Achieving Stability in the Round Robin Algorithm

Kamal ElDahshan
Dept. of mathematics,
Computer science Division
Faculty of science, Al-Azhar
University
Cairo, Egypt

Afaf Abd El-kader
Dept. of mathematics,
Computer science Division
Faculty of science, Al-Azhar
University
Cairo, Egypt

Nermeen Ghazy
Dept. of mathematics,
Computer science Division
Faculty of science, Al-Azhar
University
Cairo, Egypt

ABSTRACT
The Round Robin algorithm is a CPU scheduling algorithm that is designed especially for time sharing systems. Each process is assigned a slice of time called time quantum to execute. To increase the performance of a Round Robin algorithm, it is important to choose an optimal time quantum. Many algorithms developed the Round Robin algorithm. This paper provides an algorithm that improves the performance of these algorithms and develops a time quantum that achieves stability in terms of the average waiting time and the average turnaround time.

General Terms
Scheduling algorithms, Round Robin algorithms.

Keywords
Operating Systems, Scheduling Algorithms, Round Robin algorithm, Adaptive Round Robin, Round Robin Remaining Time algorithm, Enriched Round Robin algorithm, Time quantum (TQ), Average Waiting Time, Average Turnaround Time.

1. INTRODUCTION
An operating system is a software that manages all of the software and hardware on the computer. It also provides a basis for application programs and acts as an intermediary between the computer user and the system hardware [1]. Most of the time, there are several computer programs ready to run concurrently. The CPU scheduler is responsible for selecting a certain process to run. There are many policies and mechanisms to control the scheduling operation. There are two kinds of scheduling algorithms depending on whether the process is allowed to be preempted or not, namely preemptive and non-preemptive scheduling algorithms [2].

In preemptive scheduling algorithms, the process can be preempted during its execution to execute the next selected process waiting in the ready queue.

While in the non-preemptive scheduling algorithm, the process is executed till completion [3] [4].

The objective function of any CPU scheduling algorithm may be one of the following criteria [5]:

1. Throughput: maximizing the number of processes completed per unit of time.

2. Makespan: maximum completion time.

3. CPU utilization: keeping the CPU as busy as Possible.

4. Turnaround Time: minimizing the total time which the process spent in the system to complete.

5. Waiting Time: minimizing the total time a process has been waiting in the ready queue.

6. Response Time: minimizing the time from the submission of a request until the first response is produced.

The remainder of the paper is organized as follows: Section (2) focuses on some related work. Section (3) focuses on Round Robin Remaining Time Algorithm. Section (4) presents the proposed algorithm Enriched Round Robin Algorithm. Section (5) discusses The experimental data. Section (6) explains Results analysis Section (7) concludes the paper and presents the future work.

2. RELATED WORK
Many algorithms have been used to schedule processes on their resources, some of these algorithms are briefly discussed below:

FCFS (First Come First Served) algorithm: it is a non-preemptive algorithm [6]. Processes that come first are executed first. This algorithm is so easy because it selects the first process regardless of its execution time. It has low throughput because long processes can keep the processor occupied for a long time causing small processes to suffer and have high average waiting time, so the FCFS algorithm has a poor performance[4] [5].

SJF (Shortest Job First) algorithm: Shortest Job First algorithm is a non – preemptive algorithm in which the shortest process is executed first. This algorithm associates with each process the length of the process' next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next execution time. If there are two processes have the same execution time, it will be executed on a first come first served basis [4] [5] [6].

SRTF (Shortest Remaining Time First) algorithm: it is the preemptive SJF algorithm. It preempts the currently executing process to execute another one, whereas a non- preemptive SJF algorithm allows the process to execute till completion [4] [5] [6].

Priority Scheduling Algorithm: it is a non-preemptive algorithm. Each process is assigned a priority which can be determined based on the memory requirements, time requirements or any other resource requirement [6]. The Process with the highest priority will be executed first and so on. If some processes have the same priority, they will be executed on a first come first served basis [4] [5].

Due to a number of disadvantages of these scheduling algorithms, they are rarely used except the Round Robin scheduling algorithm which has been used and ameliorated in time sharing and real-time operating systems. The Round Robin algorithm is considered to be the most widely used CPU scheduling algorithm.

Round Robin scheduling algorithm: it is a preemptive algorithm [2] [6] which is effective in time-sharing environments. These environments need to guarantee reasonable response times for interactive users [7].

