International Journal of Management, IT & Engineering
Vol. 8 Issue 11, November 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 ROUND ROBIN CPU SCHEDULING ALGORITHM IN MULTIPROCESSOR ENVIRONMENT

Rupesh Sendre*

RahulSinghai**

Abstract

Multiprocessors have come out as a powerful computing medium for running real-time applications, especially where a uni-processor system would not be adequate to execute all the tasks. The accuracy and high performanceof multiprocessors have made a powerful computing resource. Such computing fremework requires a reliable algorithm to determine when and on which processor a given task should be compile in a successive manner. In multiprocessor systems, an efficient scheduling of a parallel jobs onto the processors that minimizes the entire execution time is vital for achieving a high performance. The Round Robin CPU scheduling algorithmis one of the widely used techniques for constrained optimization. Round Robin algorithm are basically preemptive algorithms based on the mechanics of quantum-based selection of jobs. As processors become available, the processesfrom the process

Keywords:
CPU Scheduling; Multiprocessor;Round Robin SchedulingAlgorithm; Markov Chain Analysis;Transition Probability Matrix.

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

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

selection list are assigned to any of the available processor in random fashion. This paper evaluates the performance of tradiitionalround robin scheduling algorithm in multiprocessor environment under markovian concepts. The overall performance in terms of varying (random and linear) numerical data are measured and then comparative analysis is performed to justify the conclusion.Further, simulation study has been performed with the help of some numerical illustrations and graphical charts.

1. Introduction

CPU Scheduling algorithms are used by dispathcer to decide which of the available ready state process should be selected next for assignment to CPU, so that resource utilization can be improved and queueing delay can be minimized.Different scheduling algorithms have been proposed by various researchers that proved efficient in uniprocessing system as well as for multiprocessing system. Multiprocessing systems provides better performance in terms of response time, throughput,waiting time &turn around time especially, where the uni processing system does not work well. Hence, it is required to analyze the performance of scheduling algorithms in multiprocessing system too.

In the proposed study, we have implemented the traditional round robin scheduling algorithm under multiprocessing environment. Since,in conventional approach used for uniprocessing system,each process shares an equal amount of processor time and the dispatcher ensures that each queue receives service time proportional to its quantum.Ifthe size of time quantum is very small then this may result in more contest switching and each process(especially I/O bound process) may not receive a fair amount of processortime.Thus, to overcome these anomalies, we implemented round robinscheduling for multiprocessors by considering that the size of the time quantum is relatively large.To demonstrate the versatility of Multiprocessor Round Robin (MPRR) scheduling, we modeled the transition of schedular over two different processor according to stochastic process. Our outcomesshow that MPRR scheduling achieves accurate proportional processing and high performance over a diverse (random or linear) data sets [7]. The following are some observations that we realized while implementing round robin scheduling algorithms for multiprossor system.

• Accuracy: Using the Markov chain model [18-19, 24-25], MPRRscheduling achieves accurate proportional fairness with low error rate, independent of the number of queues and processors in the system.

• Efficient and scalable solution: MPRRscheduling uses per processor run processes and adds low overhead to an existing operating system scheduler, even when processes dynamically arrive, depart, or change quantum.

• Flexible user control: MPRRscheduling assigns a default quantum to each queue based on its priority and provides additional support for users to specify process quantum to control transition on dispatcher.

• High performance: MPRRscheduling works in concert with existing scheduler schemes targeting other system attributes, such as latency and throughput, and thus enables high performance as well as accurate fairness. The MPRR works in concert with these features and retains high performance of the underlying dispatcher [7, 17].

2. Related Work

Designing of an effective and efficient randomly assign ready queue processing under multiprocessor environment is always an area of interest for various researchers. So, many enhancements in various CPU scheduling algorithm had been proposed to evaluate their performance like Shukla & Jain [1] performed an estimation of ready queue processing under new CPU scheduling algorithm using multiprocessor environment with varying time quantum. The combined study of lottery scheduling and systematic lottery scheduling is found efficient in terms of model-based study using some numerical illustration and Shukla et al. [10] proposed usual lottery scheduling procedures in multiprocessor environment where variable of main interest and auxiliary information like size of the process was positively correlated and also Shukla & Jain [11] performed the size based priority scheme for the ready queue time length prediction has shown that it is better than usual lottery scheduling scheme in terms of confidence intervals and in [2], the authors describes a general estimator to estimate the functioning of ready queue processing under the multiprocessor environment and derived the confidence intervals that was calculated for comparing the efficiency of the estimate. Tam et. al. [3] suggested shared memory multiprocessors with cache sharing within a chip set. Introduced multiple processing chips and hence consist of a SMP, CMP and SMT configuration with non-uniform data sharing operating system schedulers. Levin et. al. [4] developed DP Fair scheduling policy and examined how it short out the problem of greedy scheduling algorithms. Bertogna&Cirinei [5] proposed a new approach for the analysis of real time systems globally scheduled on a symmetric multiprocessor platform. The effectiveness of the analysis has been extensively proved through mathematical formulation and their numerical illustration.

Fairness is a basic requirement of any operating system CPU scheduler. Existing round robin scheduling algorithms are either inaccurate or inefficient and non-scalable for multiprocessors. This problem is becoming increasingly severe as the designer to produce larger scale multi-core processors. Li et al. [7] introduced the new CPU scheduling algorithm that solved the problem of multiprocessor using distributed weighted round robin scheduling, they present two vastly different scheduler designs and achieved accurate proportional processing and high performance for a diverse data. Further some mathematical formulation and numerical experimental studied had been done for justification. Fedorova et al. [8] described a new operating system scheduling algorithm that improves performance isolation on chip multiprocessors. New cache-fair algorithm ensures that the application runs as quickly as it would under the allocation of the particular thread for execution. Implementation of the algorithm in Solaris™ 10, and getting the improved performance for SPEC CPU, SPEC JBB and TPC-C. Further, some comparative study performed to justify their results. Elliott & Anderson [17] studied graphics processing units that can offer significant performance advantages over traditional CPUs. Survey on real time system that had been done to integrate GPUs into multiprocessor systems. Presented an integrated soft real time multiprocessor system shows that a GPU can achieve greater levels of system performance with the help of some mathematical formulation and their numerical illustration. Burns et. al. [9] presented an EDF-based task-splitting scheme for scheduling multiprocessor systems. Comparison had been done with two scheme and generated some numerical results to performed the maximum utilization of processor. Davis & Burns [14] performed the survey on hard real-time scheduling algorithms and schedulability analysis techniques for homogeneous multiprocessor systems. It reviewed the different scheduling methods and considers the various performance metrics that can be used for comparison. Vijayalakshmi & Padmavathi [15] proposed a comparison Between genetic algorithm and list scheduling algorithm with the help of multiprocessor task scheduling environment. Based on experimental results some studies were generated that's solve scheduling problem of multiprocessors. Li & Baruah [12] proposed the inter-processor migration for the scheduling of mixed-criticality implicit-deadline sporadic task systems on identical multiprocessor platforms. Investigated theoretical analysis as well as mathematical experiments to demonstrate their effectiveness. Cheramy et. al. [13] presented a simulator for the comparison and the understanding of real time multiprocessor scheduling policies. Generated the data sets, to perform simulations and to collect data from the experiments. Chandra et. al. [16] introduced a novel weight readjustment algorithm to translate infeasible weight assignments to a set of feasible weights. Then presented surplus fair scheduling a proportional share CPU scheduler designed for symmetric multiprocessors. Implemented scheduler in the Linux kernel and demonstrate its efficacy through an experimental evaluation.

To carry out the proposed review work some of the studies are discussed, which had been previously undertaken in the field of round robin CPU scheduling like Behera et al [27] 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. [22] 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. Some other researcher's Pandey & Vandana [20] 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. [26] 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 [23] 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.

Various authors studied a variety of stochastic processes and their applications in various fields and developed a stochastic process in the management of queues like Naldi [19] developed a Markov chain model for understanding the internet traffic sharing among various operators in competitive areas. Shukla et al. [18] 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. Some more researchers Jain et al. [21] also presented a linear data model-based study of round robin CPU scheduling algorithm with features of shortest job first scheduling with varying time quantum and Jain & Jain [25] proposed work based on data model study of 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 and also Sendre et al. [6, 24] 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. Proposed System

In uniprocessor environment process are assign to the processor in a particular fashion according to CPU scheduling algorithm, like in round robin scheduling dispatcher select the processes according to FCFS order and preempts them on the basis of time-quantum. But in case of multiprocessing environment although processes are dispatched in round robin fashion but the selection of processor to whom the process should be assigned is done dynamically. In this paper we analyzed the performance of round robin scheduling algorithm using Markov chain model under multiprocessing environment by considering two processor P_1 and P_2 each having large number of processes that are assigned randomly. There are two states namely R and B, that shows the resting and busy condition of both the processors respectively. The following assumption are considered to model the allocation of varies processes among P_1 and P_2:

- All the ready processes reside in ready queue and CPU scheduler select the processes in round robin fashion and assign them to any of the available processor randomly.
- Any process may initially assign to either of the processor (indicated as P_1 and P_2) with probability Pr_1 and Pr_2 ($\sum_{i=1}^{2} Pri = 1$).
- Initially process is assigned to any of the processor Pi (i = 1, 2) with a dynamic mechanism and fix a timer to interrupt on completion of certain predefined time-quantum.
- Process hold the processor until the time quantum is over and on completion of time quantum if the processes is still pending, then it again joins the ready queue from the tail. If it completes its executionsucessfully then it goes out from Pi.
- This processor allotment to various processes continues until the ready queue becomes empty.
- The same processor may be assigned again and again for various processes or they may be assigned in an alternative way (P_1, P_2, P_1,…).
- When both the processors are free then control moves to resting states and when both the processors are busy in execution their control moves to busy states.
- If both the processors are availabe and some new processes joins ready queue then they may be assigned to any of the processors hence state will again be changed from R to either P_1 or P_2.
- When the processors get free then its states changed from B to either P_1 or P_2.

