A NOVEL METHOD BASED ON PRIORITY FOR ENHANCEMENT ROUND-ROBIN SCHEDULING ALGORITHM

[1] AHMED SUBHI ABDALKAFOR, [2] HADEEL MOHAMMED TAHER, [3] KHALID W. AL-ANI

[1] Career Development Center, University of Anbar, Anbar, Iraq

[2] Qualtiy Assurance and Academic Performance Department. University of Anbar, Anbar, Iraq

[3] Department of studies and planning, University of Anbar, Anbar, Iraq

E-mail: [1] ahmed_abdalkafor@yahoo.com, [2] hadeilmt@yahoo.com, [3] encomkh@yahoo.com

ABSTRACT

Scheduling of central processing unit is one of the greatest essential operations implemented over operating system (OS). There are many different algorithm used for scheduling but the main one called round-robin(RR) algorithm that achieved by optimum in period shared environment. The feature of RR algorithm is decrease the starvation besides integrates the improvement of priority. In this paper we propose a new optimization for the round- robin algorithm to improvement the CPU scheduling through all task allocated in CPU takes a new priorities depended on lowest value of burst time take highest priority when quantum time have the same value, then rescheduling to give a new priority after compute the burst time of tasks that will be reduce the average waiting time(A.W.T) and turnaround time (T.A.T) compare with standard round- robin algorithm and other related works.

Keywords: *CPU Scheduling, Round Robin, Priority, Waiting Time, Turnaround Time.*

1. INTRODUCTION

The system of computers depended on the CPU for any operation that will be create a problem in the time for each process. For that the CPU arrangement it's very important in operating scheme. When CPU has one process is very easy to calculate but when CPU have many processes its must be wait in queue arrange to processed in CPU [1]. On the other hand the strategy to organized different by different algorithms for CPU operation system to execution the processes. Allocating CPU to a process necessitates wary care to evade process starvation and guarantee justice in CPU [2]. There are many algorithms such as neural networks and others have been applied to scheduling the processor and other applications [3] [4] [5] as well as there are many algorithms used for scheduling algorithms such as: First Come First Serve (FCFS) Scheduling, Priority Based Scheduling, Shortest Job First (SJF) Scheduling and Round Robin algorithms. Each one of these algorithms have benefits and weaknesses. There are two main categories of scheduling algorithms called preemptive and non-preemptive. In Preemptive Scheduling, a procedure of temporarily appending the processes which are logically runnable. On the other hand Pre-emptive Scheduling a process which is allocated to the CPU can be standing and the execute state of the conforming process is changes to waiting state [6]. In this paper, a novel method based on priority for enhancement round-robin scheduling algorithm is proposed to avoid the problems of traditional round-robin algorithm and improve performance CPU Scheduling. Additionally, it has the ability to decrease the amount of waiting time and turnaround time to obtain less delay in queue waiting.

The rest of the paper is organized in Sections. Section 2 discuss the standard round robin algorithm, when the Section 3 present the previous studies of RR algorithm after that the Section 4 presents the proposed algorithm. The results and discussion is presented in Section 5 . Finally. In Section 6 conclusion is given for the proposed.

2.1 1.1 Standard Round Robin Algorithm

The standard round-robin (RR) algorithm scheduling is intended specially for timesharing methods. The round robin scheduling like first come first serve scheduling but in the round robin add the preemption to covert between the process. The unit of time in smaller period known as a time quantum or sometimes called with the time slice which can be between 10 to 100 milliseconds. The round robin need a queue usually treated as ready queue or called as circular queue [7]. The round robin used first in first out (FIFO) queue of processes. Each Process are given an equal time slice when they can be executed when each process for a time wait of up to 1 time quantum. If the quantum time is not sufficient for the process to finish its execution, it is stopped then the next process should be execute but when complete this process then the process that cant complete its run again [8]. Because this algorithm its type of preemption scheduling it doesn't have starvation, all the problem in round robin algorithm it's in the time quantum that used in each process. The main phases of RR algorithm is given through following:

- **phase 1:** The scheduler generate by a queue for processes that prepared then make a list of processes that blocked otherwise swapped out.
- **phase 2:** The newly created process typically put in the end of queue. The process is remove from the queue at what time Process Control Block (PCB)dismiss process.
- **phase 3:** Typically the Process Control Block(PCB) carefully chosen form the begin of the queue intended for scheduler.
- **phase 4:** Every process have a limited time for completed if this process don't completed in its period it will be returned to the end of the organized queue for implement in anther period.
- **phase 5**: If process request to eliminate or block from queue the Process Control Block makes it.

