IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Deadlock Analysis of Hybrid Lottery scheduling algorithm using Markov Chain model

Pradeep kumarJatav[1*], Rahul Singhai[2], Saurabh Jain[3]

[1]Dept. of Computer Science, PAHER University, Udaipur, India
[2]International Institute of Professional Studies, DAVV, Indore, India
[3]Shri Vaishnav Institute of Computer Application Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India

DOI: https://doi.org/10.26438/ijcse/v7i5.13011318 | Available online at: www.ijcseonline.org

Accepted: 10/May/2019, Published: 31/May/2019

Abstract- CPU scheduling defines the policy for deciding which of the available process in ready queue will be dispatched next to CPU by the scheduler; so that the resource utilization and overall performance of the system could be improved. Many traditional CPU scheduling algorithms have been proposed by several authors. Lottery scheduling is one of the well organized random based scheduling algorithms. It has random based ticket allocation algorithm in which one or more tickets are randomly assigned to each Process and when CPU becomes available the winner process is selected next for assignment. In this paper, we calculated the performance of the deadlock condition. The state transition from one process to another process is done by using Markov chain model and also data set based approach is used to study different transition states. The overall performances in terms of unequal and equal numerical data set are analyzed and then comparative analysis is performed to justify the results.

***Keyword*:-** Multiprocessing Environment ,Markov chain, CPU- scheduling, lottery scheduling, Process ,Deadlock Condition.

I. INTRODUCTION

In multiprogramming environment, many processes may be executing and requesting a restricted number of common resources at the same time. When few resources are requested by a process and if that resources are not available at that time, then the process enters into a rest state. That process will never be again able to modify state because of unavailability of required resources that are detained by other resting processes. This condition is called deadlock. Since deadlocked processes remain blocked for an infinite period of time and this condition affects user performances [1], [2], [3] hence it must be keep in mind while designing CPU scheduling algorithms.

Many reasons have been found in operating systems that are responsible for deadlock conditions.
1. Mutual exclusion: - any resources can be hold by one process at a time in a non-sharable mode.
2. Wait and hold: - processes are holding few resources and waiting for another one that is currently being held by some other waiting process in the same system.
3. No preemption: - any acquired resource cannot be preempted.
4 .Circular wait:- A closed chain of processes exists, such that each process holds one or more resources that are being requested by the next process in the chain.

The existence of all these conditions is responsible for the state of deadlock. Thus a deadlock condition may occur at any state during execution. CPU Scheduling algorithms manages allocation of different resources to process to avoid deadlock condition.

Lottery scheduling is one of the well-organized CPU scheduling algorithms in which at least one ticket is assigned to each process and the lottery scheduler draws random ticket to select the process. In lottery scheduling, when large number of the cooperating process are executing concurrently then due to occurrence of above mentioned condition none of the process may able to execute towards completion as there are some resources that are commonly used by them and may be required by other processes to complete their execution but due to unavailability of these resources all the process are reached to deadlock state. Under these situations the process is permanently blocked [4],[5],[6].[7]. In this paper lottery scheduling schemes with deadlock possibility is designed and its performance is evaluated under the two different assumptions of Markov Chain model by using equal and unequal data model approach.

II. RELATED WORK

Several researchers have analyzed the behaviors of the CPU Scheduling algorithms by using Markov chain model**.** Revelioits and Fei et al. [8] developed a novel version of the resource allocation systems to avoid the deadlock Problem and described a new decomposed operational model with new policies on different data set for particular resources . Kawadkar et al. [9] introduced deadlock as an complex condition where processes of a set of states that hold schedulers are locked for an indefinite period from access to schedulers held by other Processes within the states. No processes of the states can release its own schedulers before implementation its household tasks. so, the deadlock will last forever, unless a deadlock resolution Procedure is performed. They aslo proposed a different types of anomalies found in deadlock and described various models for particular defined deadlock condition and also provided Banker's algorithm for avoiding deadlock condition and improved the waiting state processes over there. Srinivasan & Rajaram [10] and Pandey & Vandana [11] developed a deadlock detection technique using wait for graph through propagating messages along the edges of wait for graph and also provide a deadlock resolution algorithms. Yadgiri and Jadhav [12] and Cai and Chain.[13] Presented algorithm for deadlock detection at local and global level. Some other authors have also described the deadlock detection technique which eliminates removable lock dependencies into thread specific partitions, consolidates equivalent lock dependencies and searches over the set of lock dependency, and analyzed numerical based results. Brzezinski et al.[14]. developed a hierarchy of deadlock models and deadlock detection Problems and also done a comparative study between deadlock models and deferent types of gates (OR, AND, etc.) models. Nazeem et al. [15] developed a deadlock-based study for timed Rebecca models and checked scheduler and analyzed a events-based behavior for actor's action and Predicted some experimental result for that .Y.Rose et al. [16]. described a different approaches to solve the state space explosion problem using heuristic and met heuristic algorithms and and proposed two new algorithms to finding deadlock in complex software systems and produced some experimental solution for that . Vyash and Jain.[17] developed a data set based hybrid Markov chain model for lottery scheduling. Authors also done a simulation study with various schemes to analyze these behavior. Vyash and Jain.[18] also developed a data sets based model for extensive round robin and done simulation study over different scheme. Shukla and Jain [19] proposed a K-Processing environment with different schemes, which used a random process without any replacement method . Shukla and Jain [20] Proposed and described a lottery scheduling algorithms for typical OS schedulers to improve interactive response time and also to reduce kernel lock collision, and implemented the state forward technique for LS which enabled contention over process execution rates and processor load. Singhai et al.[21] proposed and suggested a novel mechanism that provides efficient and responsive control over the relative executive execution rates of computations using lottery scheduling. Some other researchers compared and analyzed with different scheduling algorithms for Markov chain model [22],[23], [24], [25],[26],[27],[28],[29],[30]. In this paper, we have proposed a lottery based Markov chain model with deadlock state by considering two types of schemes and analyzes their performance of lottery scheduling over different data sets.

III. ASSUMPTIONS OF THE MODEL

In the proposed system, there are four processes residing in ready queue and waiting for their chance for assignment to the CPU. There is one more queue P_w (waiting/ blocked queue) where processes whose executions were suspended are residing. The selection of processes from ready queue is being done according to lottery scheduling. When a new process joins ready queue then Operating system assigns one or more lottery tickets for that process. Each process may have minimum one ticket thus by giving at least one lottery ticket to each process the operating system ensures that each process has non-zero probability of being selected during each scheduling cycle. We have also introduced a deadlock condition in the proposed system, where the execution of processes is permanently blocked. When CPU becomes available, the scheduler generates a random ticket and the process having that ticket will get the chance of execution for the assigned time quantum. If assigned process completes its execution within predefined time quantum then the process get exit from the system and new process is selected according to proposed lottery algorithm and if the assigned process is partially completed in given time quantum then on completion of time quantum process execution is stopped and new process is selected in proposed fashion so in either case random based selection is done by lottery scheduler.

The following assumptions are considered in proposed model.

1. The lottery scheduler has a random movement over all states including waiting state and deadlock state.
2. In lottery scheduling, the process whose execution is suspended due to any reason are moved to waiting state thus (waiting queue P_w).
3. The lottery scheduler picks any of the ready process with probability Pb_a. (Where a =1,2,3,4).
4. Since the selection of next process is done randomly by the lottery scheduler hence it is possible that the same process may be selected again immediately after completion of its former time quantum.
5.When any process has long waiting time (time limit exceed) then this process may move to deadlock state.

6. State P5 and P6 are denoted as waiting state and deadlock state respectively
7. The return back transition over deadlock state is not considered feasible.