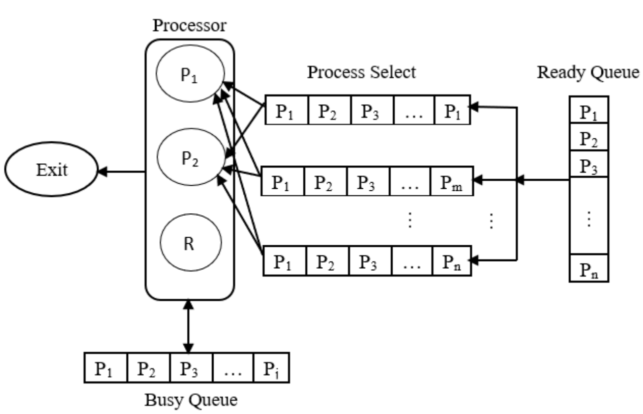


Figure 3.1: Generalized Markov chain models in Multiprocessor environment.

3.1Markov Chain Analysis

Let [$X^{(n)}$, $n \geq 1$], be a Markov chain where $X^{(n)}$ denotes the state of the processor at the n^{th} quantum of time. The state space for the random variable $X^{(n)}$ is {P_1, P_2, B, R} where Pi(P_1or P_2)are two processorseach having large number of processes that are assigned randomly. There are two states namely R and B, that shows the resting and busy condition of both the processors respectively.Predefined initial selection probabilities of state are: $P[X^{(0)} = P_1] = Pr_1$; $P[X^{(0)} = P_2] = Pr_2$; $P[X^{(0)} = B] = Pr_3$; $P[X^{(0)} = R] = Pr_4$; Where $Pr_1 + Pr_2 + Pr_3 + Pr_4 = \sum_{i=1}^{4} Pri = 1$.

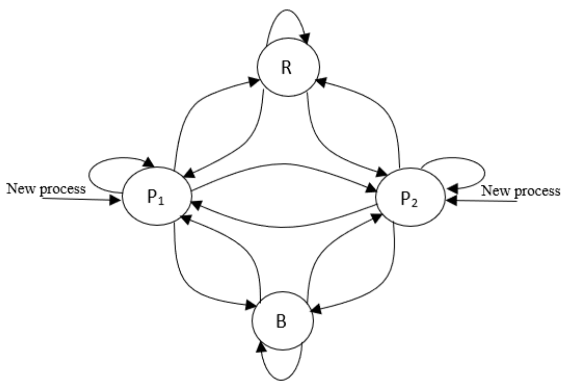


Figure 3.2: Generalized transition probability model.

Under there assumption the behavior of processors and action of scheduler can be modeled by core state discrete time markov chain (fig. 3.2) in which transition probabilities are represented by an edge connecting the circulars indicating the different chain states and the time is indicated by number of attempts.

Thus, the initial condition before the first selection of processors are:

$P[X^{(0)} = P_1] = P_{r1}$; $P[X^{(0)} = P_2] = P_{r2}$; $P[X^{(0)} = B] = 0$; $P[X^{(0)} = R] = 1 - \sum_{i=1}^{2} Pri$. …… eq. 1

Therefore, the one step transition probabilities matrix over two processors is as follows:

$$P = X^{(n-1)} \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & T_{11} & T_{12} & T_{13} & T_{14} \\ P_2 & T_{21} & T_{22} & T_{23} & T_{24} \\ B & T_{31} & T_{32} & T_{33} & 0 \\ R & T_{41} & T_{42} & 0 & T_{44} \end{array}$$

Let T_{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 following condition:

$T_{14} = (1 - \sum_{i=1}^{3} T1i)$; $T_{24} = (1 - \sum_{i=1}^{3} T2i)$; $T_{34} = (1 - \sum_{i=1}^{3} T3i)$; $T_{44} = (1 - \sum_{i=1}^{3} T4i)$; & $0 \leq T_{ij} \leq 1$,

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

$P[X^{(1)} = P_1] = P[X^{(0)} = P_1]\, P[X^{(1)} = P_1 / X^{(0)} = P_1] + P[X^{(0)} = P_2]\, P[X^{(1)} = P_1 / X^{(0)} = P_2] + P[X^{(0)} = B]\, P[X^{(1)} = P_1 / X^{(0)} = B] + P[X^{(0)} = R]\, P[X^{(1)} = P_1 / X^{(0)} = R]$

$P[X^{(1)} = P_1] = \sum_{i=1}^{3} Pri\, Ti1$; $P[X^{(1)} = P_2] = \sum_{i=1}^{3} Pri\, Ti2$;
$P[X^{(1)} = B] = \sum_{i=1}^{3} Pri\, Ti3$; $P[X^{(1)} = R] = \sum_{i=1}^{3} Pri\, Ti4$ ……eq. 2

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

$P[X^{(2)} = P_1] = P[X^{(1)} = P_1]\, P[X^{(2)} = P_1 / X^{(1)} = P_1] + P[X^{(1)} = P_2]\, P[X^{(2)} = P_1 / X^{(1)} = P_2] + P[X^{(1)} = B]\, P[X^{(2)} = P_1 / X^{(1)} = B] + P[X^{(1)} = R]\, P[X^{(2)} = P_1 / X^{(1)} = R]$

$P[X^{(2)} = P_1] = \sum_{i=1}^{4} (\sum_{j=1}^{3} Prj\, Tji)\, T_{i1}$; $P[X^{(2)} = P_2] = \sum_{i=1}^{4} (\sum_{j=1}^{3} Prj\, Tji)\, T_{i2}$;
$P[X^{(2)} = B] = \sum_{i=1}^{4} (\sum_{j=1}^{3} Prj\, Tji)\, T_{i3}$; $P[X^{(2)} = R] = \sum_{i=1}^{4} (\sum_{j=1}^{3} Prj\, Tji)\, T_{i4}$……eq. 3

The generalized expressions for n quantum are:

P [$X^{(n)} = P_1$] = $\sum_{m=1}^{4}$ .... $\sum_{l=1}^{4}$ $\sum_{k=1}^{4}$ $\sum_{i=1}^{4}$ $\sum_{j=1}^{3} Prj\ Tji\ Tik\ Tkl\ \ldots..Tm1$;

P [$X^{(n)} = P_2$] = $\sum_{m=1}^{4}$ .... $\sum_{l=1}^{4}$ $\sum_{k=1}^{4}$ $\sum_{i=1}^{4}$ $\sum_{j=1}^{3} Prj\ Tji\ Tik\ Tkl\ \ldots..Tm2$;

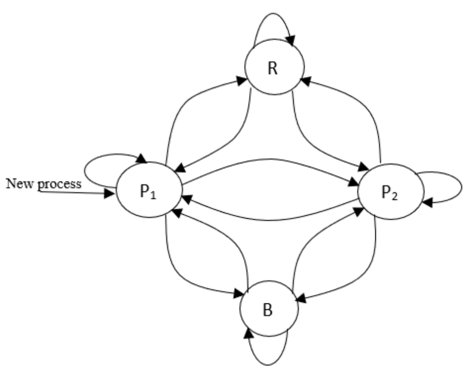
P [$X^{(n)} = B$] = $\sum_{m=1}^{4}$ .... $\sum_{l=1}^{4}$ $\sum_{k=1}^{4}$ $\sum_{i=1}^{4}$ $\sum_{j=1}^{3} Prj\ Tji\ Tik\ Tkl\ \ldots..Tm3$;

P [$X^{(n)} = R$] = $\sum_{m=1}^{4}$ .... $\sum_{l=1}^{4}$ $\sum_{k=1}^{4}$ $\sum_{i=1}^{4}$ $\sum_{j=1}^{3} Prj\ Tji\ Tik\ Tkl\ \ldots..Tm4$......eq. 4

4. RoundRobin CPU Scheduling Schemes Under Multiprocessing Environment

In this section, wehave discussed few scheduling schemes that may be produced from above mentioned generalized MPRRscheduling modelby imposing some restrictions and condition. The three schemesrealized are as follows:

4.1 Scheme - I: When process may be assigned to either the first processor P_1 or second processor P_2 only.New process joins the from ready queue from tail and the oldested process isdispatched to any of two processor P_1or $P_2$for execution. Similarly, the otherprocesses are selected from ready queue in FCFS order is assigned randomly toeither processorP_1 or P_2. Thus, the assignment of processor for processes is random.