1.2 Main Problem In Round Robin Algorithm

The processor in any computer receives many processes at the same time meaning that all those processes need to be handled regularly without leaving any one without processing. Round robin have a big problem in the concept of OS Scheduling its depended on priority that mean the processes are implemented based on FCFS[1]. Every process have a high priority implement until it complete and do not have to delay for their next burst cycle. This paper attentions on decreasing the burst time of the traditional algorithm[5].

2. RELATED WORKS

The scheduling are very essential for CPU to insure a respectable job, there are various research resolved many difficult in round robin.

Singh et al.[7]. Presents improve round robin through concentration on the drawbacks of RR architecture which be responsible for priority to entirely the processes. Then comparison the proposed with existing scheduling algorithm on the term of waiting and varying times, in addition to the number of context switches. This suggested will be implemented in three steps which help to decrease a number of performing terms. The results shows that the proposed algorithm has reduce average turnaround time over simple round robin for varying period quantum. Furthermore, it reduces the problem of starvation.

Kumar et al. [8]. In this study use a technique called integer programming problem that has been suggested that chooses a worth that's neither big nor too lesser such every technique has inexpensive reply time and consequently the throughput of the scheme isn't changed as a result of needlessly context switches. This study based on changing time quantum in every single round completed the returning queue. The result appearance the performance of proposed method is greater than the conventional Round Robin scheme.

Kumar and Rashid [9] improvement round robin scheduling for increasing operation system performance based on the article shortest task first combination with RR scheduling(IRRVQ). quantum time is set equivalent to burst time of the initial task in the prepared queue subsequently organizing the processes' in ascending directive in relation to their burst time. The results show IRRVQ algorithm is experimentally enhancement better than traditional RR.

Sangwan et al [10]. The main objective of this paper is compare between traditional round robin algorithm with the one which possesses guaranteed modifications based on term of the average waiting time(W.T) and turnaround time T.A.T without consideration to number of switch cases that used in implementing the RR algorithm. The proposed work is much better than traditional RR in addition to use in cloud computing in which balance the carry by the side of the earliest as reply time becomes decreased effectually.

Pati et al. 2017 [11]. Purpose optimized for variation to the round robin algorithm. This paper depended on enhancement received the shortages of the current scheduling in standings of waiting and turnaround times with throughput and amount of context alterations. The modification RR proved to be efficient more than the traditional RR secluding because the modified RR algorithm takes 23.94 seconds compare than the traditional that takes 35.19 seconds to whole the implementation.

Srilatha et al. [12]. This paper depended on time quantum because one of the main problem in RR is the time quantum selection many research have been attempt in the previous to select an optimal time quantum. This paper based on Optimal Round Robin (ORRSM) determines the time quantum by captivating account the match or variances of the burst times of entirely processes. In proposed alter on the Round Robin Quantum rate by use the Manhattan Distance.

Tajwar et al. [13]. This paper present an enhanced RR algorithm for more effective in CPU scheduling and avoid the weaknesses on it. To evaluate its activity, the suggested RR algorithm was compared to the other enhanced RR algorithms called as:
•Improved round robin CPU scheduling algorithm with varying time quantum (IRRVQ).

•Self-adjustment time quantum in round robin (SARR).
•Modified round robin algorithm (MRR).

•Dynamic quantum with re-adjusted round robin scheduling algorithm (DQRRR).

The results designate an improved performance in standings of waiting time, turnaround time besides context exchanging through tested the burst time in ascending instruction.
tion has to be in sentense case with no spacing above or blow the srat of it.