The transition diagrams of proposed system under given assumptions are as follows:-

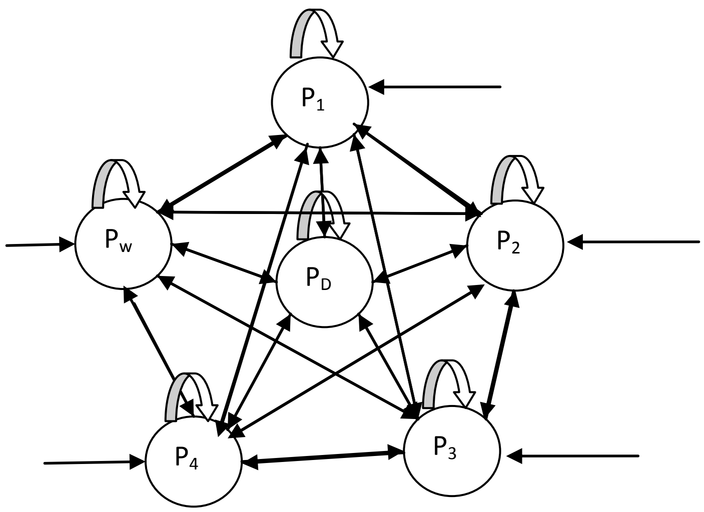


Figure:-1: Transition Probability Matrix

IV. MARKOV CHAIN MODEL

Let ($X_{(n)}$, $n \geq 1$) be a Markov chain where $X_{(n)}$ denotes the state of the lottery based scheduler at different quantum of time. The state space for the random variable $X_{(n)}$ is $\{P_1, P_2, P_3, P_4, P_w, P_D\}$ where P_5 is waiting state and P_6 is a deadlock state. The scheduler X randomly (lottery based) moves over different processes (state), waiting state and deadlock state for different quantum of time.

Predefined selections for initial Probabilities of states are:

$$P[X^{(0)} = P_1] = PB_1, P[X^{(0)} = P_2] = PB_2, P[X^{(0)} = P_3] = PB_3, P[X^{(0)} = P] = PB_4$$

$$P[X^{(0)} = P_5] = PB_5, P[X^{(0)} = P_6] = PB_6 \quad \ldots\ldots\ldots 3.1$$

With

$$PB_1 + PB_2 + PB_3 + PB_4 + PB_5 + PB_6 = \sum_{a=1}^{6} PB_a = 1 \text{ where initially } PB_5 = PB_6 = 0$$

Let $P_{a,b}$ (a,b=1,2,3,4,5,6) be the unit step transition probabilities of hybrid random based lottery scheduler over six assumed states then transition probability matrix can be given as:

$X^{(n)}$

		P_1	P_2	P_3	P_4	P_5(W)	P_6(D)
	P_1	Py_{11}	Py_{12}	Py_{13}	Py_{14}	Py_{15}	Py_{16}
	P_2	Py_{21}	Py_{22}	Py_{23}	Py_{24}	Py_{25}	Py_{26}
$X^{(n-1)}$	P_3	Py_{31}	Py_{32}	Py_{33}	Py_{34}	Py_{35}	Py_{36}
	P_4	Py_{41}	Py_{42}	Py_{43}	Py_{44}	Py_{45}	Py_{46}
	P_5(W)	Py_{51}	Py_{52}	Py_{53}	Py_{54}	Py_{55}	Py_{56}
	P_6(D)	Py_{61}	Py_{62}	Py_{63}	Py_{64}	Py_{65}	Py_{66}

Figure:-3.1: Transition Probability Matrix

lottery based transition processing for $X^{(n)}$ will be

$$P_{a,b} = P[X^{(n)}=P_a / X^{(n-1)} = P_b]$$

Unit step transition probability for waiting state W are as follow

$$\left.\begin{aligned} &P_{16} = 1-\sum_{a=1}^{6} Py_{1a},\ P_{26} = 1-\sum_{a=1}^{6} Py_{2a},\ P_{36} = 1-\sum_{a=1}^{6} Py_{3a},\\ &P_{46} = 1-\sum_{a=1}^{6} Py_{4a},\ P_{56} = 1-\sum_{a=1}^{6} Py_{5a},\ P_{66} = 1-\sum_{a=1}^{6} Py_{6a}\\ &0 \le P_{ab} \le 1 \end{aligned}\right\} \quad \ldots\ldots 3.2$$

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

$$\begin{aligned} P[X^{(1)}=P_1] = {} & P[X^{(0)}=P_1].P[X^{(1)}=P_1/ X^{(0)}=P_1] + P[X^{(0)}=P_2].P[X^{(1)}=P_1/ X^{(0)}=P_2] + \\ & P[X^{(0)}=P_3].P[X^{(1)}=P_1/ X^{(0)}=P_3] + P[X^{(0)}=P_4].P[X^{(1)}=P_1/ X^{(0)}=P_4] + \\ & P[X^{(0)}=P_5].P[X^{(1)}=P_1/ X^{(0)}=P_5] + P[X^{(0)}=P_6].P[X^{(1)}=P_1/ X^{(0)}=P_6] \end{aligned}$$

$$P[X^{(1)}=P_1] = \sum_{a=1}^{6} PB_a\, Py_{a1}$$

$$\begin{aligned} P[X^{(1)}=P_2] = {} & P[X^{(0)}=P_1].P[X^{(1)}=P_2/ X^{(0)}=P_1] + P[X^{(0)}=P_2].P[X^{(1)}=P_2/ X^{(0)}=P_2] + \\ & P[X^{(0)}=P_3].P[X^{(1)}=P_2/ X^{(0)}=P_3] + P[X^{(0)}=P_4].P[X^{(1)}=P_2/ X^{(0)}=P_4] + \\ & P[X^{(0)}=P_5].P[X^{(1)}=P_2/ X^{(0)}=P_5] + P[X^{(0)}=P_6].P[X^{(1)}=P_2/ X^{(0)}=P_6] \end{aligned}$$

$$P[X^{(1)}=P_2] = \sum_{a=1}^{6} PB_a\, Py_{a2}$$

Hence we obtained the following:

$$\left.\begin{aligned} &P[X^{(1)}=P_1] = \sum_{a=1}^{6} PB_a\, Py_{a1},\ P[X^{(1)}=P_2] = \sum_{a=1}^{6} PB_a\, Py_{a2}\\ &P[X^{(1)}=P_3] = \sum_{a=1}^{6} PB_a\, Py_{a3},\ P[X^{(1)}=P_4] = \sum_{a=1}^{6} PB_a\, Py_{a4}\\ &P[X^{(1)}=P_5] = \sum_{a=1}^{6} PB_a\, Py_{a5},\ P[X^{(1)}=P_6] = \sum_{a=1}^{6} PB_a\, Py_{a6} \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots 3.3$$

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

$$P[X(^2)=P_1] = \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_a\ Py_{ab})\}Py_{b1},\quad P[X(^2)=P_2] = \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_a\ Py_{ab})\}Py_{b2}$$

$$P[X(^2)=P_3] = \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_a\ Py_{ab})\}Py_{b3},\quad P[X(^2)=P_4] = \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_a\ Py_{ab})\}Py_{b4}$$

$$P[X(^2)=P_5] = \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_a\ Py_{ab})\}Py_{b5},\quad P[X(^2)=P_6] = \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_a\ Py_{ab})\}Py_{b6}$$

In a similar way, the generalized equations for the n^{th} quantum are:-

$$\left.\begin{array}{l}
P[X^{(n)}=P_1]=\sum_{q=1}^{6}\dots\sum_{f=1}^{6}\{\sum_{e=1}^{6}\{\sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_i\ \ Py_{ab})\}Py_{bc}\}Py_{cd}\}Py_{ef}\}Py_{f1}\dots Py_{q1} \\
P[X^{(n)}=P_2]=\sum_{q=1}^{6}\dots\sum_{f=1}^{6}\{\sum_{e=1}^{6}\{\sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_i\ \ Py_{ab})\}Py_{bc}\}Py_{cd}\}Py_{ef}\}Py_{f1}\dots Py_{q2} \\
P[X^{(n)}=P_3]=\sum_{q=1}^{6}\dots\sum_{f=1}^{6}\{\sum_{e=1}^{6}\{\sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_i\ \ Py_{ab})\}Py_{bc}\}Py_{cd}\}Py_{ef}\}Py_{f1}\dots Py_{q3} \\
P[X^{(n)}=P_4]=\sum_{q=1}^{6}\dots\sum_{f=1}^{6}\{\sum_{e=1}^{6}\{\sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_i\ \ Py_{ab})\}Py_{bc}\}Py_{cd}\}Py_{ef}\}Py_{f1}\dots Py_{q4} \\
P[X^{(n)}=P_5]=\sum_{q=1}^{6}\dots\sum_{f=1}^{6}\{\sum_{e=1}^{6}\{\sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_i\ \ Py_{ab})\}Py_{bc}\}Py_{cd}\}Py_{ef}\}Py_{f1}\dots Py_{q5} \\
P[X^{(n)}=P_6]=\sum_{q=1}^{6}\dots\sum_{f=1}^{6}\{\sum_{e=1}^{6}\{\sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB_i\ \ Py_{ab})\}Py_{bc}\}Py_{cd}\}Py_{ef}\}Py_{f1}\dots Py_{q6}
\end{array}\right\} \quad \dots 3.4$$

V. DEADLOCK ANALYSIS OF LOTTERY SCHEDULING SCHEMES.

Following Schemes can be obtained by imposing restrictions and conditions over different states in the given generalized model while considering deadlock condition :-

5.1 Scheme –I

It is assumed that initially process

1. Join the ready queue in FIFO order and lottery scheduler select the oldest process i.e. P_1 and dispatched this process to CPU where it is executed for a given period of time. and on completion of time quantum if it is partially executed then it again joins ready queue otherwise if its execution is completed successfully then the process will exit from system. in both cases the next process is selected by given lottery mechanism. Thus it is possible that same process may be selected immediately next after completion of former quantum.
2. State P_5 is waiting state. Where process execution is suspended and process is waiting for occurrence of the event or completion of the requests condition.
3 .Any process that is waiting for long time (time limit exceeded) many reach to deadlock state.
4. State P_6 is deadlock state where the execution of process cannot progress towards completion.