OR

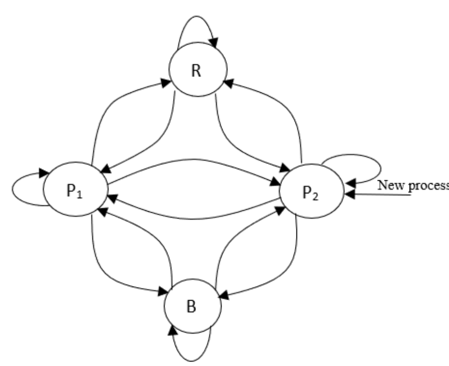


Figure 4.1: Restricted transition probability diagram

Thus, the initial probabilities under scheme-I are:

P [$X^{(0)} = P_1$] = 1 ; P [$X^{(0)} = P_2$] = 0 ; P [$X^{(0)} = B$] = 0 ; P [$X^{(0)} = R$] = 0

Unit step transaction probability matrix for $X^{(n)}$ under scheme-I is:

$$\begin{array}{cc} & \longleftarrow X^{(n)} \longrightarrow \\ P = \; X^{(n-1)} & \begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & T_{11} & T_{12} & T_{13} & T_{14} \\ P_2 & T_{21} & T_{22} & T_{23} & T_{24} \\ B & T_{31} & T_{32} & T_{33} & 0 \\ R & T_{41} & T_{42} & 0 & T_{44} \end{array} \end{array}$$

By using eq. 2 the state probabilities after the first-time quantum are:

$P[X^{(1)} = P_1] = T_{11}$; $P[X^{(1)} = P_2] = T_{12}$; $P[X^{(1)} = B] = T_{13}$; $P[X^{(1)} = R] = T_{14}$

By using eq. 3 the state probabilities after the second time quantum are:

$P[X^{(2)} = P_1] = P[X^{(1)} = P_1]\, P[X^{(2)} = P_1 / X^{(1)} = P_1] + P[X^{(1)} = P_2]\, P[X^{(2)} = P_1 / X^{(1)} = P_2] + P[X^{(1)} = B]\, P[X^{(2)} = P_1 / X^{(1)} = B] + P[X^{(1)} = R]\, P[X^{(2)} = P_1 / X^{(1)} = R]$

$P[X^{(2)} = P_1] = T_{11}T_{11} + T_{12}T_{21} + T_{13}T_{31} + T_{14}T_{41}$

$P[X^{(2)} = P_2] = P[X^{(1)} = P_1]\, P[X^{(2)} = P_2 / X^{(1)} = P_1] + P[X^{(1)} = P_2]\, P[X^{(2)} = P_2 / X^{(1)} = P_2] + P[X^{(1)} = B]\, P[X^{(2)} = P_2 / X^{(1)} = B] + P[X^{(1)} = R]\, P[X^{(2)} = P_2 / X^{(1)} = R]$

$P[X^{(2)} = P_2] = T_{12}T_{21} + T_{22}T_{22} + T_{23}T_{32} + T_{24}T_{42}$

$P[X^{(2)} = B] = P[X^{(1)} = P_1]\, P[X^{(2)} = B / X^{(1)} = P_1] + P[X^{(1)} = P_2]\, P[X^{(2)} = B / X^{(1)} = P_2] + P[X^{(1)} = B]\, P[X^{(2)} = B / X^{(1)} = B] + P[X^{(1)} = R]\, P[X^{(2)} = B / X^{(1)} = R]$

$P[X^{(2)} = B] = T_{13}T_{31} + T_{23}T_{32} + T_{33}T_{33}$

$P[X^{(2)} = R] = P[X^{(1)} = P_1]\, P[X^{(2)} = R / X^{(1)} = P_1] + P[X^{(1)} = P_2]\, P[X^{(2)} = R / X^{(1)} = P_2] + P[X^{(1)} = B]\, P[X^{(2)} = R / X^{(1)} = B] + P[X^{(1)} = R]\, P[X^{(2)} = R / X^{(1)} = R]$

$P[X^{(2)} = R] = T_{14}T_{41} + T_{24}T_{42} + T_{44}T_{44}$

Similarly, third time quantum are:

$P[X^{(3)} = P_1] = (T_{11}T_{11}+ T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{11} + (T_{12} T_{21} + T_{22}T_{22}+ T_{23} T_{32} + T_{24} T_{42}) T_{21} + (T_{13} T_{31} + T_{23} T_{32} + T_{33} T_{33}) T_{31} + (T_{14} T_{41} + T_{24} T_{42} + T_{44} T_{44}) T_{41}$

$P[X^{(3)} = P_2] = (T_{11}T_{11}+ T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{12} + (T_{12} T_{21} + T_{22}T_{22}+ T_{23} T_{32} + T_{24} T_{42}) T_{22} + (T_{13} T_{31} + T_{23} T_{32} + T_{33} T_{33}) T_{32} + (T_{14} T_{41} + T_{24} T_{42} + T_{44} T_{44}) T_{42}$

$P[X^{(3)} = B] = (T_{11}T_{11}+ T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{13} + (T_{12} T_{21} + T_{22}T_{22}+ T_{23} T_{32} + T_{24} T_{42}) T_{23} + (T_{13} T_{31} + T_{23} T_{32} + T_{33} T_{33}) T_{33}$

$P[X^{(3)} = R] = (T_{11}T_{11}+ T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{14} + (T_{12} T_{21} + T_{22}T_{22}+ T_{23} T_{32} + T_{24} T_{42}) T_{24} + (T_{14} T_{41} + T_{24} T_{42} + T_{44} T_{44}) T_{44}$

Similarly, we can find fourth, fifth and so on time quantum.

4.2 Scheme - II: When processors assigned in alternative manner (i.e. P_1,P_2, P_1, …).The following transition are restricted in this scheme:

- A new process can only enter to first processor P_1 only.
- Transition from processor P_1 to P_1 or P_2 to P_2 are restricted.

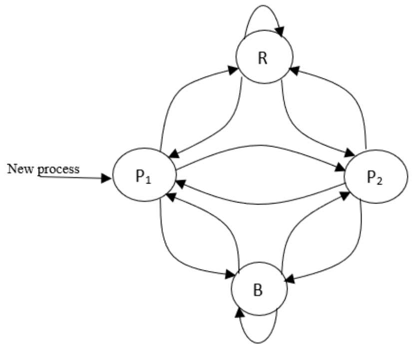


Figure 4.2: Restricted transition probability diagram.

Thus, the initial probabilities under scheme-II are:

P [$X^{(0)} = P_1$] = 1 ; P [$X^{(0)} = P_2$] = 0 ; P [$X^{(0)} = B$] = 0 ; P [$X^{(0)} = R$] = 0

Unit step transaction probability matrix for $X^{(n)}$ under scheme-II is:

$$P = X^{(n-1)} \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0 & T_{12} & T_{13} & T_{14} \\ P_2 & T_{21} & 0 & T_{23} & T_{24} \\ B & T_{31} & T_{32} & T_{33} & 0 \\ R & T_{41} & T_{42} & 0 & T_{44} \end{array}$$

By using eq. 2 the state probabilities after the first-time quantum are:

P [$X^{(2)} = P_1$] = P [$X^{(1)} = P_1$] P [$X^{(2)} = P_1 / X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = P_1 / X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = P_1 / X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = P_1 / X^{(1)} = R$]

P [$X^{(2)} = P_1$] = $T_{12}\, T_{21} + T_{13}\, T_{31} + T_{14}\, T_{41}$

P [$X^{(2)} = P_2$] =P [$X^{(1)} = P_1$] P [$X^{(2)} = P_2 / X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = P_2 / X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = P_2 / X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = P_2 / X^{(1)} = R$]

P [$X^{(2)} = P_2$] = $T_{12}\, T_{21} + T_{23}\, T_{32} + T_{24}\, T_{42}$

P [$X^{(2)} = B$]=P [$X^{(1)} = P_1$] P [$X^{(2)} = B / X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = B / X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = B / X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = B / X^{(1)} = R$]

P [$X^{(2)} = B$] = $T_{13}\, T_{31} + T_{23}\, T_{32} + T_{33} T_{33}$

P [$X^{(2)} = R$] =P [$X^{(1)} = P_1$] P [$X^{(2)} = R / X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = R / X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = R / X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = R / X^{(1)} = R$]

P [$X^{(2)} = R$] = $T_{14}\, T_{41} + T_{24}\, T_{42} + T_{44} T_{44}$

Similarly, third time quantum are:

P [$X^{(3)} = P_1$] = (T_{12} T_{21} + T_{23} T_{32} + T_{24} T_{42}) T_{21} + (T_{13} T_{31} + T_{23} T_{32} + T_{33} T_{33}) T_{31} + (T_{14} T_{41} + T_{24} T_{42} + T_{44} T_{44}) T_{41}

P [$X^{(3)} = P_2$] = (T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{12} + (T_{13} T_{31} + T_{23} T_{32} + T_{33} T_{33}) T_{32} + (T_{14} T_{41} + T_{24} T_{42} + T_{44} T_{44}) T_{42}

P [$X^{(3)} = B$] = (T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{13} + (T_{12} T_{21} + T_{23} T_{32} + T_{24} T_{42}) T_{23} + (T_{13} T_{31} + T_{23} T_{32} + T_{33} T_{33}) T_{33}

P [$X^{(3)} = R$] = (T_{12} T_{21} + T_{13} T_{31} + T_{14} T_{41}) T_{14} + (T_{12} T_{21} + T_{23} T_{32} + T_{24} T_{42}) T_{24} + (T_{14} T_{41} + T_{24} T_{42} + T_{44} T_{44}) T_{44}

Similarly, we can find fourth, fifth and so on time quantum.