3. PROPOSED WORK

Figures A problem of the standard round-robin scheduling algorithm has been solved and introduce a new proposed scheduling algorithm by decreasing the total of waiting time W.T and the turnaround time T.A.T to obtain a higher level efficiency of the CPU throughput. The core of the proposed algorithm is done when the tasks arrive at the ready, firstly, examining and calculate the number of tasks then re-arranged all tasks by starting to assign a priority to each task ascending, i.e. the lowest tasks value of CPU burst time given a highest priority according to the value of the burst time at a ready queue. Yet, this proposed method plays a significant role in improving overall performance and throughput of CPU utilization. Figure 1 show the block diagram of the proposed work.

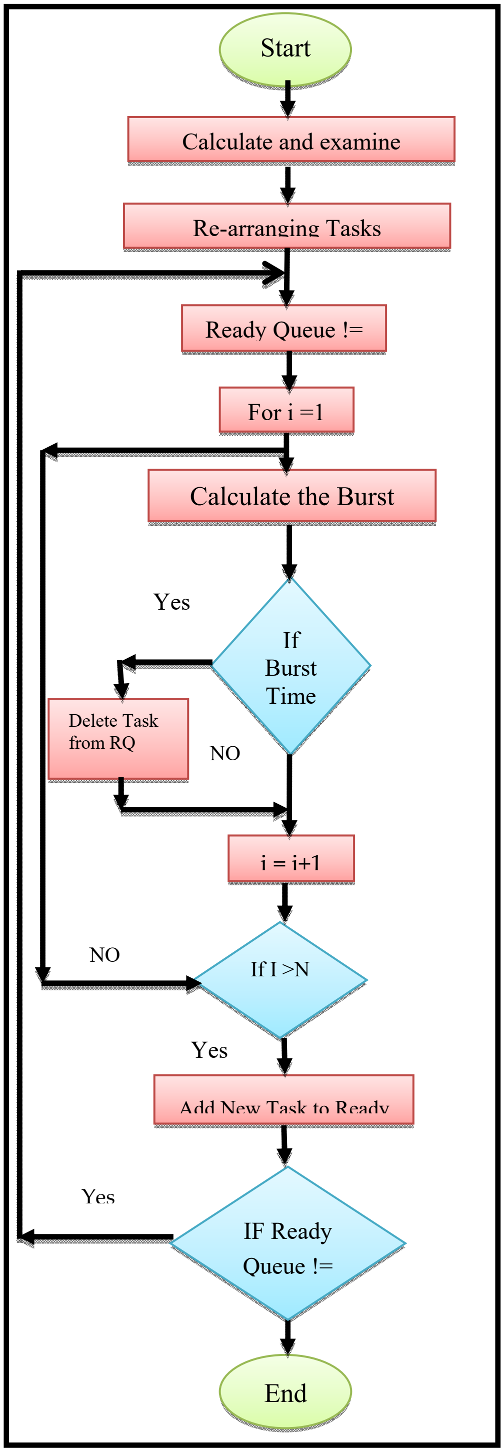

Figure 1. Flowchart of The Proposed Work

principle work of the proposed algorithm procedure is shown in algorithm

The Algorithm of Proposed work

```
Input:    N, BT, AT, Priority, TQ
Output:   Table of running Tasks, Gantt Chart, Average W.T, Average T.A.T.

1:  Arrange all the Tasks in RQ in order of their the lowest value of BT a highest priority;
2:  While RQ is not Null;
3:  For I equal 1 to N;
4:  Calculate the BT of the Tasks;
5:  Delete the task from RQ if the BT equal 0;
6:  If the new tasks arrived RQ go to Step 2;
7:  Draw Table of running Tasks, Gantt Chart and Calculate Average W, Average T.A.T;
8:  End while.

RQ:   Ready-Queue.
BT:   Burst-Time.
N:    Number of Tasks.
TQ:   Time-Quantum.
```

The work of the proposed algorithm is as follows:

Step 1: Allocate CPU to each task in standard round-robin algorithm, new priorities are allocated to all arrived tasks according to given the lowest value of burst time the highest priority with the same value of time quantum time. Then all task get the control of the CPU according to the new priority based on the remaining CPU bursts time in case the ready queue is not empty.