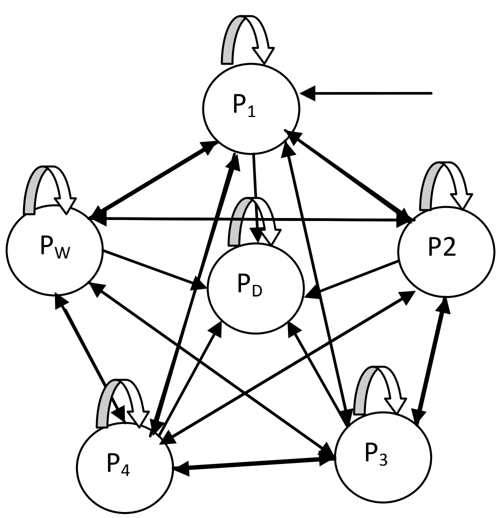


Figure:-5.1.1: Transition Diagram Scheme -I

Initial Probability for selection of processes under the scheme-I are:-

$P[X^{(0)}=P_1] = 1, P[X^{(0)}=P_2] = 0, P[X^{(0)}=P_3] = 0, P[X^{(0)}=P_4] = 0, P[X^{(0)}=P_5] = 0, P[X^{(0)}=P_6] = 0$

← $X^{(n)}$ →

		P_1	P_2	P_3	P_4	P_5 (W)	P_6 (D)
↑	P1	Py_{12}	Py_{13}	Py_{14}	Py_{15}	Py_{15}	Py_{16}
	P2	Py_{21}	Py_{22}	Py_{23}	Py_{24}	Py_{25}	Py_{26}
X(n-1)	P3	Py_{31}	Py_{32}	Py_{33}	Py_{34}	Py_{35}	Py_{36}
	P4	Py_{41}	Py_{42}	Py_{43}	Py_{44}	Py_{45}	Py_{46}
	P_5 (W)	Py_{51}	Py_{52}	Py_{53}	Py_{54}	Py_{55}	Py_{56}
↓	P_6 (D	0	0	0	0	0	1

Transition Matrix for Scheme-II

Define indicator functions $h_{a,b,}$ (a,b=1,2,3,4,5) such that

$h_{a,b}$ =0 When(a=6,b= 1,2,3,4,5, for P_6)

$h_{a,b}$ =1, Otherwise

Remark 5.1.1: - Using equation 3.3 State probability after the first quantum for scheme –I are as below.

$P[X^{(1)}=P_1] = P[X^{(0)}=P_1] .P[X^{(1)}=P_1/ X^{(0)}=P_1] + P[X^{(0)}=P_2].P[X^{(1)}=P_1/ X^{(0)}=P_2] + P[X^{(0)}=P_3].P[X^{(1)}=P_1/ X^{(0)}=P_3] + P[X^{(0)}=P_4].P[X^{(1)}=P_1/ X^{(0)}= P_4] +P[X^{(0)}=P_5].P[X^{(1)}=P_1/ X^{(0)}=P_5] + P[X^{(0)}=P_6].P[X^{(1)}=P_1/ X^{(0)}= P_6]$

$P[X^{(1)}=P_1] = Py_{11} h_{11}$

Hence for the first quantum ,we have the following outcomes

$P[X^{(1)}=P_1] = Py_{11} h_{11}$,$P[X^{(1)}=P_2] = Py_{12} h_{12}$,$P[X^{(1)}=P_3] = Py_{13} h_{13}$,$P[X^{(1)}=P_4] = Py_{14} h_{14}$,$P[X^{(1)}=P_5] = Py_{15} h_{15}$,$P[X^{(1)}=P_6] = Py_{16} h_{16}$

Remark 5.1.2: Using equation 3.4 the State probability after the first quantum for sachem –I are

Hence the generalized expression for nth quantum of scheme-I are

$P[X^{(n)}=P_1] =\sum_{q=1}^{6}\ldots \ldots \sum_{d=1}^{6}\{ \sum_{c=1}^{6}\{ \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ah1a)\}Py_{ab}\, h_{j1} \} \ldots . Py_{q1}\, h_{q1}$

$P[X^{(n)}=P_2] =\sum_{q=1}^{6}\ldots \ldots \sum_{d=1}^{6}\{ \sum_{c=1}^{6}\{ \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ah1a)\}Py_{a\,b}\, h_{b2} \} \ldots . Py_{q1}\, h_{q2}$

$P[X^{(n)}=P_3] =\sum_{q=1}^{6}\ldots \ldots \sum_{d=1}^{6}\{ \sum_{c=1}^{6}\{ \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ah1a)\}Py_{ab}\, h_{b13}\} \ldots . Py_{q1}\, h_{q3}$

$P[X^{(n)}=P_4] =\sum_{q=1}^{6}\ldots \ldots \sum_{l=1}^{6}\{ \sum_{k=1}^{6}\{ \sum_{j=1}^{6}\{\sum_{i=1}^{6}(PB1ah1a)\}Py_{ab}\, h_{b4} \} \ldots . Py_{q1}\, h_{q4}$

$P[X^{(n)}=P_5] =\sum_{q=1}^{6}\ldots \ldots \sum_{d=1}^{6}\{ \sum_{c=1}^{6}\{ \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ah1a)\}Py_{ab}\, h_{b6} \} \ldots . Py_{q1}\, h_{q5}$

$P[X^{(n)}=P_6] =\sum_{q=1}^{6}\ldots \ldots \sum_{d=1}^{6}\{ \sum_{c=1}^{6}\{ \sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ah1a)\}P_{ab}\, h_{b6} \}\ldots . Py_{q1}\, h_{q6}$

5.2 Scheme II:- It is assumed that :

1. Lottery scheduler initially always dispatched the process P_1 for the execution.
2. After completion of the time quantum lottery scheduler select next process in sequentially manner.
3. Lottery scheduler cannot reach to P_4 without passing through P_2 and p_3 in order.
4. Lottery scheduler comes to P_4 only P_1 P_2 to P_3 are not ready states thus it restricts the transition from P_2 to P_3 and p_4 how ever transition from P_4 to p_1 happens only when process P_1 is in ready state.
5. The lottery scheduler is ideal when none of the process is in ready state. Otherwise it continues in same fashion.
6. When any process has long waiting time. then process is move to deadlock state
7. The D is an Deadlock state and the transition from P_6 to P_a (a=1,2, 3, 4) is not possible.

The Transition model according to above assumption is down below.