4.3 Scheme - III: When some restriction is applied to control transition –

Transition from processors P_1 orP_2 to R is restricted as it is assuming that no processor can move to resting state until there exists at least one process in ready queue and if any of the processor become free then operating systems immediately assigns few processes that are currently assign to another processor.

- Transition from state R to R or B to B is restricted.

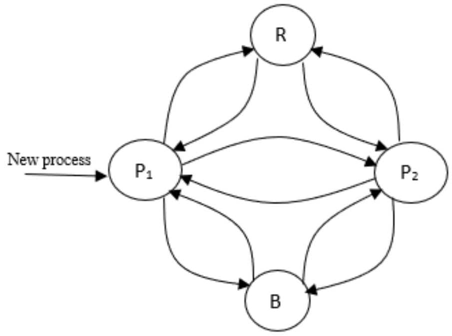


Figure 4.3: Restricted transition probability diagram.

Thus, the initial probabilities under scheme-III are:

P [$X^{(0)} = P_1$] = 1 ; P [$X^{(0)} = P_2$] = 0 ; P [$X^{(0)} = B$] = 0 ; P [$X^{(0)} = R$] = 0

Unit step transaction probability matrix for $X^{(n)}$ under scheme-3 is:

$$P = X^{(n-1)} \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0 & T_{12} & T_{13} & T_{14} \\ P_2 & T_{21} & 0 & T_{23} & T_{24} \\ B & T_{31} & T_{32} & 0 & 0 \\ R & T_{41} & T_{42} & 0 & 0 \end{array}$$

By using eq. 2 the state probabilities after the first-time quantum are:

P [$X^{(1)} = P_1$] = 0 ; P [$X^{(1)} = P_2$] = T_{12} ; P [$X^{(1)} = B$] = T_{13}; P [$X^{(1)} = R$] = T_{14}

By using eq. 3 the state probabilities after the second time quantum are:

P [$X^{(2)} = P_1$] = P [$X^{(1)} = P_1$] P [$X^{(2)} = P_1 / X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = P_1 / X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = P_1 / X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = P_1 / X^{(1)} = R$]

P [$X^{(2)} = P_1$] = $T_{12}\, T_{21} + T_{13}\, T_{31} + T_{14}\, T_{41}$

P [$X^{(2)} = P_2$] =P [$X^{(1)} = P_1$] P [$X^{(2)} = P_2/ X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = P_2/ X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = P_2/ X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = P_2/ X^{(1)} = R$]

P [$X^{(2)} = P_2$] = $T_{12}\, T_{21} + T_{23}\, T_{32} + T_{24}\, T_{42}$

P [$X^{(2)} = B$]=P [$X^{(1)} = P_1$] P [$X^{(2)} = B/ X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = B/ X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = B/ X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = B/ X^{(1)} = R$]

P [$X^{(2)} = B$] = $T_{13}\, T_{31} + T_{23}\, T_{32}$

P [$X^{(2)} = R$] =P [$X^{(1)} = P_1$] P [$X^{(2)} = R/ X^{(1)} = P_1$] + P [$X^{(1)} = P_2$] P [$X^{(2)} = R/ X^{(1)} = P_2$] + P [$X^{(1)} = B$] P [$X^{(2)} = R/ X^{(1)} = B$] + P [$X^{(1)} = R$] P [$X^{(2)} = R/ X^{(1)} = R$]

P [$X^{(2)} = R$] = $T_{14}\, T_{41} + T_{24}\, T_{42}$

Similarly, third time quantum are:

P [$X^{(3)} = P_1$] = ($T_{12}\, T_{21} + T_{23}\, T_{32} + T_{24}\, T_{42}$) T_{21} + ($T_{13}\, T_{31} + T_{23}\, T_{32}$) T_{31} + ($T_{14}\, T_{41} + T_{24}\, T_{42}$) T_{41}

P [$X^{(3)} = P_2$] = ($T_{12}\, T_{21} + T_{13}\, T_{31} + T_{14}\, T_{41}$) T_{12} + ($T_{13}\, T_{31} + T_{23}\, T_{32}$) T_{32} + ($T_{14}\, T_{41} + T_{24}\, T_{42}$) T_{42}

P [$X^{(3)} = B$] = ($T_{12}\, T_{21} + T_{13}\, T_{31} + T_{14}\, T_{41}$) T_{13} + ($T_{12}\, T_{21} + T_{23}\, T_{32} + T_{24}\, T_{42}$) T_{23}

P [$X^{(3)} = R$] = ($T_{12}\, T_{21} + T_{13}\, T_{31} + T_{14}\, T_{41}$) T_{14} + ($T_{12}\, T_{21} + T_{23}\, T_{32} + T_{24}\, T_{42}$) T_{24}

Similarly, we can find fourth, fifth and so on time quantum.

5. Simulation Study with Numerical Analysis Using Data Sets

In order to analyze three schemes mentioned in section 4.1, 4.2 and 4.3 under Markov chain model with varying quantum probability (random and linear) transition elements using different data sets are as follows:

5.1 Data set – I

Scheme I: Let initial probabilities are:$Pr_1 = 1$; $Pr_2 = 0$; $Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0.12 & 0.34 & 0.18 & 0.36 \\ P_2 & 0.21 & 0.28 & 0.30 & 0.21 \\ B & 0.35 & 0.31 & 0.34 & 0 \\ R & 0.28 & 0.32 & 0 & 0.40 \end{array} \quad \updownarrow X^{(n-1)}$$

Linear

$$P = \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0.10 & 0.10 & 0.10 & 0.70 \\ P_2 & 0.10 & 0.10 & 0.10 & 0.70 \\ B & 0.10 & 0.10 & 0.80 & 0 \\ R & 0.10 & 0.10 & 0 & 0.80 \end{array} \quad \updownarrow X^{(n-1)}$$

Table 5.1.1: The transition probabilities P [$X^{(n)} = P_i$]for random and linear cases:

Quantum No.	Random				Linear			
	P_1	P_2	B	R	P_1	P_2	B	R
n = 1	0.12	0.34	0.18	0.36	0.1	0.1	0.1	0.7
n = 2	0.25	0.307	0.185	0.259	0.1	0.1	0.1	0.7
n = 3	0.232	0.311	0.2	0.258	0.1	0.1	0.1	0.7
n = 4	0.235	0.311	0.203	0.252	0.1	0.1	0.1	0.7
n = 5	0.235	0.311	0.205	0.251	0.1	0.1	0.1	0.7
n = 6	0.236	0.311	0.205	0.25	0.1	0.1	0.1	0.7
n = 7	0.235	0.311	0.205	0.25	0.1	0.1	0.1	0.7
n = 8	0.235	0.311	0.205	0.25	0.1	0.1	0.1	0.7
n = 9	0.235	0.311	0.205	0.25	0.1	0.1	0.1	0.7
n = 10	0.235	0.311	0.205	0.25	0.1	0.1	0.1	0.7

Scheme II: Let initial probabilities are: $Pr_1 = 1$; $Pr_2 = 0$; $Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.34 & 0.18 & 0.48 \\ P_2 & 0.21 & 0 & 0.30 & 0.49 \\ B & 0.35 & 0.31 & 0.34 & 0 \\ R & 0.28 & 0.32 & 0 & 0.40 \end{array} \quad \updownarrow X^{(n-1)}$$

Linear

$$P = \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.10 & 0.10 & 0.80 \\ P_2 & 0.10 & 0 & 0.10 & 0.80 \\ B & 0.10 & 0.10 & 0.80 & 0 \\ R & 0.10 & 0.10 & 0 & 0.80 \end{array} \quad \updownarrow X^{(n-1)}$$

Table 5.1.2: The transition probabilities P [$X^{(n)} = P_i$]for random and linear cases:

Quantum No.	Random				Linear			
	P_1	P_2	B	R	P_1	P_2	B	R
n = 1	0	0.34	0.18	0.48	0	0.1	0.1	0.8
n = 2	0.269	0.209	0.163	0.359	0.1	0.09	0.09	0.72
n = 3	0.201	0.257	0.167	0.375	0.09	0.091	0.091	0.728
n = 4	0.217	0.24	0.17	0.372	0.091	0.091	0.091	0.727
n = 5	0.214	0.246	0.169	0.371	0.091	0.091	0.091	0.727
n = 6	0.215	0.244	0.17	0.372	0.091	0.091	0.091	0.727
n = 7	0.215	0.245	0.17	0.372	0.091	0.091	0.091	0.727
n = 8	0.215	0.245	0.17	0.372	0.091	0.091	0.091	0.727
n = 9	0.215	0.245	0.17	0.372	0.091	0.091	0.091	0.727
n = 10	0.215	0.245	0.17	0.372	0.091	0.091	0.091	0.727