The Round Robin scheduling algorithm assigns a time slice (time quantum) for each process in the ready queue. When the TQ terminates, the current process is added to the end of the ready queue. It is better than other scheduling algorithms because it reduces turnaround time and waiting time.

The most interesting issue with the Round Robin algorithm is the time quantum length. Setting a too short time quantum causes too many context switches and the efficiency of the CPU will be low [4] [5]. On the other hand, setting the TQ too long may cause a poor response time and the algorithm tends to FCFS.

Adaptive Round Robin Scheduling Algorithm: this algorithm improved the Round Robin algorithm. It is a "priority scheduling algorithm" based on the burst time of processes. It is composed of two phases. First, the processes are arranged according to the burst time. The highest priority is given to the process with smallest burst time [8]. This approach chooses the smart time slice depending on the number of processes. If the number of processes is even, then the smart time slice will be the average of the burst of all processes [9]. If the number of processes is odd, then the smart time slice is equal to the burst time of the mid-process.

3. ROUND ROBIN REMAINING TIME ALGORITHM

This algorithm improved the Adaptive Round Robin Scheduling algorithm. It assumes that all processes arrive at the same time in the ready queue, then they are arranged in an ascending order according to their burst time. TQ is calculated by Σ pi / 2n. If the remaining CPU burst time of the currently running process is less than the TQ, the CPU is again allocated to the currently running process for the remaining CPU burst time. Otherwise, the process will be added to the end of the ready queue [10].

4. THE PROPOSED ALGORITHM (ENRICHED ROUND ROBIN ALGORITHM)

If all the processes arrive at the same time to the ready queue, then all the processes are arranged in an ascending order according to their burst time. The time quantum is calculated by (¾*mean). If the remaining CPU burst time of the currently running process is less than the time quantum, the CPU is again allocated to the currently running process for the remaining CPU burst time. Otherwise, the process will be appended to the end of the ready queue.

Enriched Round Robin Algorithm:

1. Assign processes to the ready queue.
2. Arrange all the processes an in ascending order according to their burst time
3. If (ready queue! =NULL) then TQ ← 3*mean / 4
4. While (ready queue! =NULL)
5. If (remaining burst time < TQ)
6. Allocate CPU again to the current running process for the remaining burst time.

    Else

7. Put the remaining of the current process at the end of the ready queue.

8. If number of processes > 0
9. Go to step 5
10. End while
11. Calculate average waiting time, average turnaround time.

5. EXPERIMENTAL DATA

There are three cases for the incoming processes' burst time; random, increasing and decreasing. The order of incoming processes affects the results of the Round Robin algorithm since the processes are executed in their incoming order. Whereas in the Round Robin Remaining Time algorithm and the Enriched Round Robin, the processes are arranged in an ascending order according to their burst time. The results of the three cases are described below.

Case I: The incoming burst time in an increasing order
Consider five processes with increasing burst time shown in Table 1. The Gantt charts for the three algorithms Standard Round Robin, Round Robin Remaining Time Algorithm and the Enriched Round Robin are shown in Figures 1, 2 and 3 respectively.
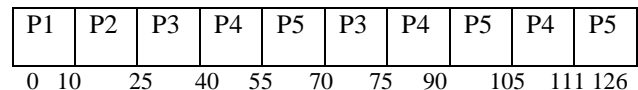
Table 1. The incoming burst time in an increasing order

Process Name	CPU Burst Time (ms)
P1	10
P2	15
P3	20
P4	36
P5	45

-Standard Round Robin Algorithm
Enter the processes in the ready queue in their order; P1=10, P2=15, P3=20, P4=36 and P5=45 with TQ=15.

Gantt chart

P1	P2	P3	P4	P5	P3	P4	P5	P4	P5

0 10   25   40   55   70   75   90   105   111 126

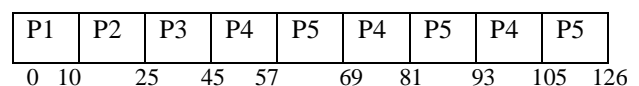**Figure 1: Gantt chart for Standard Round Robin Algorithm (Case I)**

Average Waiting Time= 44.2ms
Average Turnaround Time = 69.4ms

**-Round Robin Remaining Time algorithm**
Arrange all the processes in the ready queue in an ascending order according to their burst time; P1=10, P2=15, P3=20, P4=36 and P5=45. Calculate the time quantum according to the Round Robin Remaining Time Algorithm. The calculated time quantum is 12.