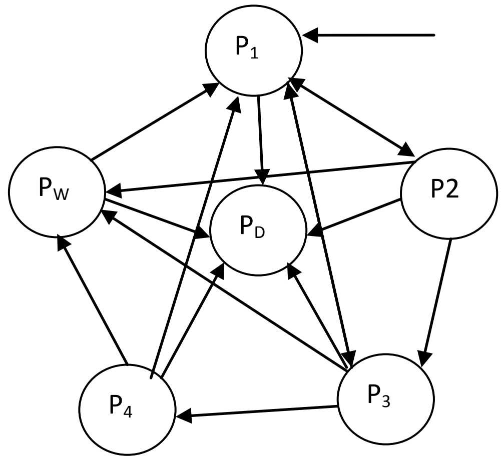


Figure:-4.2.1: Transition Diagram Scheme –II

thus the initial probability are

$P[X^{(0)}=P_1] = 1, P[X^{(0)}=P_2] = 0, P[X^{(0)}=P_3] = 0, P[X^{(0)}=P_4] = 0, P[X^{(0)}=P_5] = 0, P[X^{(0)}=P_6] = 0$

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

		P_1	P_2	P_3	P_4	P_5(W)	P_6 (D)
	P_1	0	Py_{12}	0	0	Py_{15}	Py_{16}
	P_2	Py_{21}	0	Py_{21}	0	Py_{25}	Py_{26}
$X^{(n-1)}$	P_3	Py_{31}	0	0	Py_{34}	Py_{35}	Py_{36}
	P_4	Py_{41}	0	0	0	Py_{45}	Py_{46}
	P_5(W)	Py_{51}	0	0	0	0	Py_{56}
	P_6 (D)	0	0	0	0	0	1

Transition Matrix for Scheme-II

Define an indicator function

$g_{ab} = 0$, (a=1,b=1,3,4), (a=2,b =2,4), (a=3,b=2,3)

(a=4,b=2,3,4),(a = 6,b=2,3,4,5)

(a = 6,b=1,2,3,4,5)

Otherwise $g_{ab} = 1$

The state probabilities after the first quantum are:-

$P[X^{(1)}=P_1] = P[X^{(0)}=P_1].P[X^{(1)}=P_1/ X^{(0)}=P_1] + P[X^{(0)}=P_2].P[X^{(1)}=P_1/ X^{(0)}=P_2] + P[X^{(0)}=P_3].P[X^{(1)}=P_1/ X^{(0)}=P_3] + P[X^{(0)}=P_4].P[X^{(1)}=P_1/ X^{(0)}= P_4] + P[X^{(0)}=P_5].P[X^{(1)}=P_1/ X^{(0)}=P_5] + P[X^{(0)}=P_6].P[X^{(1)}=P_1/ X^{(0)}= P_6]$

$P[X^{(1)}=P_1] = Py_{11}\, g_{11}$

Hence for the first quantum, we obtained

$P[X^{(1)}=P_1] = Py_{11}\ g_{11}$,$P[X^{(1)}=P_2] = Py_{12}\ g_{12}$,$P[X^{(1)}=P_3] = Py_{13}\ g_{13}$,$P[X^{(1)}=P_4] = Py_{14}\ g_{14}$,$P[X^{(1)}=P_5] = Py_{15}\ g_{15}$,$P[X^{(1)}=P_6] = Py_{16}\ g_{16}$

Then using (3.3) state Probability after second quantum for scheme-II are

$P[X^{(2)}=P_1] = \sum_{b=1}^{6} PB_{1a}\ g_{b1})\ (p_{yj1} g_{b1})$, $P[X^{(2)}=P_2] = \sum_{b=1}^{6} PB_{1a}\ g_{j2})\ (p_{y1} g_{b2})$

$P[X^{(2)}=P_3] = \sum_{b=1}^{6} PB_{1a}\ g_{b3})\ (Py_{b1} g_{b3})$, $P[X^{(2)}=P_4] = \sum_{b=1}^{6} PB_{1a}\ g_{b4})\ (P_{yb1} g_{b4})$

$P[X^{(2)}=P_5] = \sum_{b=1}^{6} PB_{1a}\ g_{b5})\ (Py_{b1} g_{b5})$, $P[X^{(2)}=P_6] = \sum_{b=1}^{6} PB_{1a}\ g_{b6})\ (p_{yb1} g_{b6})$

Remark 5.2.4: using (3.4) the generalized expression for n quantum of scheme-II are

$P[X^{(n)}=P_1] = \sum_{q=1}^{6} \ldots \sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ag1a)\}Py_{ab}\ g_{b1}\}Py_{b1}\ g_{b1} \ldots Py_{q1}\ g_{q1}$

$P[X^{(n)}=P_2] = \sum_{q=1}^{6} \ldots \sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ag1a)\}Py_{ab}\ g_{b1}\}Py_{b1}\ g_{b1} \ldots Py_{q1}\ g_{q1}$

$P[X^{(n)}=P_3] = \sum_{q=1}^{6} \ldots \sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ag1a)\}Py_{ab}\ g_{b1}\}Py_{b1}\ g_{b1} \ldots Py_{q1}\ g_{q1}$

$P[X^{(n)}=P_4] = \sum_{q=1}^{6} \ldots \sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ag1a)\}Py_{ab}\ g_{b1}\}Py_{b1}\ g_{b1} \ldots Py_{q1}\ g_{q1}$

$P[X^{(n)}=P_5] = \sum_{q=1}^{6} \ldots \sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ag1a)\}Py_{ab}\ g_{b1}\}Py_{b1}\ g_{b1} \ldots Py_{q1}\ g_{q1}$

$P[X^{(n)}=P_6] = \sum_{q=1}^{6} \ldots \sum_{d=1}^{6}\{\sum_{c=1}^{6}\{\sum_{b=1}^{6}\{\sum_{a=1}^{6}(PB1ag1a)\}Py_{ab} g_{b1}\}Py_{b1}\ g_{b1} \ldots Py_{q1}\ g_{q1}$

6. Simulation study

The following simulation study has performed to compare proposed two schemes mentioned in selection 4.1.1 and 4.2.1 under a uniform setup of Markov chain model with equal and unequal transition elements probability. Lets us consider the following data sets.

6.1Data Set-I

Scheme-1: Let initial probability are $PB_1 = 0.2$, $PB_2 = 0.1$, $PB_3 = 0.33$, $PB_4 = 0.1$, $PB_5 = 0.27$

The unequal and equal transition probability Matrices are given as below:-

Unequal

X (n) →

X(n-1) ↓	P1	P2	P3	P4	P5	P6
P1	0.11	0.12	0.16	0.13	0.36	0.12
P2	0.12	0.11	0.13	0.16	0.34	0.14
P3	0.1	0.09	0.12	0.13	0.46	0.1
P4	0.1	0.07	0.09	0.12	0.46	0.16
P5	0.07	0.1	0.16	0.45	0.13	0.09
P6	0	0	0	0	0	1

Equal

X (n) →

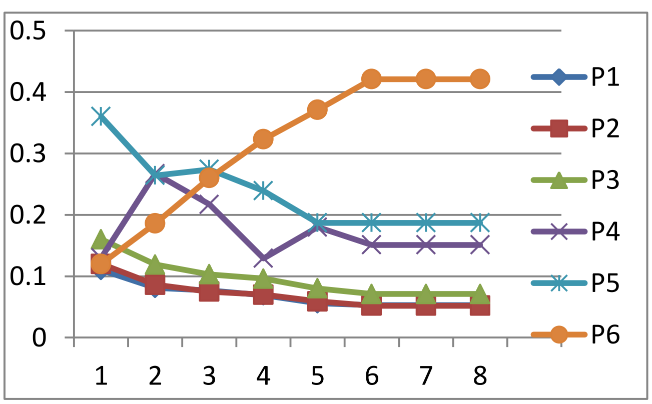
X(n-1) ↓	P1	P2	P3	P4	P5	P6
P1	0.15	0.15	0.15	0.15	0.25	.0.15
P2	0.15	0.15	0.15	0.15	0.25	0.15
P3	0.15	0.15	0.15	0.15	0.25	0.15
P4	0.15	0.15	0.15	0.15	0.25	0.15
P5	0.15	0.15	0.15	0.15	0.25	0.15
P6	0	0	0	0	0	1

	Unequal						Equal						
	P1	P2	P3	P4	P5	P6		P1	P2	P3	P4	P5	P6
N=1	0.11	0.12	0.16	0.13	0.36	0.12	N=1	0.15	0.15	0.15	0.15	0.25	0.15
N=2	0.081	0.086	0.119	0.267	0.264	0.186	N=2	0.127	0.127	0.127	0.127	0.212	0 .277
N=3	0.077	0.075	0.103	0.217	0.274	0.26	N=3	0.108	0.108	0.108	0.108	0.18	0.385
N=4	0.069	0.07	0.096	0.129	0.239	0.323	N=4	0.091	0 .091	0 .091	0.091	0.153	0.476
N=5	0.056	0.059	0.08	0.18	0.187	0.371	N=5	0.077	0.077	0.077	0.077	0.129	0.553
N=6	0.053	0.052	0.071	0.151	0.187	0.421	N=6	0.065	0 .065	0.065	0.065	0.110	0.619
N=7	0.053	0.052	0.071	0.151	0.187	0.421	N=7	0.056	0.056	0.056	0.056	0.093	0.675
N=8	0.053	0.052	0.071	0.151	0.187	0.421	N=8	0.048	0.048	0.048	0.048	0.085	0.723

Table 6.1.1 Below, The Transition Probability for unequal and equal cases.