Scheme III: Let initial probabilities are: $Pr_1 = 1$; $Pr_2 = 0$; $Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \begin{array}{c} \uparrow \\ X^{(n-1)} \\ \downarrow \end{array} \quad \begin{array}{c|cccc} \multicolumn{5}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.34 & 0.18 & 0.48 \\ P_2 & 0.21 & 0 & 0.30 & 0.49 \\ B & 0.35 & 0.65 & 0 & 0 \\ R & 0.28 & 0.72 & 0 & 0 \end{array}$$

Linear

$$P = \begin{array}{c} \uparrow \\ X^{(n-1)} \\ \downarrow \end{array} \quad \begin{array}{c|cccc} \multicolumn{5}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.10 & 0.10 & 0.80 \\ P_2 & 0.10 & 0 & 0.10 & 0.80 \\ B & 0.10 & 0.90 & 0 & 0 \\ R & 0.10 & 0.90 & 0 & 0 \end{array}$$

Table 5.1.3: The transition probabilities P [$X^{(n)} = P_i$]for random and linear cases:

Quantum No.	Random				Linear			
	P_1	P_2	B	R	P_1	P_2	B	R
n = 1	0	0.34	0.18	0.48	0	0.1	0.1	0.8
n = 2	0.269	0.463	0.102	0.167	0.1	0.81	0.01	0.08
n = 3	0.18	0.278	0.187	0.356	0.09	0.091	0.091	0.728
n = 4	0.224	0.439	0.116	0.223	0.091	0.746	0.018	0.145
n = 5	0.195	0.312	0.172	0.323	0.091	0.156	0.084	0.67
n = 6	0.216	0.411	0.129	0.246	0.091	0.688	0.025	0.198
n = 7	0.2	0.334	0.162	0.305	0.091	0.21	0.078	0.623
n = 8	0.212	0.393	0.136	0.26	0.091	0.64	0.03	0.241
n = 9	0.203	0.348	0.156	0.294	0.091	0.253	0.073	0.585
n = 10	0.21	0.382	0.141	0.268	0.091	0.601	0.034	0.275

5.2 Data set – II

Scheme I: Let initial probabilities are: $Pr_1 = 1$; $Pr_2 = 0$; $Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \begin{array}{c} \uparrow \\ X^{(n-1)} \\ \downarrow \end{array} \quad \begin{array}{c|cccc} \multicolumn{5}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0.33 & 0.24 & 0.21 & 0.22 \\ P_2 & 0.30 & 0.28 & 0.27 & 0.15 \\ B & 0.38 & 0.34 & 0.28 & 0 \\ R & 0.23 & 0.36 & 0 & 0.41 \end{array}$$

Linear

$$P = \begin{array}{c} \uparrow \\ X^{(n-1)} \\ \downarrow \end{array} \quad \begin{array}{c|cccc} \multicolumn{5}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0.20 & 0.20 & 0.20 & 0.40 \\ P_2 & 0.20 & 0.20 & 0.20 & 0.40 \\ B & 0.20 & 0.20 & 0.60 & 0 \\ R & 0.20 & 0.20 & 0 & 0.60 \end{array}$$

Table 5.2.1: The transition probabilities P [$X^{(n)} = P_i$]for random and linear cases:

Quantum No.	Random					Linear			
	P_1	P_2	B	R		P_1	P_2	B	R
n = 1	0.33	0.24	0.21	0.22		0.2	0.2	0.2	0.4
n = 2	0.311	0.297	0.193	0.199		0.2	0.2	0.2	0.4
n = 3	0.311	0.295	0.2	0.195		0.2	0.2	0.2	0.4
n = 4	0.312	0.295	0.201	0.193		0.2	0.2	0.2	0.4
n = 5	0.312	0.295	0.201	0.192		0.2	0.2	0.2	0.4
n = 6	0.312	0.295	0.201	0.192		0.2	0.2	0.2	0.4
n = 7	0.312	0.295	0.201	0.192		0.2	0.2	0.2	0.4
n = 8	0.312	0.295	0.201	0.192		0.2	0.2	0.2	0.4
n = 9	0.312	0.295	0.201	0.192		0.2	0.2	0.2	0.4
n = 10	0.312	0.295	0.201	0.192		0.2	0.2	0.2	0.4

Scheme II: Let initial probabilities are: $Pr_1 = 1$; $Pr_2 = 0$; $Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \quad \overset{\longleftarrow X^{(n)} \longrightarrow}{X^{(n-1)} \updownarrow \begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.24 & 0.21 & 0.55 \\ P_2 & 0.30 & 0 & 0.27 & 0.43 \\ B & 0.38 & 0.34 & 0.28 & 0 \\ R & 0.23 & 0.36 & 0 & 0.41 \end{array}}$$

Linear

$$P = \quad \overset{\longleftarrow X^{(n)} \longrightarrow}{X^{(n-1)} \updownarrow \begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.20 & 0.20 & 0.60 \\ P_2 & 0.20 & 0 & 0.20 & 0.60 \\ B & 0.20 & 0.20 & 0.60 & 0 \\ R & 0.20 & 0.20 & 0 & 0.60 \end{array}}$$

Table 5.2.2: The transition probabilities P [$X^{(n)} = P_i$]for random and linear cases:

Quantum No.	Random					Linear			
	P_1	P_2	B	R		P_1	P_2	B	R
n = 1	0	0.24	0.21	0.55		0	0.2	0.2	0.6
n = 2	0.278	0.269	0.124	0.329		0.2	0.16	0.16	0.48
n = 3	0.203	0.227	0.166	0.403		0.16	0.168	0.168	0.504
n = 4	0.224	0.25	0.15	0.374		0.168	0.166	0.166	0.499
n = 5	0.218	0.239	0.157	0.384		0.166	0.167	0.166	0.5
n = 6	0.22	0.244	0.154	0.38		0.167	0.166	0.166	0.5
n = 7	0.219	0.242	0.155	0.382		0.166	0.167	0.166	0.5
n = 8	0.219	0.243	0.155	0.381		0.166	0.167	0.166	0.5
n = 9	0.219	0.242	0.155	0.381		0.166	0.167	0.166	0.5
n = 10	0.219	0.242	0.155	0.381		0.166	0.167	0.166	0.5

Scheme III: Let initial probabilities are: $Pr_1 = 1$; $Pr_2 = 0$; $Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \quad \overset{\longleftarrow X^{(n)} \longrightarrow}{X^{(n-1)} \updownarrow \begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.24 & 0.21 & 0.55 \\ P_2 & 0.30 & 0 & 0.27 & 0.43 \\ B & 0.38 & 0.62 & 0 & 0 \\ R & 0.23 & 0.77 & 0 & 0 \end{array}}$$

Linear

$$P = \quad \overset{\longleftarrow X^{(n)} \longrightarrow}{X^{(n-1)} \updownarrow \begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.20 & 0.20 & 0.60 \\ P_2 & 0.20 & 0 & 0.20 & 0.60 \\ B & 0.20 & 0.80 & 0 & 0 \\ R & 0.20 & 0.80 & 0 & 0 \end{array}}$$

Table 5.2.3: The transition probabilities P [$X^{(n)} = P_i$]for random and linear cases:

Quantum No.	Random					Linear			
	P_1	P_2	B	R		P_1	P_2	B	R
n = 1	0	0.24	0.21	0.55		0	0.2	0.2	0.6
n = 2	0.278	0.554	0.065	0.103		0.2	0.64	0.04	0.12
n = 3	0.215	0.186	0.208	0.391		0.16	0.168	0.168	0.504
n = 4	0.225	0.482	0.095	0.198		0.168	0.57	0.066	0.197
n = 5	0.226	0.265	0.177	0.331		0.167	0.244	0.148	0.443
n = 6	0.223	0.419	0.119	0.238		0.167	0.506	0.082	0.247
n = 7	0.226	0.311	0.16	0.303		0.167	0.297	0.135	0.404
n = 8	0.224	0.387	0.131	0.258		0.167	0.465	0.093	0.278
n = 9	0.225	0.334	0.152	0.29		0.167	0.33	0.126	0.379
n = 10	0.225	0.372	0.137	0.267		0.167	0.437	0.1	0.298

5.3 Data set – III

Scheme I: Let initial probabilities are: $Pr_1 = 1$; $Pr_2 = 0$;$Pr_3 = 0$ and $Pr_4 = 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \begin{array}{c} \uparrow \\ X^{(n-1)} \\ \downarrow \end{array} \quad \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0.27 & 0.35 & 0.18 & 0.20 \\ P_2 & 0.40 & 0.25 & 0.18 & 0.17 \\ B & 0.45 & 0.32 & 0.23 & 0 \\ R & 0.38 & 0.40 & 0 & 0.22 \end{array}$$