Gantt chart

P1	P2	P3	P4	P5	P4	P5	P4	P5

0 10   25   45   57   69   81   93   105   126

Figure 2: Gantt chart for Round Robin Remaining Time Algorithm (Case I)

Average Waiting Time= 37ms
Average Turnaround Time = 62.2ms

-The Enriched Round Robin algorithm
Arrange all the processes in the ready queue in an ascending order according to their burst time; P1=10, P2=15, P3=20, P4=36 and P5=45. Calculate the time quantum according to the Enriched Round Robin Algorithm. The calculated time quantum is 18.

Gantt chart

P1	P2	P3	P4	P5	P4	P5	P5
0	10	25	45	63	81	99	117 126

Figure 3: Gantt chart for the Enriched Round Robin Algorithm (Case I)

Average Waiting Time= 35.8ms
Average Turnaround Time = 61ms

The following table presents a comparative study among the three algorithms with respect to time quantum, average waiting time and average turnaround time.

Table 2 Comparative study of Round Robin, Round Robin Remaining Time and the Enriched RR Algorithms

Algorithm	Time quantum	Average waiting time	Average turnaround time
Round Robin	15	44.2	69.4
Round Robin Remaining Time	12	37	62.2
Enhanced RR	18	35.8	61

Graphical Representation of Case-I
A bar chart in Figure 4 illustrates the comparative study of Standard Round Robin, Round Robin Remaining Time Algorithm and the Enriched Round Robin algorithms.
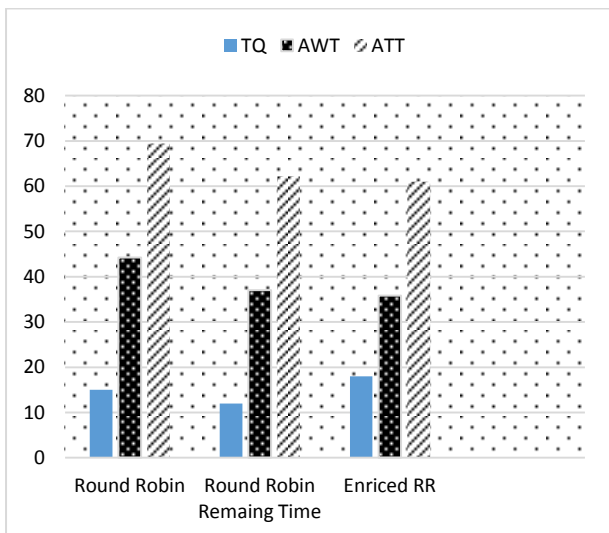
Figure 4: Comparative Graph for TQ, Average Waiting Time and Average Turnaround Time. (Case I)

Case II: The incoming burst time in a decreasing order
Consider five processes with a decreasing order of burst time as shown in Table 3. The Gantt charts are shown in Figure 5, Figure 6 and Figure 7 for the three algorithms

Standard Round Robin, Round Robin Remaining Time Algorithm and the Enriched Round Robin respectively.

Table 3. The incoming burst time in a decreasing order

Process Name	CPU Burst Time (ms)
P1	83
P2	57
P3	30
P4	19
P5	5

-Standard Round Robin Algorithm
Enter the processes into the ready queue in their order; P1=83, P2=57, P3=30, P4=19 and P5=5 with time quantum=26.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P3	P1	P2	P1
0	26	52	78	97	102	128	154	158	184	189 194

Figure 5: Gantt chart for Standard Round Robin Algorithm (Case II)

Average Waiting Time= 109.2ms
Average Turnaround Time = 148ms

-Round Robin Remaining Time algorithm
Arrange all the processes in the ready queue in an ascending order according to their burst time; P5=5, P4=19, P3=30, P2=57 and P1=83. Calculate the time quantum according to the Round Robin Remaining Time Algorithm. The calculated time quantum is 19.