Graphical pattern for Unequal and Equal

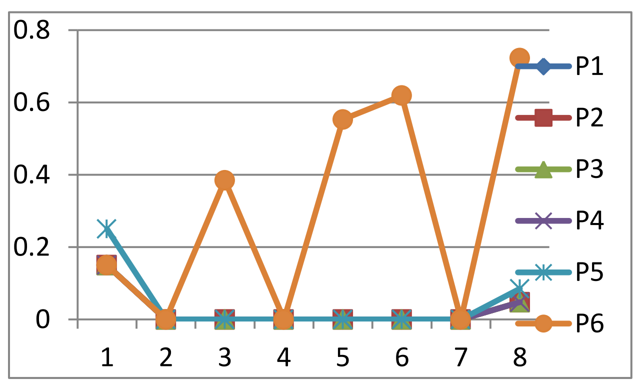
Data Set-I, Scheme-I- Unequal



No. Of quantum

Figure 6.1.1

Scheme-I- Equal



No. Of quantum

Figure 6.1.2

Unequal: - The state Probability P_1, P_2, P_3, P_4, P_6, and P_6 of the lottery scheduling makes constant pattern when number of quantum n>=3 but for two and n<3 it reflects changing in the pattern. The key point is that the probability of deadlock state P_6 is higher in this data set then the other state Probabilities as shown in figure 6.1.1, but also waiting state P_5 is very high over remaining state P_1, P_2, P_3, P_4. The Probability of scheduler in the state P_4 is a little higher value as compare to other states (P_1, P_2, P_3) over different quantum which is a sign of increasing the performance efficiency of the lottery scheduler in the data sets.
Equal: - Graphical pattern (figure 6.1.2) reveals initially higher probability of the P_6 (deadlock state) than the other states (P_1, P_2, P_3, P_4, and P_5) but after few quantum, it decreased and follow the stable pattern. and All remaining states are showing independent behavior.

Scheme-II: Let initial probability are: PB_1 =1.0, PB_2 =0.00, PB_3 =0.00, PB_4 =0.00, PB_6 =0.00.
Unequal and equal probability Matrix is follow:-

Unequal

X (n) →

X(n-1)		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0	0.30	0	0	.40	0.30
	P_2	.40	0	0.30	0	.15	0.15
X(n-1)	P_3	.40	0	0	0.15	.30	0.15
	P_4	.50	0	0	0	0.25	0.25
	P_5	.70	0	0	0	0	0.30
	P_6	0	0	0	0	0	1

Equal

X (n) →

		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0	0.15	0	0	.70	0.15
X(n-1)	P_2	0.15	0	0.15	0	.55	0.15
	P_3	0.15	0	0	0.15	.55	0.15
	P_4	0.15	0	0	0	0.70	0.15
	P_5	0.15	0	0	0	0	0.85
	P_6	0	0	0	0	0	1

Table 6.1.1 Below, The Transition probability for unequal and equal cases.
Graphical pattern for Unequal and Equal

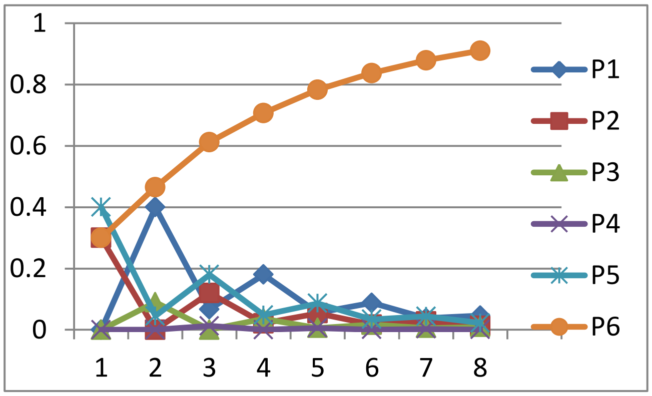
Unequal

	P_1	P_2	P_3	P_4	P_5	P_6
N=1	0	0.3	0	0	0.4	0.3
N=2	0.4	0	0.09	0	0.045	0.46
N=3	0.067	0.12	0	0.013	0.18	0.612
N=4	0.18	0.021	0.036	0	0.048	0.707
N=6	0.056	0.054	0.006	0.006	0.086	0.783
N=6	0.088	0.017	0.017	0.0009	0.034	0.837
N=7	0.038	0.027	0.006	0.003	0.044	0.879
N=8	0.046	0.012	0.009	0.0009	0.022	0.91

Unequal

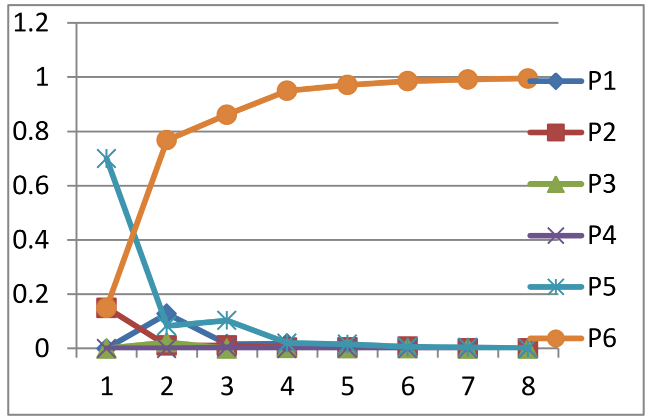
	P_1	P_2	P_3	P_4	P_5	P_6
N=1	0	0.15	0	0	0.7	0.15
N=2	0.128	0.011	0.023	0	0.083	0.768
N=3	0.016	0.011	0	0.004	0.103	0.862
N=4	0.018	0.003	0.002	0	0.02	0.95
N=6	0.004	0.004	0.0005	0.0004	0.016	0.971
N=6	0.004	0.007	0.0007	0.00008	0.006	0.985
N=7	0.002	0.0006	0.0002	0.0002	0.004	0.991
N=8	0.0008	0.0003	0.00009	0.000003	0.002	0.995

Data Set-I, Scheme-II- Unequal



No. of quantum
Figure 6.1.3

Scheme-II- Equal



No. of quantum
Figure 6.1.4

Unequal: - We empirical that, the probability of a system moving to absorbing state (deadlock state) is high as compared to other states. As no of quantum n ≥ 3, scheduler reflect changing in the pattern and the probability of execution of (P_1, P_2, P_3, P_4.) and waiting state gets low but probability of P_1 is high over the all reaming in state(P_2, P_3, P_4, P_5). Therefore, we find that equal chance of receiving the job of lottery scheduler.

Equal: - The state probability is moved independently of the quantum variation because the pattern of distribution of state probabilities is almost similar including deadlock state (P_6) in this figure (6.1.4). the probability of the lottery scheduler in the waiting state is very high as compared to other states. The special remark for this process scheduling is that probability for the state P_6 is very high.

6.2 Data Set- II

Scheme-1: Let initial probability are PB_1 =0.21, PB_2 =0.34, PB_3 =0.41, PB_4 =0.03, PB_6 =0.01,
Unequal and equal probability Matrix are follow

Unequal

X (n)

X(n-1)		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0.15	0.08	0.11	0.11	0.16	0.47
	P_2	0.06	0.1	0.16	0.19	0.19	0.3
	P_3	0.16	0.11	0.16	0.2	0.2	0.17
	P_4	0.1	0.12	0.18	0.29	0.1	0.21
	P_5	0.09	0.1	0.16	0.25	0.19	0.21
	P_6	0	0	0	0	0	1

Equal

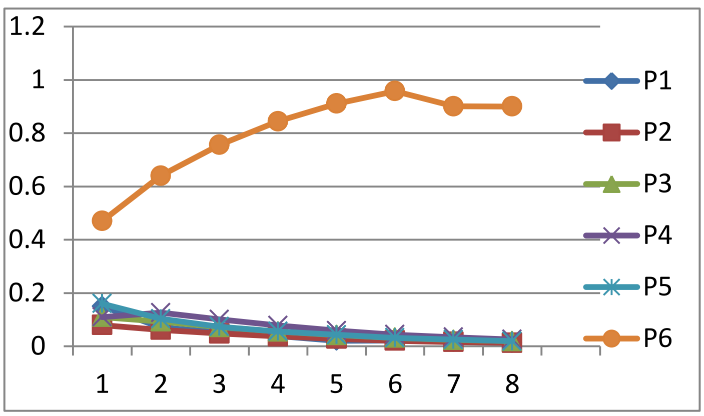
X (n)