Linear

$$P = \begin{array}{c} \uparrow \\ X^{(n-1)} \\ \downarrow \end{array} \quad \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ & P_1 & P_2 & B & R \\ \hline P_1 & 0.30 & 0.30 & 0.30 & 0.10 \\ P_2 & 0.30 & 0.30 & 0.30 & 0.10 \\ B & 0.30 & 0.30 & 0.40 & 0 \\ R & 0.30 & 0.30 & 0 & 0.40 \end{array}$$

Table 5.3.1: The transition probabilities P [$X^{(n)} = P_i$] for random and linear cases:

Quantum No.	Random					Linear			
	P_1	P_2	B	R		P_1	P_2	B	R
n = 1	0.27	0.35	0.18	0.2		0.3	0.3	0.3	0.1
n = 2	0.37	0.32	0.153	0.158		0.3	0.3	0.3	0.1
n = 3	0.357	0.322	0.159	0.163		0.3	0.3	0.3	0.1
n = 4	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1
n = 5	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1
n = 6	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1
n = 7	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1
n = 8	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1
n = 9	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1
n = 10	0.359	0.322	0.159	0.162		0.3	0.3	0.3	0.1

Scheme II: Let initial probabilities are: $Pr_1= 1$; $Pr_2= 0$;$Pr_3= 0$ and $Pr_4= 0$

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \; X^{(n-1)} \updownarrow \; \overset{\longleftarrow X^{(n)} \longrightarrow}{\begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.35 & 0.18 & 0.47 \\ P_2 & 0.40 & 0 & 0.18 & 0.42 \\ B & 0.45 & 0.32 & 0.23 & 0 \\ R & 0.38 & 0.40 & 0 & 0.22 \end{array}}$$

Linear

$$P = \; X^{(n-1)} \updownarrow \; \overset{\longleftarrow X^{(n)} \longrightarrow}{\begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.30 & 0.30 & 0.40 \\ P_2 & 0.30 & 0 & 0.30 & 0.40 \\ B & 0.30 & 0.30 & 0.40 & 0 \\ R & 0.30 & 0.30 & 0 & 0.40 \end{array}}$$

Table 5.3.2: The transition probabilities P [$X^{(n)} = P_i$] for random and linear cases:

Quantum No.	Random					Linear			
	P_1	P_2	B	R		P_1	P_2	B	R
n = 1	0	0.35	0.18	0.47		0	0.3	0.3	0.4
n = 2	0.4	0.246	0.104	0.25		0.3	0.21	0.21	0.28
n = 3	0.24	0.273	0.14	0.346		0.21	0.237	0.237	0.316
n = 4	0.304	0.267	0.125	0.304		0.237	0.229	0.229	0.305
n = 5	0.279	0.268	0.132	0.322		0.229	0.231	0.231	0.308
n = 6	0.289	0.269	0.129	0.314		0.231	0.23	0.23	0.307
n = 7	0.285	0.268	0.13	0.318		0.23	0.23	0.23	0.307
n = 8	0.286	0.269	0.129	0.316		0.23	0.23	0.23	0.307
n = 9	0.287	0.268	0.129	0.317		0.23	0.23	0.23	0.307
n = 10	0.287	0.268	0.129	0.317		0.23	0.23	0.23	0.307

Scheme III: Let initial probabilities are: Pr_1= 1; Pr_2= 0 ;Pr_3= 0 and Pr_4= 0

Consider data set of random and linear probabilities matrix are follows:

Random

$$P = \; X^{(n-1)} \updownarrow \; \overset{\longleftarrow X^{(n)} \longrightarrow}{\begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.35 & 0.18 & 0.47 \\ P_2 & 0.40 & 0 & 0.18 & 0.42 \\ B & 0.45 & 0.55 & 0 & 0 \\ R & 0.38 & 0.62 & 0 & 0 \end{array}}$$

Linear

$$P = \; X^{(n-1)} \updownarrow \; \overset{\longleftarrow X^{(n)} \longrightarrow}{\begin{array}{c|cccc} & P_1 & P_2 & B & R \\ \hline P_1 & 0 & 0.30 & 0.30 & 0.40 \\ P_2 & 0.30 & 0 & 0.30 & 0.40 \\ B & 0.30 & 0.70 & 0 & 0 \\ R & 0.30 & 0.70 & 0 & 0 \end{array}}$$

Table 5.3.3: The transition probabilities P [$X^{(n)} = P_i$] for random and linear cases:

Quantum No.	Random					Linear			
	P_1	P_2	B	R		P_1	P_2	B	R
n = 1	0	0.35	0.18	0.47		0	0.3	0.3	0.4
n = 2	0.4	0.39	0.063	0.147		0.3	0.49	0.09	0.12
n = 3	0.24	0.266	0.142	0.352		0.21	0.237	0.237	0.316
n = 4	0.304	0.38	0.091	0.225		0.237	0.45	0.134	0.179
n = 5	0.278	0.3	0.123	0.302		0.229	0.29	0.206	0.274
n = 6	0.29	0.352	0.104	0.257		0.231	0.405	0.156	0.208
n = 7	0.285	0.318	0.116	0.284		0.231	0.324	0.191	0.254
n = 8	0.287	0.34	0.108	0.268		0.231	0.381	0.167	0.222
n = 9	0.286	0.326	0.113	0.278		0.231	0.342	0.184	0.245
n = 10	0.287	0.334	0.11	0.271		0.231	0.37	0.172	0.229

6. Graphical Analysis

Graphical analysis is performed under above mentioned three schemes in section 4.1, 4.2 and 4.3 with different data sets in section 5.1, 5.2 and 5.3 considering random and linear probability matrix to put various quantum values. So, this analytical discussion on graphs about the variation P [$X^{(n)} = Pi$] over three data sets are as follows:

6.1 Data set – I:

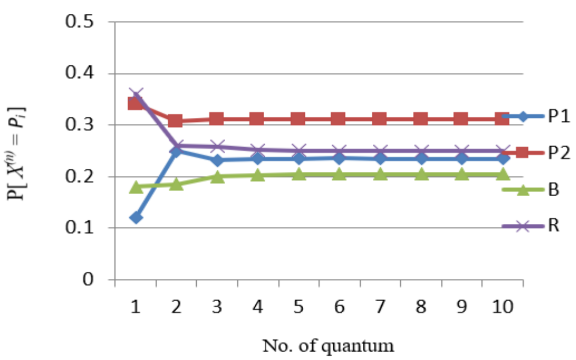


Figure 6.1.1: Scheme – I, random probability.

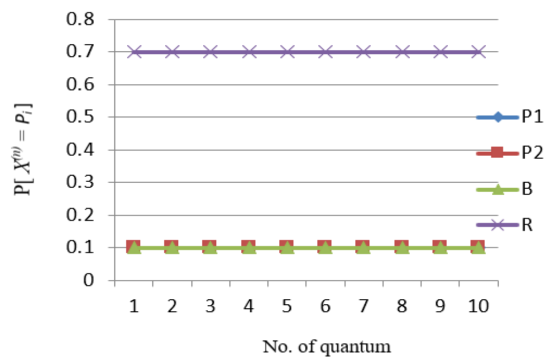


Figure 6.1.4: Scheme – I, linear probability.

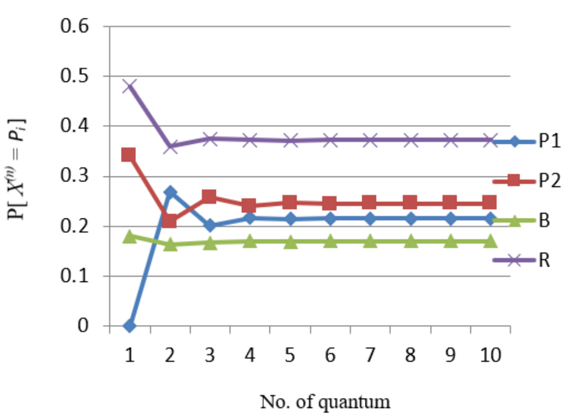


Figure 6.1.2: Scheme – II, random probability.

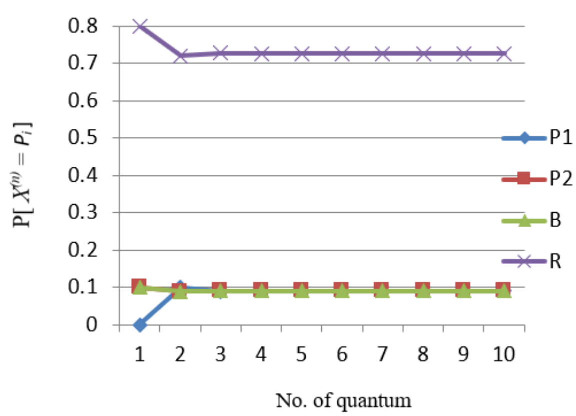


Figure 6.1.5: Scheme – II, linear probability.

