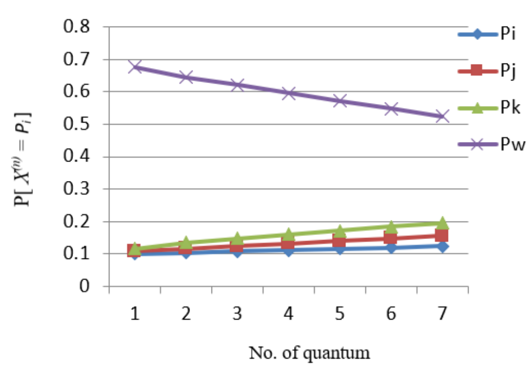


Figure 4.1.2

(c) when $x = 0.1$ and $y = 0.006$

(d) when $x = 0.1$ and $y = 0.008$

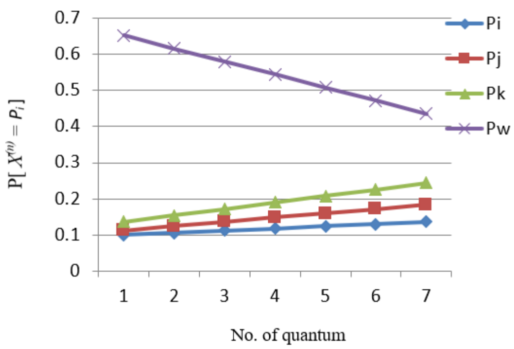


Figure 4.1.3

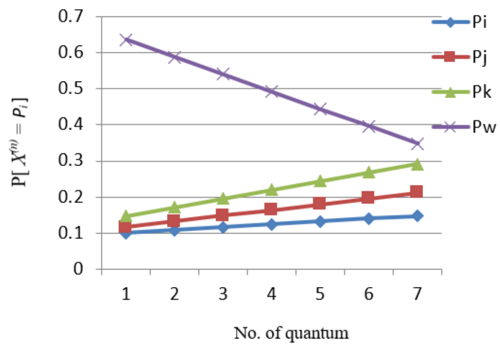


Figure 4.1.4

(e) when $x = 0.1$ and $y = 0.01$

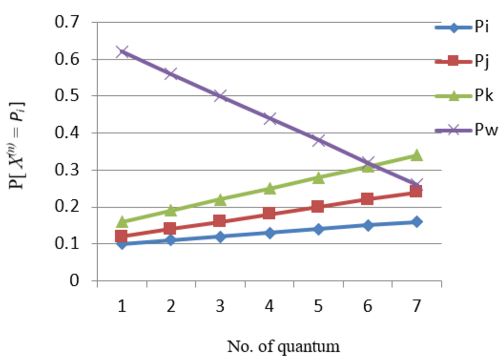
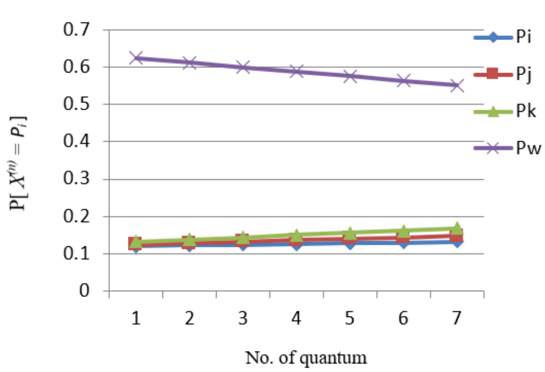


Figure 4.1.5

Remark: In case – I, it is observed that, the data analysis in these graphs are almost similar and the probability of the scheduler in the waiting state is very high as compare to other states. Since the probability of waiting state is getting very high, it means the performance of the scheduler is also degrading proportionally.