X(n-1)		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0.13	0.13	0.13	0.13	0.35	.0.13
	P_2	0.13	0.13	0.13	0.13	0.35	0.13
	P_3	0.13	0.13	0.13	0.13	0.35	0.13
	P_4	0.13	0.13	0.13	0.13	0.35	0.13
	P_5	0.13	0.13	0.13	0.13	0.35	0.13
	P_6	0	0	0	0	0	1

Table 6.2.1 Transition probability below for Unequal cases and Equal Cases.
Graphical pattern for Unequal and Equal

	Unequal						Equal						
	P_1	P_2	P_3	P_4	P_5	P_6		P_1	P_2	P_3	P_4	P_5	P_6
N=1	0.15	0.08	0.11	0.11	0.16	0.47	N=1	0.13	0.13	0.13	0.13	0.35	.0.13
N=2	0.071	0.062	0.093	0.126	0.103	0.64	N=2	0.114	0.114	0.114	0.114	0.305	0.243
N=3	0.052	0.048	0.072	0.101	0.074	0.756	N=3	0.099	0.099	0.099	0.099	0.267	0.342
N=4	0.039	0.037	0.055	0.078	0.056	0.844	N=4	0.087	0.087	0.087	0.087	0.233	0.429
N=5	0.021	0.028	0.042	0.059	0.043	0.911	N=5	0.076	0.076	0.076	0.076	0.204	0.505
N=6	0.022	0.021	0.032	0.044	0.032	0.958	N=6	0.067	0.067	0.067	0.067	0.178	0.572
N=7	0.017	0.016	0.024	0.034	0.025	0.901	N=7	0.058	0.058	0.058	0.058	0.157	0.63
N=8	0.013	0.013	0.019	0.026	0.019	0.9	N=8	0.051	0.051	0.051	0.051	0.137	0.681

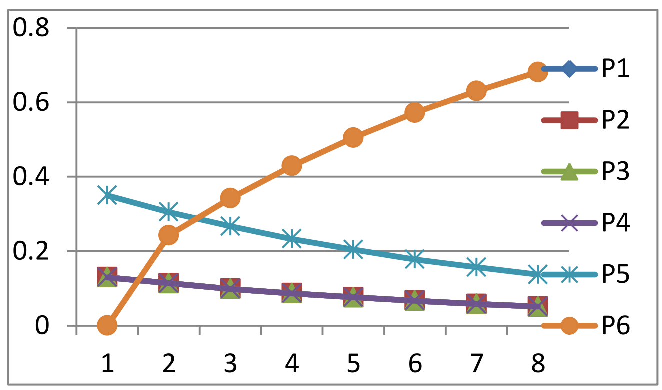
Data Set –II, Scheme-I-Unequal



No. Of quantum

Figure 6.2.1

Scheme-I-Equal



No. Of quantum

Figure 6.2.2

Unequal:- - It comes to know that the probability of a system moving to deadlock state) is high as compared to other states. And all reaming state is showing equal performance including waiting state . Therefore, we find that equal chance of receiving the job of lottery scheduler.

Equal :- We analyzed that, the probability of the scheduler in the absorbing (deadlock) state is very high value as compared to other states over different quantum which is a sign of increasing the performance efficiency of the lottery scheduler in the data sets. The probability of state P_2 is higher than the other states (P_1, P_2, P_3, P_4). Most of the transition probabilities are almost equal in fig. (6.2.2). Therefore, this dataset Provides chance for job Processing in deadlock state as well.

Scheme-II: Let initial probabilities are PB_1 =1.0, PB_2 =0.00, PB_3 =0.00, PB_4 =0.00, PB_5 =0.00, Unequal and equal probability Matrix is follow:

Unequal

X (n) →, X(n-1) ↓

	P1	P2	P3	P4	P5	P6
P1	0	0.20	0	0	0.50	0.30
P2	0.35	0	0.20	0	0.20	0.25
P3	0.50	0	0	0.20	.15	0.15
P4	0.40	0	0	0	0.30	0.30
P5	0.80	0	0	0	0	0.20
P6	0	0	0	0	0	1

Equal

X (n) →, X(n-1) ↓

	P1	P2	P3	P4	P5	P6
P1	0	0.13	0	0	.74	0.13
P2	0.12	0	0.13	0	0.61	0.13
P3	0.13	0	0	0.13	0.61	0.13
P4	0.13	0	0	0	0.74	0.13
P5	0.13	0	0	0	0	0.87
P6	0	0	0	0	0	1

Unequal

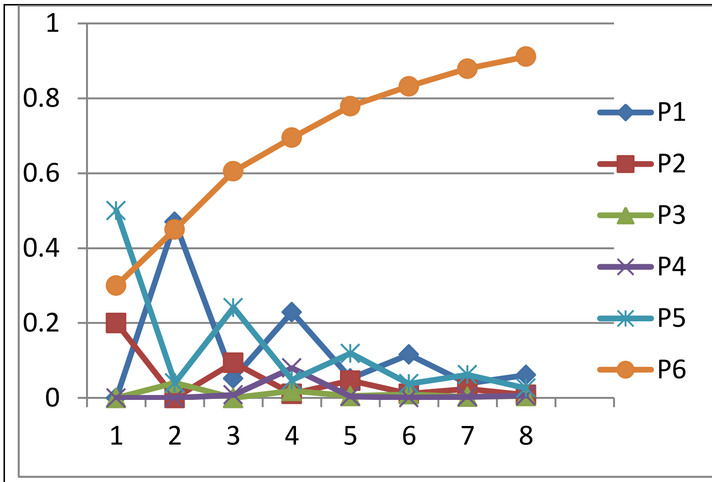
	P1	P2	P3	P4	P5	P6
N=1	0	0.2	0	0	0.5	0.3
N=2	0.47	0	0.04	0	0.04	0.45
N=3	0.052	0.094	0	0.008	0.241	0.605
N=4	0.229	0.011	0.019	0.08	0.048	0.695
N=5	0.052	0.046	0.005	0.004	0.119	0.779
N=6	0.116	0.011	0.01	0.001	0.038	0.832
N=7	0.039	0.024	0.003	0.003	0.062	0.879
N=8	0.061	0.008	0.005	0.006	0.026	0.911

Equal

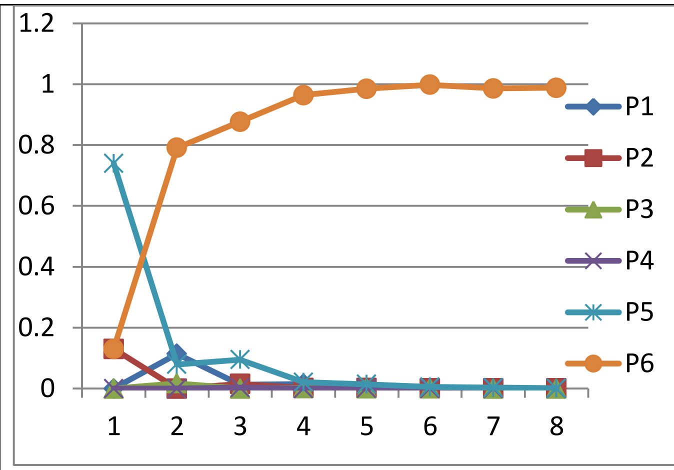
	P1	P2	P3	P4	P5	P6
N=1	0	0.13	0	0	0.74	0.13
N=2	0.114	0	0.017	0	0.079	0.791
N=3	0.013	0.015	0	0.003	0.095	0.877
N=4	0.015	0.002	0.002	0	0.021	0.964
N=5	0.004	0.002	0.0003	0.0003	0.014	0.985
N=6	0.003	0.0006	0.0003	0.00004	0.005	0.998
N=7	0.0008	0.0004	0.000008	0.00004	0.003	0.986
N=8	0.0005	0.0002	0.00006	0.00002	0.0009	0.988

Table 6.2.1 Transition Probability below for Unequal cases and Equal Cases.

Data Set II, Scheme –II-Unequal	Data Set II, Scheme –II-equal



No. Of quantum
Figure6.2.3



No. Of quantum
Figure6.2.4

Graphical pattern for Unequal and Equal

Unequal:- all state Probabilities are shown independent behavior over the all quantum variance including deadlock state. But the special remark is that P_1 state is P_4 increasing manner after few quantum. And P_2 is also high as compared to other state (P_3, P_4, P_5) So this is a good sign for lottery scheduling.

Equal:- We intended that, the probability of the scheduler in the absorbing (deadlock) state is low to high value as compared to other states (p_1, p_2, p_3, p_4,p_5) over different quantum which is a sign of increasing the performance efficiency of the lottery scheduler in the data sets. And the Probability of state P_6 is higher than the other data sets.