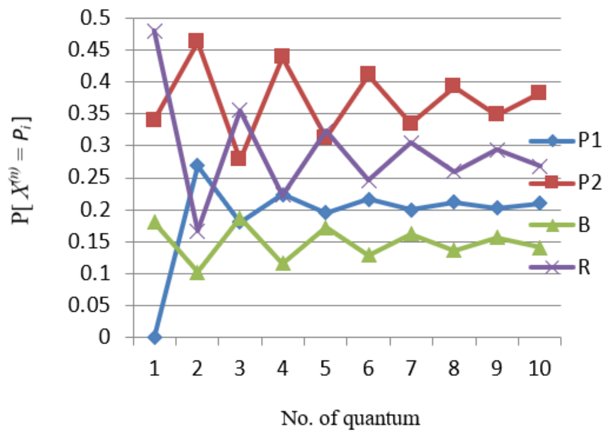


Figure 6.1.3: Scheme – III, random probability.

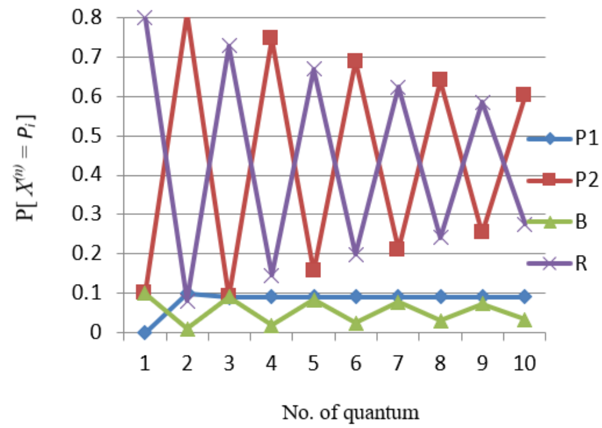


Figure 6.1.6: Scheme – III, linear probability.

Remark: In data set – I, we observed that, the data analysis in these graphs are almost similar and the probability of the dispatcher in the resting state R is very high as compare to other transition states. The special remark for this multiprocessor process scheduling is that random probability in fig 6.1.1 and fig. 6.1.3 for the state processor P_2 is little bit high as compare to resting state. It means the performance of the dispatcher is also increasing proportionally. Therefore, there are much chance for jobs assigned in state processor P_2 to be executed more rather the jobs assigned to processor P_1.

6.2 Data set – II:

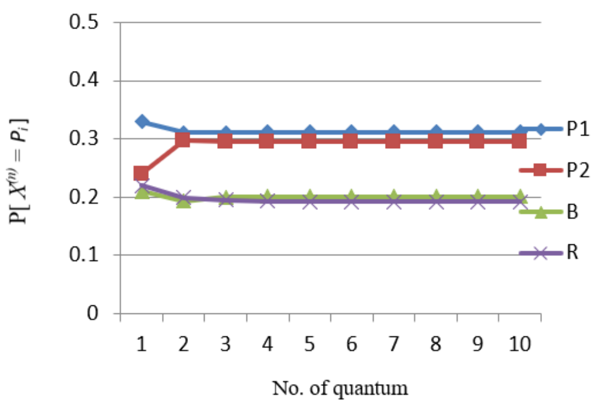


Figure 6.2.1: Scheme – I, random probability.

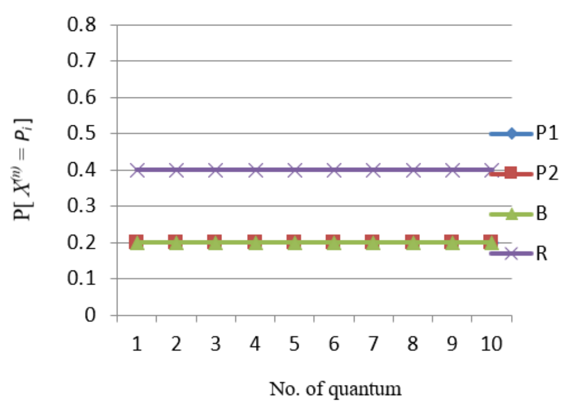


Figure 6.2.4: Scheme – I, linear probability.

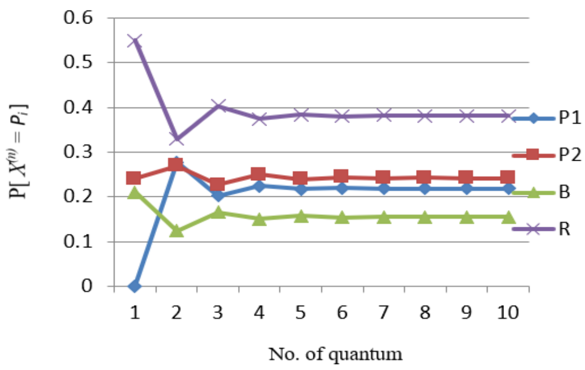


Figure 6.2.2: Scheme – II, random probability.

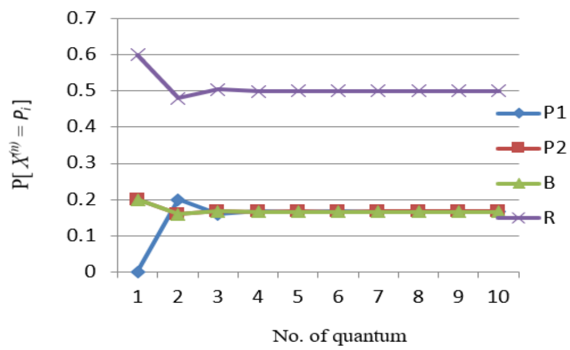


Figure 6.2.5: Scheme – II, linear probability.

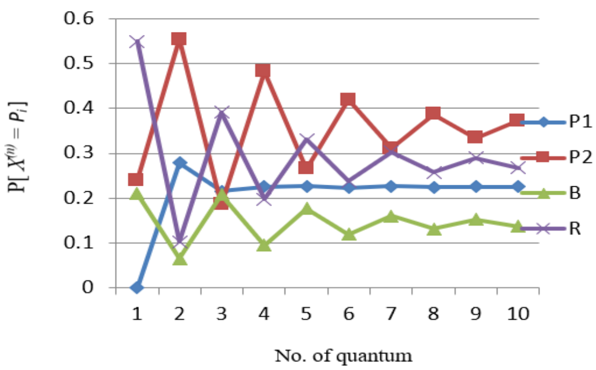


Figure 6.2.3: Scheme – III, random probability.

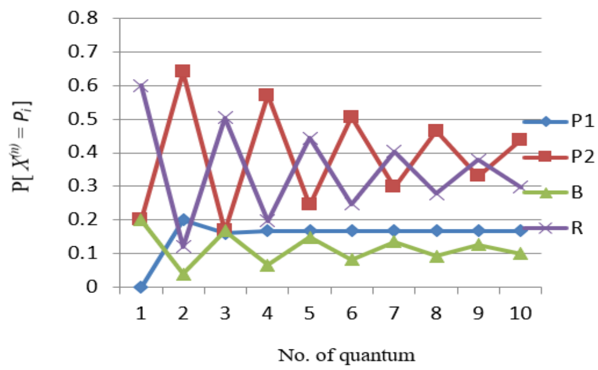


Figure 6.2.6: Scheme – III, linear probability.

Remark: In data set – II, we observed that, the data analysis in the processor statesP_1 and P_2 of the dispatcher makes stable pattern when number of quantum n >= 5 but up to n = 5 it reflects changing in graphical patterns. The remarkable point is that the probability of rest state Ris average (except fig. 6.2.4 and fig. 6.2.5) in all data sets and the probability of busy state B is comparatively low. This shows a better use of processors state and processor utilization is optimum. Therefore, less restricted scheduling scheme lead to a better use of processors time.

6.3 Data set – III:

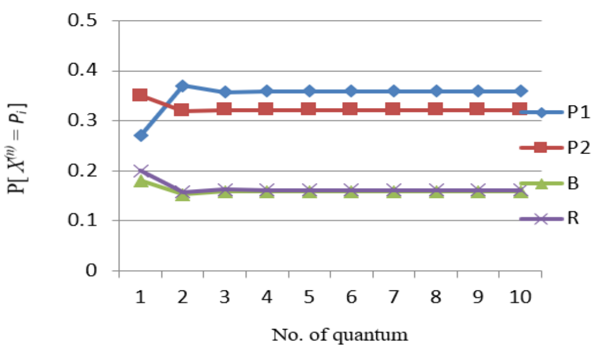


Figure 6.3.1: Scheme – I, random probability.

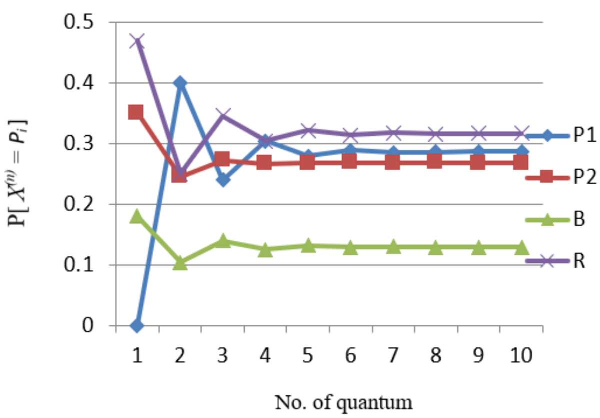


Figure 6.3.2: Scheme – II, random probability.