Gantt chart

P5	P4	P3	P2	P1	P2	P1	P2	P1
0	5	24	54	73	92	111	130	149 194

Figure 6: Gantt chart for Round Robin Remaining Time Algorithm (Case II)

Average Waiting Time= 46.4ms
Average Turnaround Time = 85.2ms

-The Enriched Round Robin algorithm
Arrange all the processes in the ready queue in an ascending order according to their burst time; P5=5, P4=19, P3=30, P2=57 and P1=83. Calculate the time quantum according to Enriched Round Robin Algorithm. The calculated time quantum is 28.

Gantt chart

P5	P4	P3	P2	P1	P2	P1
0	5	24	54	82	110	139 194

Figure 7: Gantt chart for the Enriched Round Robin Algorithm (Case II)

Average Waiting Time= 44.4ms
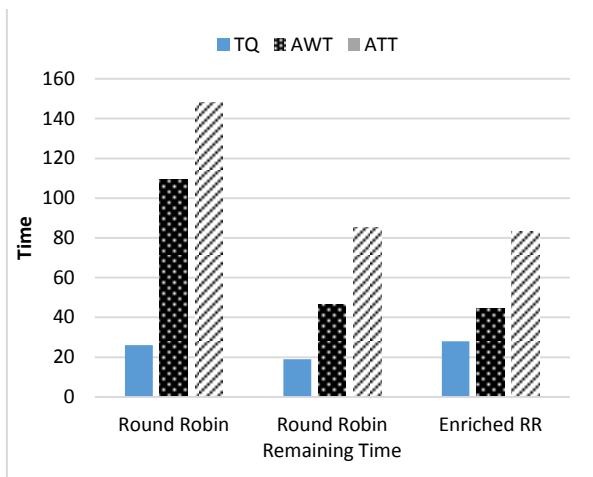Average Turnaround Time = 83.2ms

The following table presents a comparative study among the three algorithms with respect to time quantum, average waiting time and average turnaround time for case II.

Table 4: Comparative study of Round Robin, Round Robin Remaining Time and the Enriched RR Algorithms

Algorithm	Time quantum	Average waiting time	Average turnaround time
Round Robin	26	109.2	148
Round Robin Remaining Time	19	46.4	85.2
Enriched RR	28	44.4	83.2

Graphical Representation of Case-II
A bar chart in Figure 8 illustrates the comparative study of Standard Round Robin, Round Robin Remaining Time Algorithm and the Enriched Round Robin algorithms.

Figure 8: Comparative Graph for TQ, Average Waiting Time and Average Turnaround Time (Case II)

Case III: Random incoming burst time
Consider five processes with random burst time shown in Table 5. Figure 9, Figure 10 and Figure 11 shows Gantt chart for the three algorithms; Standard Round Robin, Round Robin Remaining Time Algorithm and the Enriched Round Robin respectively.
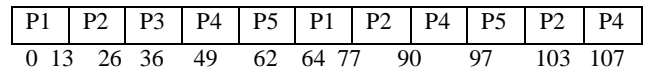
Table 5. Random incoming burst time

Process Name	CPU Burst Time (ms)
P1	15
P2	32
P3	10
P4	30
P5	20

-Standard Round Robin Algorithm
Enter the processes in the ready queue in their order; P1, P2, P3, P4 and P5. Let TQ=13.

Gantt chart

| P1	P2	P3	P4	P5	P1	P2	P4	P5	P2	P4
0	13	26	36	49	62	64	77	90	97	103	107

Figure 9: Gantt chart for Standard Round Robin Algorithm (Case III)

Average Waiting Time= 60ms
Average Turnaround Time = 81.4ms

-Round Robin Remaining time algorithm
Arrange all the processes in the ready queue in an ascending order according to their burst time; P3=10, P1=15, P5=20, P4=30 and P2=32. Calculate the time quantum according to Round Robin Remaining Time Algorithm. The calculated time quantum is 10.

Gantt chart

| P3	P1	P5	P4	P2	P5	P4	P2	P4	P2
0	10	25	35	45	55	65	75	85	95	107
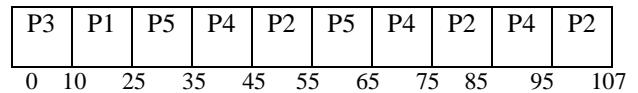
Figure 10: Gantt chart for Round Robin Remaining Time Algorithm (Case III)

Average Waiting Time= 39ms
Average Turnaround Time = 60.4ms

-The Enriched Round Robin algorithm
Arrange all the processes in the ready queue in an ascending order according to their burst time; P3=10, P1=15, P5=20, P4=30 and P2=32. Calculate the time quantum according to Enriched Round Robin Algorithm. The calculated time quantum is 15.