6.3 Data Set –III

Scheme-1.: let initial Probability are PB_1=.20, PB_2=.34, PB_3=.40, PB_4=0.03, PB_5=.03

Unequal and equal Probability Matrix are follow:

Unequal

X (n)

		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0.06	0.2	0.13	0.13	0.24	0.24
	P_2	0.04	0.14	0.16	0.16	0.2	0.3
X(n-1)	P_3	0.15	0.05	0.16	0.27	0.19	0.26
	P_4	0.1	0.14	0.2	0.26	0.19	0.11
	P_5	0.06	0.16	0.27	0.16	0.22	0.13
	P_6	0	0	0	0	0	1

Equal

X (n)

		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	.18	.18	.18	.18	.1	.18
X(n-1)	P_2	.18	.18	.18	.18	.1	.18
	P_3	.18	.18	.18	.18	.1	.18
	P_4	.18	.18	.18	.18	.1	.18
	P_6	.18	.18	.18	.18	.1	.18
	P_6	0	0	0	0	0	1

Unequal

	P_1	P_2	P_3	P_4	P_5	P_6
N=1	0.06	0.2	0.13	0.13	0.24	0.24
N=2	0.059	0.101	0.152	0.148	0.157	0.394
N=3	0.055	0.071	0.121	0.129	0.126	0.515
N=4	0.045	0.066	0.098	0.105	0.103	0.612
N=5	0.037	0.055	0.081	0.087	0.086	0.694
N=6	0.031	0.046	0.068	0.072	0.071	0.762
N=7	0.026	0.038	0.056	0.06	0.059	0.819
N=8	0.021	0.031	0.047	0.049	0.049	0.866

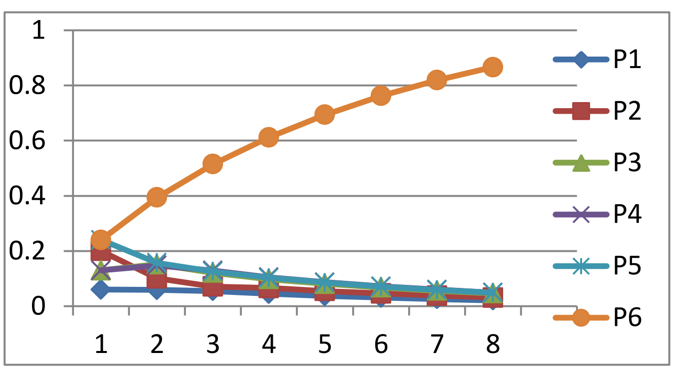
Equal

	P_1	P_2	P_3	P_4	P_5	P_6
N=1	0.18	0.18	0.18	0.18	0.1	0.18
N=2	0.148	0.148	0.148	0.148	0.129	0.31
N=3	0.13	0.13	0.13	0.13	0.12	0.417
N=4	0.116	.116.	0.116	0.116	0.107	0.511
N=6	0.103	0.103	0.103	0.103	0.095	0.595
N=6	0.092	0.092	0.092	0.092	0.085	0.67
N=7	0.082	0.082	0.082	0.082	0.076	0.734
N=8	0.073	0.073	0.073	0.073	0.068	0.794

Table 6.3.1The Transition Probabilities for Unequal and Equal Probabilities

Graphical pattern for Unequal and Equal

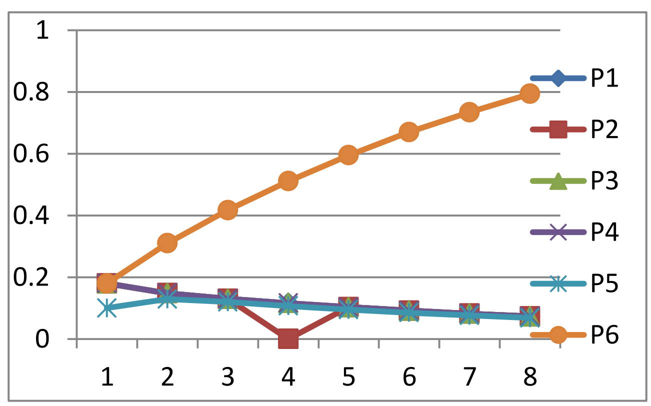
Data Set –III, Scheme-I-Unequal



No. Of quantum

Figure 6.2.1

Scheme-I-Equal



No. Of quantum

Figure 6.2.2

Unequal:- We experiential that, the probability of a system moving to absorbing state (deadlock state) is high as compared to other states. As no of quantum n ≥ 2, the scheduler reflect changes in the pattern and the probability of working (P_1, P_2, P_3, P_4.) and waiting state gets high over the other states (P_1, P_2, P_3, $_4$).Therefore, we find that equal chance of receiving the job of lottery scheduler.

Equal: - We observed that, the probability of system moving to absorbing state (deadlock state) is high as compared to other states (P_1, P_2, P_3, P_4,P_5) and also P_2 is little bit high over the remaining states.. Therefore, we find that there is less equal chance of receiving the job of lottery scheduler.

Scheme-II: - Let initial Probability arePB_1 =1.0, PB_2 =0.00, PB_3 =0.00, PB_4 =0.00, PB_5 =0.00.

Unequal and equal Probability Matrix are follow:

Unequal

X (n) →

X(n-1) ↓		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0	0.50	0	0	.25	0.25
	P_2	0.55	0	0.15	0	.15	0.15
	P_3	0.25	0	0	0.25	0.25	0.25
	P_4	0.35	0	0	0	0.35	0.30
	P_5	0.90	0	0	0	0	0.10
	P_6	0	0	0	0	0	1

Equal

X (n) →

X(n-1) ↓		P_1	P_2	P_3	P_4	P_5	P_6
	P_1	0	0.18	0	0	0.84	0.18
	P_2	0.18	0	0.18	0	0.46	0.18
	P_3	0.18	0	0	0.18	.46	0.18
	P_4	0.18	0	0	0	0.64	0.18
	P_6	0.18	0	0	0	0	0.82
	P_6	0	0	0	0	0	1

Unequal

	P_1	P_2	P_3	P_4	P_5	P_6
N=1	0	0.5	0	0	0.25	0.25
N=2	0.5	0	0.075	0	0.075	0.35
N=3	0.087	0.25	0	0.019	0.144	0.502
N=4	0.274	0.044	0.038	0	0.066	0.582
N=5	0.094	0.137	0.007	0.009	0.085	0.674
N=6	0.157	0.047	0.021	0.002	0.049	0.731
N=7	0.076	0.079	0.008	0.006	0.053	0.789
N=8	0.096	0.096	0.012	0.002	0.055	0.829

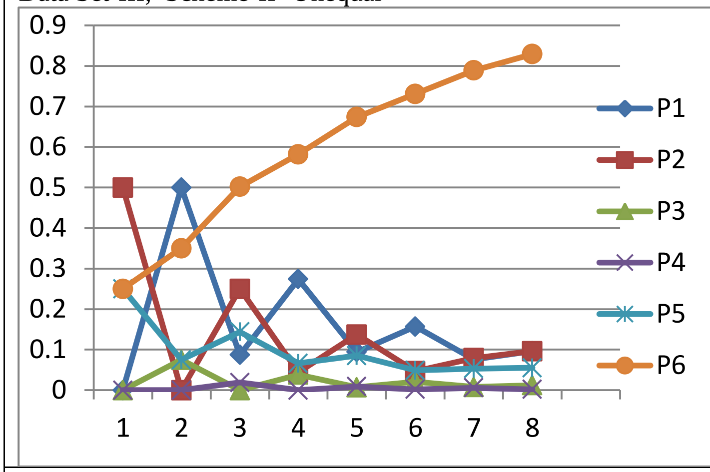
Equal

	P_1	P_2	P_3	P_4	P_5	P_6
N=1	0	0.18	0	0	0.64	0.18
N=2	0.147	0	0.033	0	0.083	0.737
N=3	0.201	0.027	0	0.006	0.109	0.8
N=4	0.026	0.037	0.005	0	0.145	0.968
N=5	0.034	0.005	0.007	0.0009	0.036	0.932
N=6	0.009	0.007	0.009	0.002	0.028	0.969
N=7	0.009	0.002	0.002	0.002	0.002	0.97
N=8	0.002	0.002	0.0004	0.0004	0.009	0.975