Case – II:

(a) when $x = 0.12$ and $y = 0.002$



(b) when $x = 0.12$ and $y = 0.004$

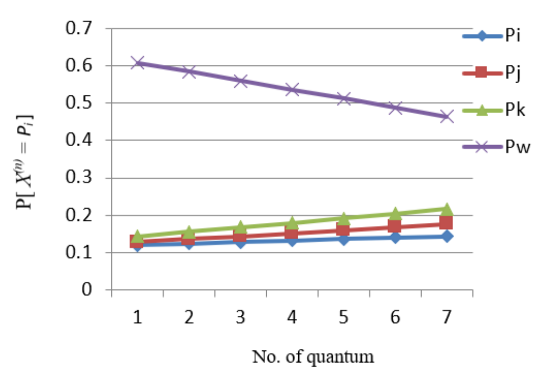


Figure 4.2.1

Figure 4.2.2

(c) when x = 0.12 and y = 0.006

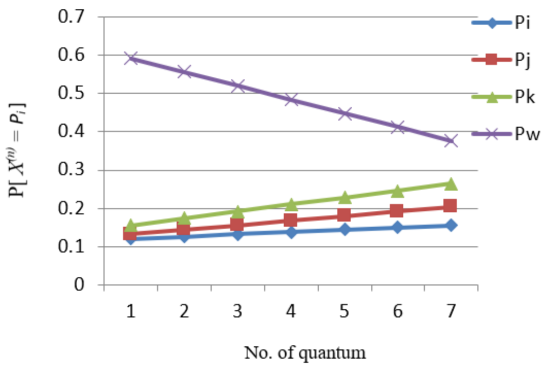


Figure 4.2.3

(d) when x = 0.12 and y = 0.008

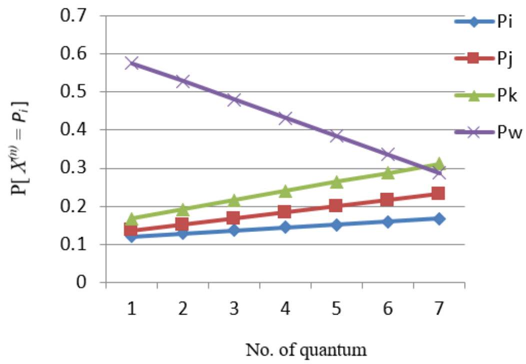


Figure 4.2.4

(e) when x = 0.12 and y = 0.01

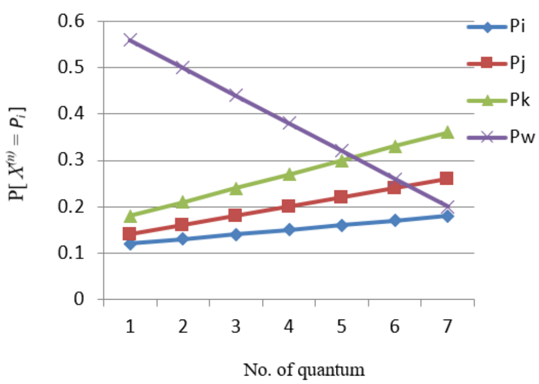


Figure 4.2.5

Remark: In case – II, we observed that, the probability of scheduler in the waiting state is same as with case – I. When x = 0.12 and with increasing value of y from 0.006 to 0.01, the graphical pattern of the transition probabilities of P_i ,P_j and P_k are similar over varying quantum. But the waiting state P_w is moving down as the quantum value increases.

Case – III:

(a) when x = 0.14 and y = 0.002

(b) when x = 0.14 and y = 0.004

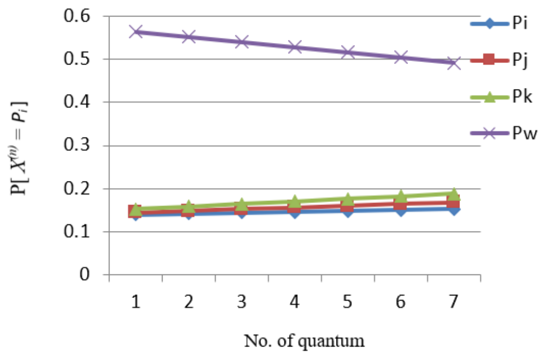


Figure 4.3.1

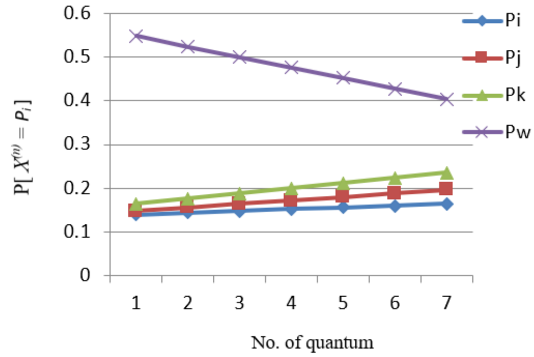


Figure 4.3.2

(c) when$x$ = 0.14 and y = 0.006

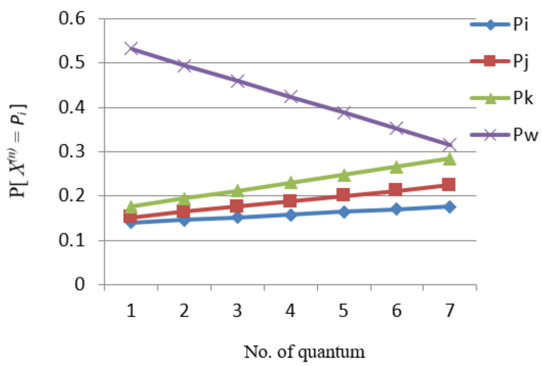


Figure 4.3.3

(d) when$x$ = 0.14 and y = 0.008

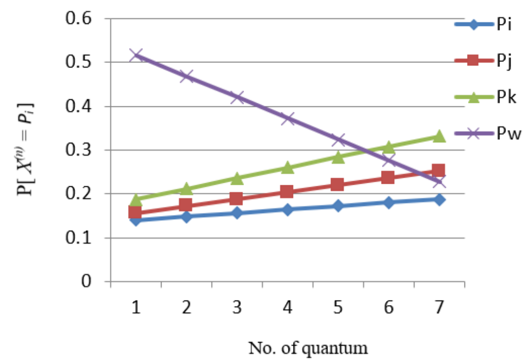


Figure 4.3.4

(e) when$x$ = 0.14 and y = 0.01

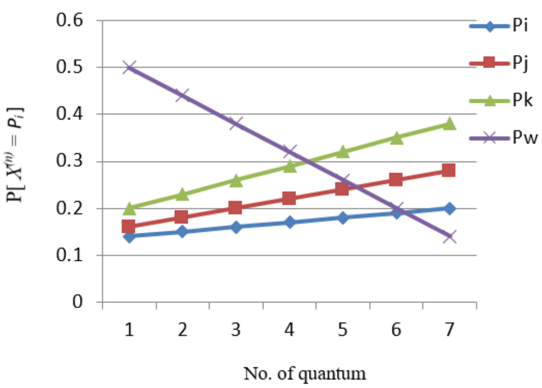


Figure 4.3.5

Remark: In case – III, when x = 0.14 and with varying values of y (0.002 – 0.006), almost all the graphical pattern in figure 4.3.1 – figure 4.3.3 remains same. This is leads to more waiting of the scheduler. This case special remark is that, when x = 0.14 and with varying values of y (0.008

and 0.01), we find that waiting state is getting down and other states are moving upward. Therefore, there are more chances for the processes P_i ,$P_j$and $P_k$to get execute without moving to waiting state.

Case – IV:

(a) whenx = 0.16 and y = 0.002

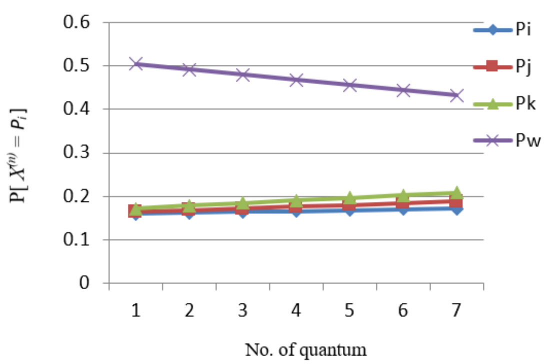


Figure 4.4.1

(b) whenx = 0.16 and y = 0.004

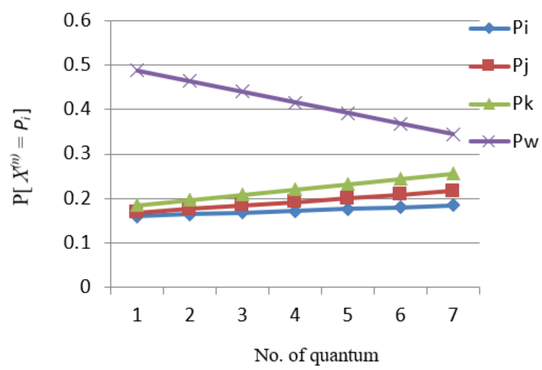


Figure 4.4.2

(c) whenx = 0.16 and y = 0.006

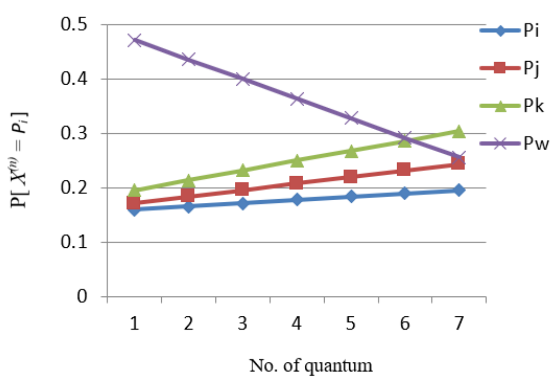


Figure 4.4.3

(d) whenx = 0.16 and y = 0.008

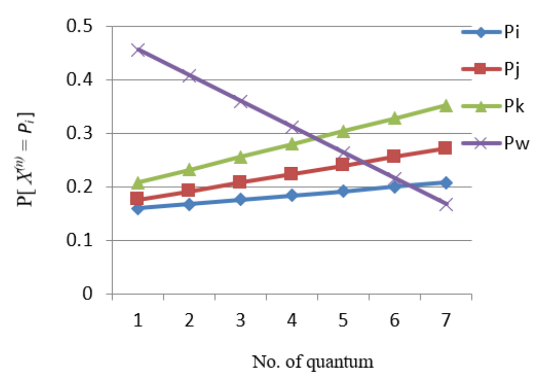


Figure 4.4.4

(e) whenx = 0.16 and y = 0.01

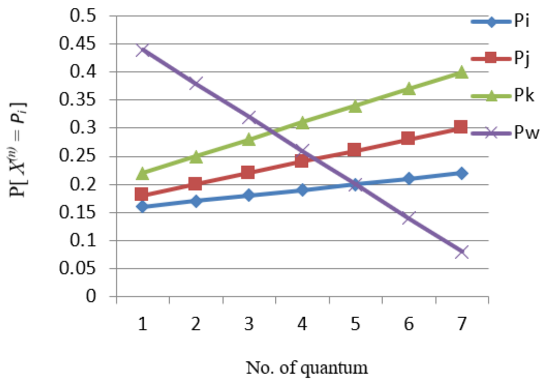


Figure 4.4.5

Remark: In case – IV, when x = 0.16 and with varying values of y (0.002 and 0.004), almost all the graphical pattern in figure 4.4.1 and figure 4.4.2 remains same. This leads to more waiting of the scheduler. Now, remarkable point is that, when x = 0.16 and with varying values of y (0.006 – 0.01), we find that waiting state is getting down and other states are moving upward. Therefore, there are more chances for the processes P_i ,$P_j$and $P_k$to get execute without moving to waiting state.

Case – V:

(a) when$x$ = 0.18 and y = 0.002

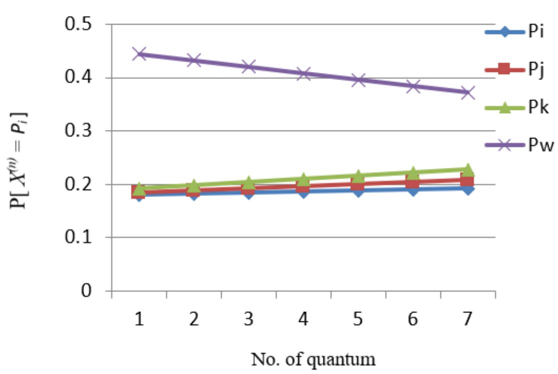


Figure 4.5.1

(b) when$x$ = 0.18 and y = 0.004

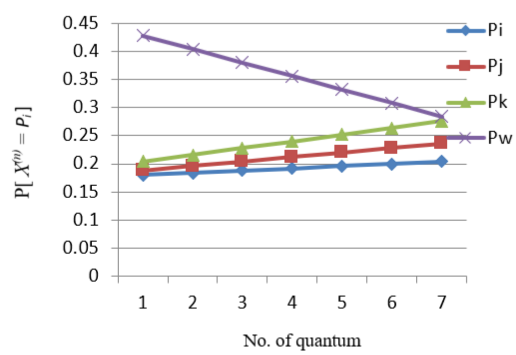


Figure 4.5.2

(c) when$x$ = 0.18 and y = 0.006

(d) when$x$ = 0.18 and y = 0.008

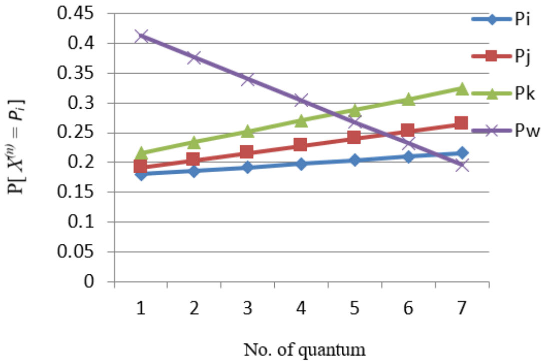


Figure 4.5.3

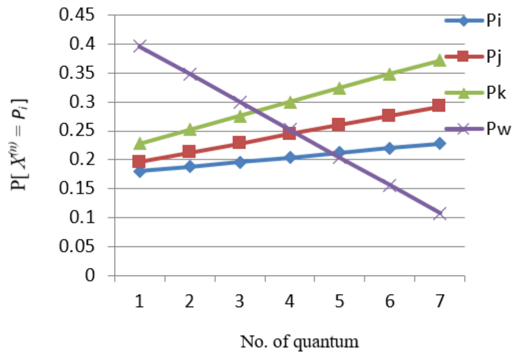


Figure 4.5.4

(e) when x = 0.18 and y = 0.01

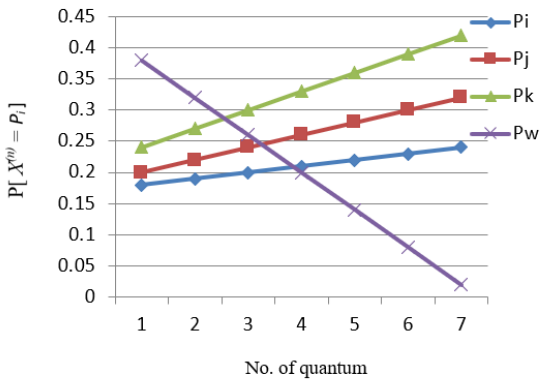


Figure 4.5.5

Remark: In case – V, the probability of the scheduler in the waiting state P_w is lower than the state P_k over varying quantum (when x = 0.18 and y = 0.004 – 0.01) which is a sign of improved performance of the scheduler. The most of the transition state P_k,P_j and P_i probabilities are almost similar in figure 4.5.1 – figure 4.5.5 and minor variation in the graphical pattern is observed. That provided more chances to job processing than the waiting situation.

Case – VI:

(a) when x = 0.2 and y = 0.002

(b) when x = 0.2 and y = 0.004

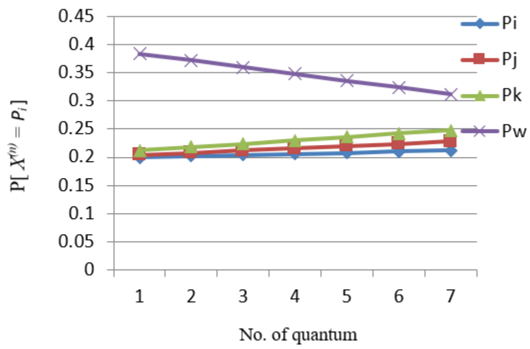


Figure 4.6.1

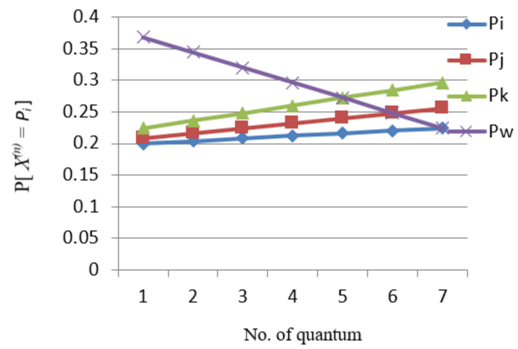


Figure 4.6.2

(c) when x = 0.2 and y = 0.006

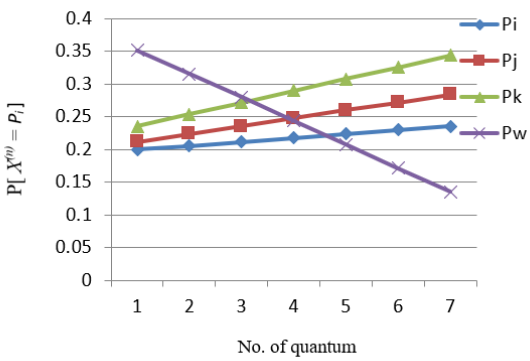


Figure 4.6.3

(d) when x = 0.2 and y = 0.008

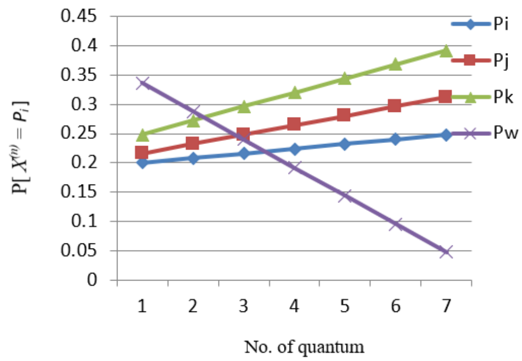


Figure 4.6.4

(e) when x = 0.2 and y = 0.01

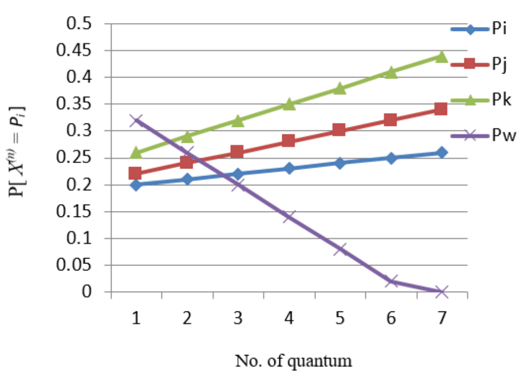


Figure 4.6.5

Remark: In case – VI, when x = 0.2 and with increasing values of y from 0.002 to 0.01, we observed the graphical pattern of the transition probabilities of P_i, P_j and P_k are moving upward. Therefore, it gives upgrading in performance of the scheduler, that means there are more chances for execution of processes P_k, P_j and P_i in order.

5. Conclusion

In this paper we evaluated the efficiency and performance of improved round robin scheduling algorithm using markov chain model under data model-based analysis for different numerical cases. We also analyzed the graphical pattern with varying quantum while having restrict transition state to observe the impact on waiting state and on the overall throughput and performance of the system. The simulation study of different graphical pattern concluded that with increasing values of y in the different specified cases, the probability of waiting state is low which shows the stability of scheduler that in turn leads to improved performance of the system. Further, we suggest that the higher combinations of *x* and *y* are the better choice for best scheduler utilization. Hence it is recommended that the operating system designer should keep this idea while designing quantum based preemptive algorithm.

References

[1] D. Shukla, S. Jain, R. Singhai& R. Agrawal, "Markov Chain Model for the Analysis of Round-Robin Scheduling Scheme", IJANA, Vol. 1, No. 1, pp. 1-7, 2009.

[2] D. Shukla, S. Ojha & S. Jain, "Data Model Approach and Markov Chain Based Analysis of Multi-Level Queue Scheduling", Journal of ACSM, Vol. 8, No. 4, pp. 50-56, 2010.

[3] D. Shukla, S. Ojha & S. Jain, "Effect of Data Model Approach in State Probability Analysis of Multi-Level Queue Scheduling", IJANA, Vol. 2, Issue 1, pp. 419-427, 2010.