Gantt chart

P3	P1	P5	P4	P2	P4	P2
0	10	25	45	60	75	90	107
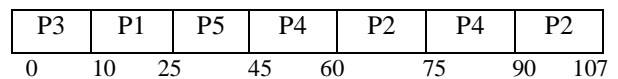
Figure 11: Gantt chart for the Enriched Round Robin Algorithm (Case III)

Average Waiting Time= 34ms
Average Turnaround Time = 55.4ms

The following table presents a comparative study among the three algorithms with respect to time quantum, average waiting time and average turnaround time for case III.

Table 6. Comparative study of Round Robin, Round Robin Remaining Time and the Enriched RR Algorithm

Algorithm	Time quantum	Average waiting time	Average turnaround time
Round Robin	13	60	81.4
Round Robin Remaining Time	10	39	60.4
Enriched RR	15	34	55.4

Figure 12 below shows the comparison of average waiting time, average turnaround time and TQ for the three algorithms.
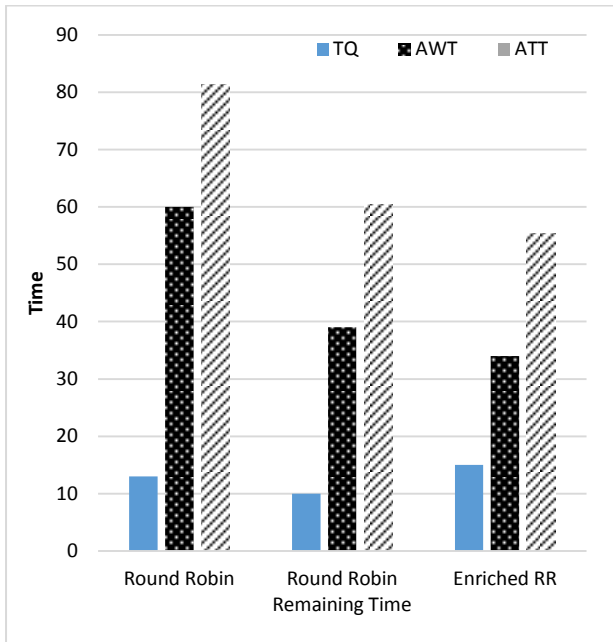

Figure 12: Graph for TQ, Average Waiting Time and Average Turnaround Time (Case III)

6. RESULTS ANALYSIS

A simulation is made for the Standard Round Robin, Enriched RR algorithm and the Round Robin Remaining Time algorithm to analyze the performance of each algorithm. The simulation is written using the C++ language. The model consists of one resource, n processes with n varying from 2000 to 10000. The values of the burst time of processes are chosen randomly. The simulation is done on laptop, 4 GB (RAM), Core i5 and 64-bit operating system.

To present the advantages of the Enriched RR algorithm against the Round Robin Remaining Time algorithm; Figure 13 plots the average waiting time for both the Enriched RR algorithm and the Round Robin Remaining Time algorithm. In this figure, when n=2000 processes, the Round Robin Remaining Time algorithm gives AWT= 43053.37ms while the Enriched RR algorithm gives AWT= 36810.33ms. When n=10000 processes, the Round Robin Remaining Time algorithm gives AWT=212610.31ms while the Enriched RR algorithm gives AWT=184675.32ms.

Figure 14 plots the average turnaround time for both the Enriched RR algorithm and the Round Robin Remaining Time algorithm. In this figure, the number of processes n varies from 2000 to 10000. It is observed that the results of the average turnaround time of the Round Robin Remaining Time algorithm vary from 43104.4ms to 212661.18ms whereas the results of the average turnaround time of the Enriched Round Robin algorithm vary from 36810.3ms to 184726.19ms.