Table 6.3.2The Transition probabilities for Unequal and Equal Probabilities

Graphical pattern for Unequal and Equal

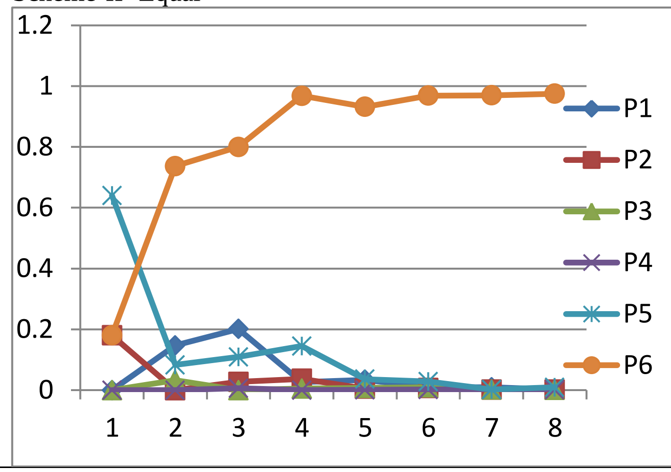
Data Set-III, Scheme-II- Unequal



No. Of quantum

Figure 6.3.3

Scheme-II- Equal



No. Of quantum

Figure 6.3.4

Unequal:- In this graphical pattern (figure 6.3.3), we observed that state probability P_1 and P_2 is showing the best performance as compared to other states (P_3, P_4, P_5). The special remark is that state probability P_3 also perform little bit high as compared to other processes (P_4). Although the scheduler execute more jobs as compared to previous one. But it still shows excellent performance, efficiency under this data set due to higher probabilities of waiting state and lower Probabilities for the stateP_1, P_2, P_3, $P_4$and P_5 as compared to state P_6 with Unequal data set.

Equal:- In this graphical pattern (figure 6.3.4), we observed that state probability P_1 is showing the best performance as compared to other states (P_2, P_3, P_4,). The special remark is that state probability P_3 also perform little bit high as compared to other processes (P_2, P_3). Thus the scheduler executes more jobs as compared to other data sets. But it still shows average performance, efficiency under this data set due to low Probabilities of waiting state and lower probabilities for the stateP_1, P_2 as compared to state P_6 in equal data set. Therefore this situation is good for lottery scheduling.

CONCLUSION

The purpose of this work describe the performance of a lottery scheduling algorithm by introducing deadlock condition and did a comparative analysis of two type of schemes which using Markov chain model and analyzed unequal and equal probability matrix with a three different data sets. and we calculated the probability of variation over quantum in the there three datasets (Data set-I, Data Set II and Data Set III). And also obtained that with decreased values in terms of number of quantum the probability of occurrence of deadlock also decreased proportionally that means for shorter process that requires less number of quantum the probability of occurrence deadlock is also proportionally less, Therefore, there are more chance to get executed for jobs contained in state Data set I and Data set III. Further, the transition state for higher value, probability lead to quantum Independence and the information overlapping occurrence in data sets (Data set-II), which shows a loss of system efficiency and serious degradation in performance of deadlock analysis of LR scheduling algorithms. Therefore, data sets (Data set-II) are not recommended for perfect utilization. Hence it is concluded that the state probabilities of the lottery-based system over these scheme-II are very useful that leads to improved performance.

REFERENCES

[1]. D. Shukla & S. Ojha(2010). "Deadlock index analysis of multi-level queue scheduling in operating system using data model approach", GESJ, 6(29), pp. 93-110.

[2]. Ruesh sendre & R.Singhai(2018). "Deadlock Analysis of Improved Round Robin CPU Scheduling Algorithm Using Markov Chain Model ,IJREAM 4(4) .

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

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

[5]. Petrou, D., Milford, J. W., & Gibson, G. A. (1999). "Implementing Lottery Scheduling: Matching the Specializations in Traditional Schedulers". In USENIX Annual Technical Conference, General Track , pp. 1-14.

[6]. Carl.A.Waldspurger, & William E. Weihl(1994). "Lottery scheduling Flexible Proportional –share resource management". IT laboratory for computer science, Cambridge.
[7]. Shweta Ojha & Saurabh jain (2011). "Hybrid Lottery multilevel queue scheduling with a Markov Model" GESJ Computer Science and Telecommunication, 03(32).
[8]. S. Reveliotis, &Z. Fei (2017). "Robust deadlock avoidance for sequential resource allocation systems with resource outages", IEEE Transactions on automation science and engineering, 14(4), pp-1696-1711.
[9]. P. Kawadkar, S. Prasad & A.D. Dwivedi(2014). "Deadlock Avoidance based on banker's algorithm for waiting state Processes", IJISME, 2(12), pp. 1-2.
[10]. S. Srinivasan & R. Rajaram(2011)."A decentralized deadlock detection and resolution algorithm for generalized model in distributed systems", Springer-Diatribe Parallel Databases, Vol 29, pp. 261-276.
[11]. D. Pandey & Vandana(2010). "Improved round robin policy a mathematical approach", IJCSE, 2(4) pp. 948-964.
[12]. S.H. Yadgire & Y.B. Jadhav(2017)."Analysis of deadlock with fungible resources", IJRSE, 3(3), pp. 290-294.
[13]. Y. Cai & W.K. Chan(2014). "Magiclock- scalable detection of potential deadlocks in large-scale multithreaded Programs", IEEE Transaction on Software Engineering, 40(3), pp. 266-281.
[14]. Brzezinski J., Helary J.M., Raynal M. & Singhal M(2002)."Deadlock Models and a general algorithm for distributed deadlock detection", Journal of Parallel & Distributed Computing, 31(2), pp. 112-126.
[15]. A. Nazeem, S. Reveliotis, Y. Wang & S. Lafortune(2011). "Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: the linear case", IEEE Transactions on automatic control, 66(8), pp. 1818-1833.
[16]. Y. Rosa, A. Shahyar & R. Vahid,(2016). "A greedy algorithm versus metaheuristic solutions to deadlock detection in graph transformation systems", JIFS, 31(1), pp. 137-149,
[17]. Manish Vyas, & Dr. Saurabh Jain(2016)."Stochastic Modeling for Analyzing Scalability Impact of Lottery Scheduling using Proportion Reformation" International Journal Of Engineering And Computer Science ,6(10), pp. 18668-18674.
[18]. Manish Vyas,& Dr. Saurabh Jain(2016)." Comparative Study of Extensive Round Robin Scheduling by Data Model Approach under Markov Chain" International Journal of Recent Research Aspects, 3(2), pp. 92-100.
[19]. D.shukla & Sweta ojha(2010)."Prediction of Ready Queue Processing Time in Multiprocessor Environment Using Lottery Scheduling (ULS)" International Journal of the Computer, the Internet and Management Vol.18.pp 68-66.
[20]. D. Shukla & S. Jain,(2007). "A Markov chain model for multi-level queue scheduler in operating system", Proceedings of the International Conference on Mathematics and Computer Science, ICMCS-07, pp. 622-626.
[21]. R. Sendre, R. Singhai& S. Jain(2018). "Markov Chain Analysis of Improved Round Robin CPU Scheduling Algorithm", IJREAM, 3(4), pp. 728-737.
[22]. I.S. Rajput, & D. Gupta(2012). "A Priority based round robin CPU scheduling algorithm for real time systems", IJIET, 1(3), pp. 01-11.
[23]. D. Shukla & S. Jain(2007) ."A Markov chain model for multi-level queue scheduler in operating system", Proceedings of the International Conference on Mathematics and Computer Science, ICMCS-07, pp. 622-626.
[24]. D. Shukla & S. Jain(2007). "Deadlock State Study in Security Based Multilevel Queue Scheduling Scheme in Operating System", Proceedings of National Conference on Network Security and Management, NCNSM-07, pp. 166-176.
[25]. J. Medhi(1991). "Stochastic Processes" Ed. 4, Wiley Limited (Fourth Reprint), New Delhi,
[26]. P.K. Jatav & R. Singhai(2018). "Comparative Study of CPU Scheduling Algorithms based on Markov Chain", IJREAM, 4(3), pp. 668-672.
[27]. A.K. Gupta, N.S. Yadav & D. Goyal(2017). "Review of Round Robin CPU Scheduling Algorithm on Varying Time Quantum", IJESI, 6(8), pp. 68-72.
[28]. A.S. Tenenbaum & A.S. Woodhull(2012). Operating Systems: Design and Implementation, 3rd edition, PHI Publication.
[29]. J. Medhi(1991). "Stochastic Models in Queuing Theory", Academic Press Professional, Inc, San Diego, CA.