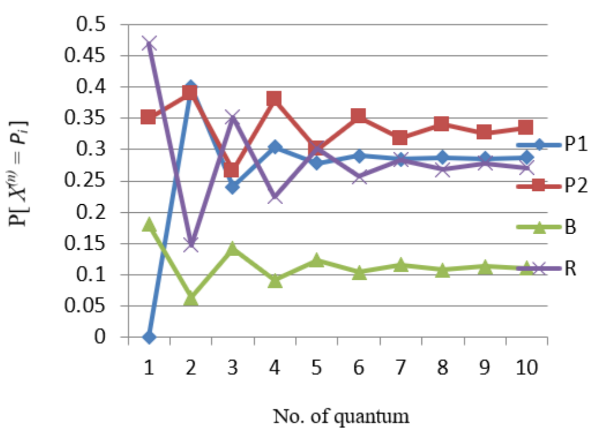


Figure 6.3.3: Scheme – III, random probability.

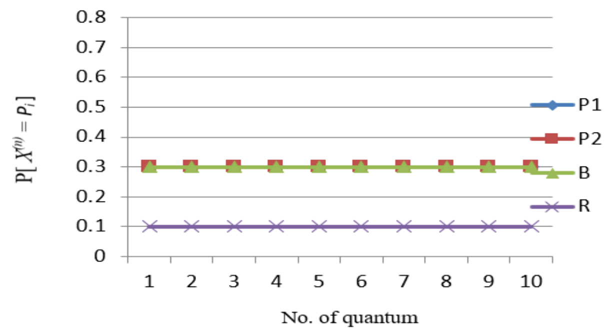


Figure 6.3.4: Scheme – I, linear probability.

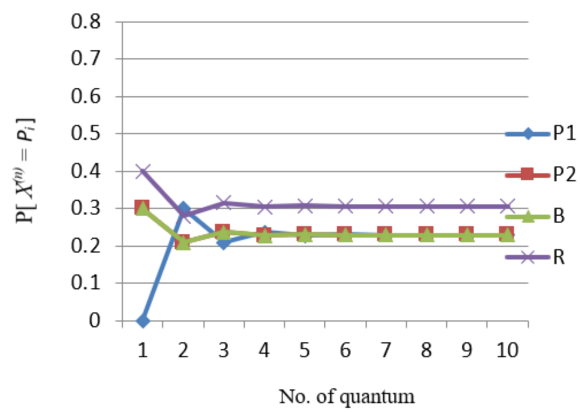


Figure 6.3.5: Scheme – II, linear probability.

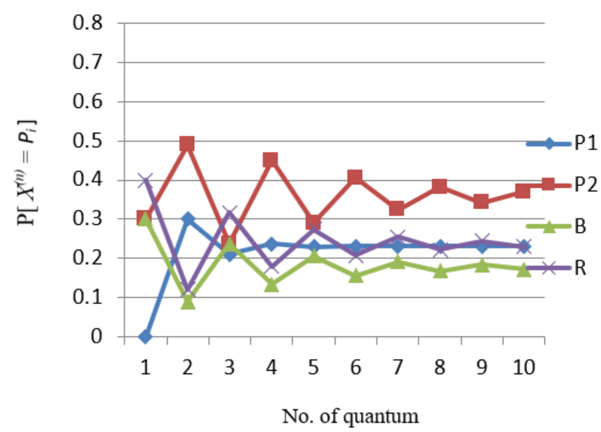


Figure 6.3.6: Scheme – III, linear probability.

Remark: In data set – III, we observed that, the random probability of processor state P_1 and P_2 are higher than the linear probability states over different quantum number which is a good sign of increase performance efficiency of the MPRRscheduling in the given data sets. But in another case of linear probability the data analysis in these graphs are varying accordingly, it means the performance of the dispatcher is not predictable. Here, there are lesser chance for jobs contained in processors state for execution. The special remark for data set – III is random probability provides more chance to job processing in multiprocessor environment than linear probability.

7. Conclusion

Efficient and optimum use of processor is a key to every operating system scheduler. Previous scheduling algorithms suffer from poor efficiency, high overhead or incompatibility with existing scheduler schemes. It is essential for operating systems to keep efficient, accurate and high-performance CPU scheduling algorithm designs. This paper proposes an adequate and high-performance analysis and comparison between three schemes of the MPRRCPU scheduling algorithm under markov chain model using varying probability matrix with number of data sets which have functions of restriction in terms of some state transition probabilities. MPRR scheduling integrates seamlessly with existing schedulers using per-processor run processes and presents a practical solution for existing operating systems. We have evaluated MPRRscheduling experimentally and numerical analytically. Using a diverse (random and linear) data set of probability ratio, our experiments demonstrate that MPRR achieves efficient, accurate and high performance over uniprocessing system. Our formal graphical analysis proves that MPRR scheduling scheme-I and scheme-III for random data sets are achieves positive stability in terms of security measure that is highly useful and recommendable to improve the performance of study. Further, we suggest that the higher transition probabilities are the better choice for best processors utilization. Hence it is recommended that the multiprocessor system designer should keep this idea while designing quantum based preemptive algorithm.

References

[1] D. Shukla & A. Jain, “Estimation of ready queue processing time under SL-Scheduling scheme in Multiprocessor Environment”, IJCSS, Vol. 4, Issue 1, pp. 74-81, 2010.

[2] D. Shukla, A. Jain & K. Verma, “Estimation of Ready Queue Processing Time using Transformed Factor-Type Estimator in Multiprocessor Environment”, IJCA, Vol. 79, No. 16, pp. 40-48, 2013.

[3] D. Tam, R. Azimi& M. Stumm, “Thread Clustering: Sharing-Aware Scheduling on SMP-CMP-SMT Multiprocessors”, ACM, pp. 47-58, 2007.

[4] G. Levin, S. Funk, C. Sadowski, I. Pye& S. Brandt, "DP-Fair: A Simple Model for Understanding Optimal Multiprocessor Scheduling", 22nd Euromicro Conference on RTS, IEEE pro., pp. 3-13, 2010.

[5] M. Bertogna& M. Cirinei, "Response-Time Analysis for Globally Scheduled Symmetric Multiprocessor Platforms", In proceedings of the Real-Time Systems Symposiun, pp. 149-158, 2007.

[6] 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.

[7] T. Li, D. Baumberger& S. Hahn, "Efficient and Scalable Multiprocessor Fair Scheduling Using Distributed Weighted Round-Robin", PPoPP'09 ACM, pp. 1-10, 2009.

[8] A. Fedorova, M. Seltzer & M.D. Smith, "Improving Performance Isolation on Chip Multiprocessors via an Operating System Scheduler", 16th International Conference PACT 2007, IEEE Computer Society, pp. 2007.

[9] A. Burns, R.I. Davis, P. Wang & F. Zhang, "Partitioned EDF Scheduling for Multiprocessors using a C=D Scheme", 18th International Conference on RTNS, France, pp. 169-178, 2010.

[10] D. Shukla, A. Jain & A. Chowdhary, "Estimation of ready queue processing time under Usual Lottery Scheduling scheme in Multiprocessor Environment", JACSM, Vol. 11, No. 11, pp. 58-63, 2011.

[11] D. Shukla & A. Jain, "Analysis of ready queue processing time under PPS-LS and SRS-LS scheme in Multiprocessor Environment", GESJ: Computer Science and Telecommunication, Vol. 33, No. 1, pp. 54-61, 2012.

[12] H. Li & S. Baruah, "Global Mixed-Criticality Scheduling on Multiprocessors", 24th Euromicro Conference on RTS, IEEE pro., pp. 166-175, 2012.

[13] M. Cheramy, P.E. Hladik& A.M. Deplanche, "SimSo: A Simulation Tool to Evaluate Real-Time Multiprocessor Scheduling Algorithm",5th International Workshop on WATERS, 2014.

[14] R.I. Davis & A. Burns, "A Survey of Hard Real-Time Scheduling for Multiprocessor Systems", ACM Computing Surveys, 2010.

[15] S.R. Vijayalakshmi & G. Padmavathi, "A Performance Study of GA and LHS in Multiprocessor Job Scheduling", IJCSI, Vol. 7, Issue 1, No. 1, pp. 37-42, 2010.

[16] A. Chandra, M. Adler, P. Goyal & P. Shenoy, "Surplus Fair Scheduling: A Proportional-Share CPU Scheduling Algorithm for Symmetric Multiprocessors", In proceeding of 4th Symposium on Operating System Design and Implementation, 2000.

[17] G.A. Elliott & J.H. Anderson, "Globally Scheduled Real-Time Multiprocessor Systems with GPUs", Work Supported NSF, ARO, AFOSR and AFRL, pp. 1-54, 2009.

[18] 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.

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

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

[21] 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.

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

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

[24] R. Sendre& R. Singhai, "Stochastic Process to Analyze Behavior of Improved Round Robin CPU Scheduling Algorithm", IJMIE, Vol. 8, Issue 9, pp. 401-418, 2018.

[25] 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.

[26] 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.

[27] 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.

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

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

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