It is noticed that from Figures 13 and 14 that the Enriched Round Robin algorithm gives better results than the Round Robin Remaining Time algorithm in terms of average waiting time and average turnaround time.
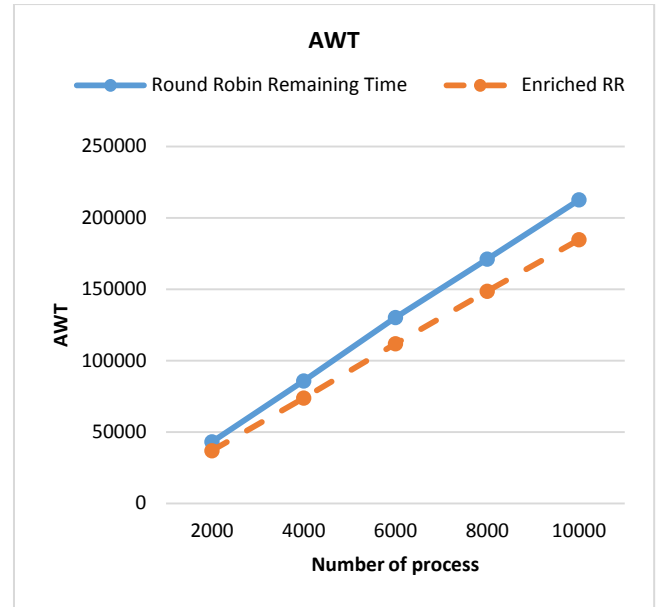

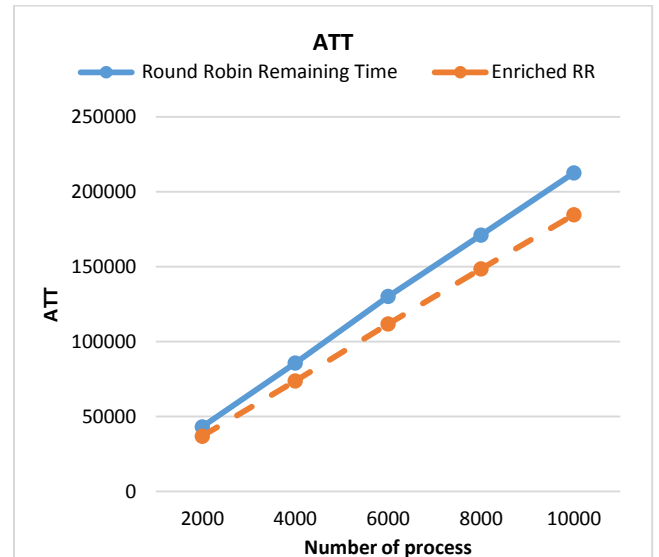**Figure 13: Comparative Graph for Average Waiting Time**

Figure 14: Comparative Graph for Average Turnaround Time

Different values for the time quantum are studied; (3/4*mean), median and mode. The number of processes n=2000 processes, where 1000 processes with random burst time in the range (1000 to 2000) and 1000 processes with random burst time in the range(1 to 100). Results are shown in Table 7 below.

Table 7 Comparison of different time quantum

	(3/4* Mean)	median	mode
Time Quantum	579	550	14
Average Waiting Time	474355.0	490288.49	710576.29
Average Turnaround Time	475127.2	491060.70	711348.50

Case1: TQ=mode. If the mode is small, it causes a lot of context switches, high average waiting time and high average turnaround time. If the mode is large, it causes poor response time and the algorithm tends to be FCFS.

Case2: TQ=median. If the random set of burst time contains many large numbers and few small numbers, then the median becomes large which causes poor response time and the algorithm tends to be FCFS as shown in Figure 15. On the other hand, if the random set of burst time contains many small numbers and few large numbers, then the median becomes small which causes a lot of context switches and high average waiting time and average turnaround time.

Hence, the mode and the median results are not reliable.

Different ratios of mean have been studied. By taking ratios from TQ= (0.75*mean) until TQ= (0.90*mean) the results become unreliable, which may give either high or low average waiting time and average turnaround time.

Figure 16 shows that from TQ=mean until TQ=2*mean, the resulting average waiting time has the same value and stable.