[4] S. Jain, D. Shukla & R. Jain, "Linear Data Model Based Study of Improved Round Robin CPU Scheduling Algorithm", IJARCCE, Vol. 4, No. 6, pp. 562-564, 2015.

[5] M. Naldi, "Internet Access Traffic Sharing in a Multi-user Environment", Computer Networks, Vol. 38, pp. 809-824, 2002.

[6] D. Pandey & Vandana, "Improved Round Robin Policy a Mathematical Approach", IJCSE, Vol. 2, No. 4, pp. 948-954, 2010.

[7] R. Sendre, R. Singhai& S. Jain, "Markov Chain Analysis of Improved Round Robin CPU Scheduling Algorithm", IJREAM, Vol. 04, Issue 03, pp. 728-737, 2018.

[8] A.J. Stankovic, "Simulation of Three Adaptive Decentralized Controlled Task Scheduling Algorithm", Computer Networks, Vol. 8, No. 3, pp. 199-217, 1984.

[9] A. Demer, S. Keshar& S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm”, Proceedings of SIGCOMM, pp. 1-12, 1989.

[10] T. Li, D. Baumberger& S. Hahn, “Efficient and Scalable Multiprocessor Fair Scheduling using Distributed Weighted Round-Robin, ACM, 2009.

[11] A. Abdulrahim, S.E. Abdullahi & J.B. Sahalu, “A New Improved Round Robin (NIRR) CPU Scheduling Algorithm”, IJCA, Vol. 90, No. 4, pp. 27-34, 2014.

[12] M.K. Mishra, A.K. Khan, “An Improved Round Robin CPU Scheduling Algorithm”, JGRCS, Vol. 3, No. 6, pp. 64-69, 2012.

[13] D. Shukla & S. Ojha, “Deadlock Index Analysis of Multi-level Queue Scheduling in Operating System using Data Model Approach”, GESJ: CST, No. 6(29), pp. 93-110, 2010.

[14] N. Goel, & R.B. Garg, “A Simulation of an Optimum Multilevel Dynamic Round Robin Scheduling Algorithm”, IJCA, Vol. 76, No. 7, 2013.

[15] S. Jain & S. Jain, “Analysis of Multi Level Feedback Queue Scheduling using Markov Chain Model with Data Model Approach”, IJANA, Vol. 7, No. 6, pp. 2915-2924, 2016.

[16] N. Goel, A. Saxena & R.B. Garg, “An Optimal Mathematical Approach for CPU Scheduling – OMDRRS vs RR”, IJCAM, Vol. 12, pp. 65-80, 2017.

[17] R. Mohanty, H.S. Behera, K. Patwari, M.R. Dash, M. Dash &Sudhashree, “Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm”, In proceedings of ISCET, Vol. 17, 2010.

[18] H.S. Behera, R. Mohanty, S. Sahu& S.K. Bhoi, “Design and Performance Evaluation of Multi Cyclic Round Robin Algorithm using Dynamic Time Quantum”, IGRCS, Vol. 2, No. 2, pp. 48-53, 2011.

[19] J. Medhi, “Stochastic Processes”, Ed. 4, Wiley Limited (Fourth Reprint), New Delhi, 1991(a).

[20] J. Medhi, “Stochastic Models in Queuing Theory”, Academic Press Professional, Inc, San Diego, CA, 1991(b).

[21] A. Silberschatz, P.B. Galvin & G. Gagne, Operating Systems Concepts, 9th edition, John Wiley & Sons Publication, 2016.

[22] D.M. Dhamdhere, Operating Systems: A Concept Based Approach, 3rd edition, Tata McGraw-Hill Publication, 2012.

[23] A.S. Tenenbaum & A.S. Woodhull, Operating Systems: Design and Implementation, 3rd edition, PHI Publication, 2012.

[24] P.K. Jatav R. Singhai, "Comparative Study of CPU Scheduling Algorithms based on Markov Chain", IJREAM, Vol. 4, Issue 3, pp. 668-672, 2018.

[25] A.K. Gupta, N.S. Yadav & D. Goyal, "Review of Round Robin CPU Scheduling Algorithm on Varying Time Quantum", IJESI, Vol. 6, Issue 8, pp. 68-72, 2017.