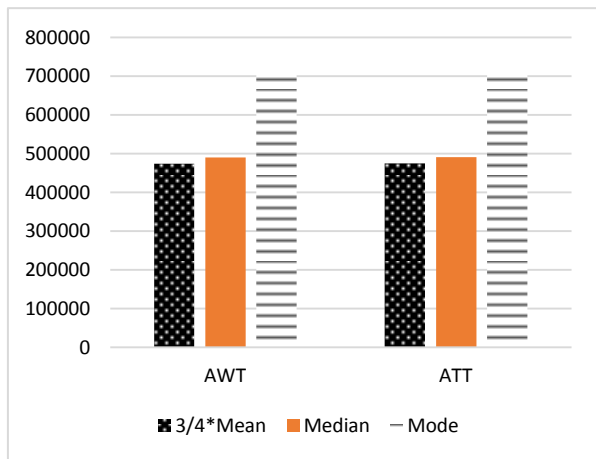So the TQ= (0.75* mean) is the optimum time quantum whether the burst times are large or small as shown below.

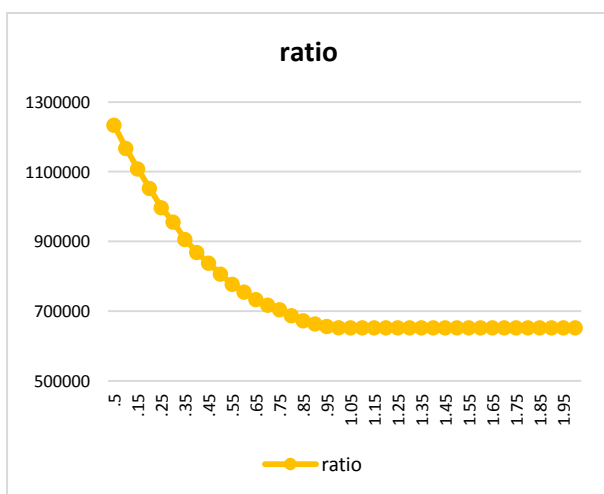**Figure 15: Comparison of different time quantum ratios**

Figure 16: Comparison of different ratios of mean

7. CONCLUSION

The Round Robin Remaining Time algorithm assigns processes in an ascending order with time quantum = Σ pi / 2n. This paper proposed the Enriched Round Robin algorithm and showed that the optimum time quantum that should be assigned to the processes in an ascending order is (0.75*mean). It showed that the Enriched Round Robin algorithm minimizes the average waiting time (AWT) and the average turnaround time (ATT).

8. REFERENCES

[1] A. Silberschatz, P. B. Galvin, and G. Gagne, "Operating System Concepts", (9th Edn. John Wiley and Sons Inc, ISBN 978-1-118-06333-0), 2013.

[2] Sukumar Babu Bandarupalli, Neelima Priyanka Nutulapati, Prof. Dr. P.Suresh Varma, "A Novel CPU Scheduling Algorithm–Preemptive & Non-Preemptive", International Journal of Modern Engineering Research (IJMER), Vol.2, Issue 6, pp 4484-4490, (Nov-Dec) 2012.

[3] Suri, P.K and Sumit, M, "Design of Stochastic Simulator for Analyzing the Impact of Scalability on CPU Scheduling Algorithms", International Journal of Computer Applications, Vol.49, Issue 17, pp 4-9, July 2012.

[4] Abdulrazaq Abdulrahim, Salisu Aliyu, Salisu Aliyu, Saleh E Abdullahi, "An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume.4, Issue 2, pp 601-610, February 2014.

[5] Manish K. M. and Abdul Kadir K, "An Improved Round Robin CPU Scheduling Algorithm", Journal of Global Research in Computer Science, Volume 3, No. 6, pp 64-69, 2012.

[6] Imran Qureshi, "CPU Scheduling Algorithms: A Survey", Int. J. Advanced Networking and Applications, Vol. 05, Issue 04, pp 1968- 1973, 2014.

[7] Sanjaya Kumar Panda, Sourav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max Dispersion Measure", International Journal on Computer Science and Engineering (IJCSE), Vol. 4, Issue 01, pp 45-53, January 2012.

[8] Saroj Hiranwal, Dr. K.C. Roy "Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice", International Journal of Computer Science and Communication, Vol.2, Issue 2, pp 319-323, (July- December) 2011.

[9] Jayanti Khatri, "An Enhanced Round Robin CPU Scheduling Algorithm", IOSR Journal of Computer Engineering (IOSR-JCE), Vol.18, Issue 4, pp 20-24, Ver. II (Jul-Aug) 2016.

[10] Arpita Sharma1, Mr. Gaurav Kakhani, "Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm", International Journal of Computer Science and Mobile Computing, Vol.4, Issue.12, pp. 139-147, December- 2015.