Step 2: After completing the step1, calculate the burst time of all tasks and re-arranged again with the new priority until the value of a burst time equal to zero.

4. RESULTS AND DISCUSSION

To demonstrate the performance of this work, several of tasks are used on a standard round robin algorithm and the proposed algorithm. five tasks were used to prove the proposed algorithm, creating a ready queue with the five tasks named T1, T2, T3, T4 and T5. The burst time of these tasks were 20, 42, 60, 37, 45, and 55 respectively with 20 MS of the time quantum in an equal and different arrival time In this paper, new priorities were added to all tasks which are 0, 2, 5, 1, 3 and 4 respectively; the

lowest value of the priority indicates to the highest priority as shown in Table 1.

Table 1: Tasks Specifications

Tasks Name	Arrival Time	Burst Time	Priority
T_1	0	24	3
T_2	0	29	2
T_3	0	86	4
T_4	0	12	0
T_5	0	19	1

In this paper, two cases are applied to test enhance of the proposed algorithm:

In the first case, firstly, we assume all tasks arrival with the zero time. In this case, the standard round-robin has arranged all tasks according to the time quantum without the feature of the priority, all tasks of standard round-robin algorithm are executed as illustrated in Table 2 and distribution is shown in Figure 2.

Table 2: Tasks Specifications After Executed

Name of Tasks	CPU Time	Arrival Time	W.T	T.A.T
T1	24	0	71	95
T2	29	0	75	104
T3	86	0	84	170
T4	12	0	60	72
T5	19	0	72	91

Average of W.T: 72.4ms
Average of TAT: 106.4ms

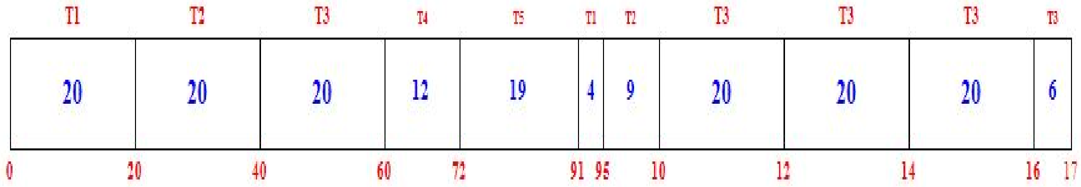

Figure 2. Gantt Chart of Standard Round-Robin Algorithm

Secondly, we applied all tasks in ready the queue according to a proposed algorithm by adding the highest priority value to the lowest value of CPU burst time respectively, all tasks are re-arranged again according to their priorities as shown in Table 3 and Gantt chart is shown Figure 3.

Table 3. *Tasks After Executed with Priority Feature*

Name of Tasks	CPU Time	Arrival Time	Priority	W.T	T.A.T
T1	24	0	3	71	95
T4	12	0	0	20	32
T5	19	0	1	32	51
T2	29	0	2	75	104
T3	86	0	4	84	170

Average of W.T: 56.4ms.

Average of T.A.T: 90.4ms.

algorithm, we obtained a high unites of saved time for the average of W.T and T.A.T. that shown in Table 4.

Nevertheless, Figure 4 shows the enactment comparison of tasks between the standard round-robin algorithm at ready queue with the same arrival time.

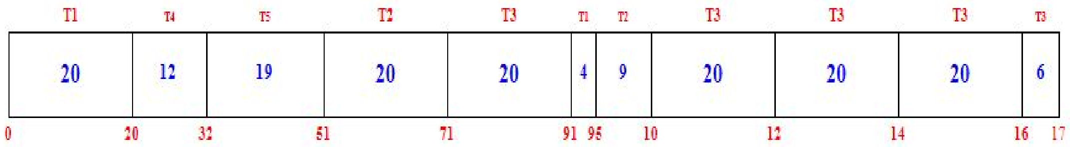

Figure 3. Gantt chart of proposed algorithm

Table 4. A high Unites of Saved Time

Performance Element	Standard Round-Robin Scheduling	Proposed Algorithm	State
Average W.T	72.4ms	56.4ms	16 Unites of Saved Time
Average T.A.T	106.4ms	90.4ms	16 Unites Saved Time

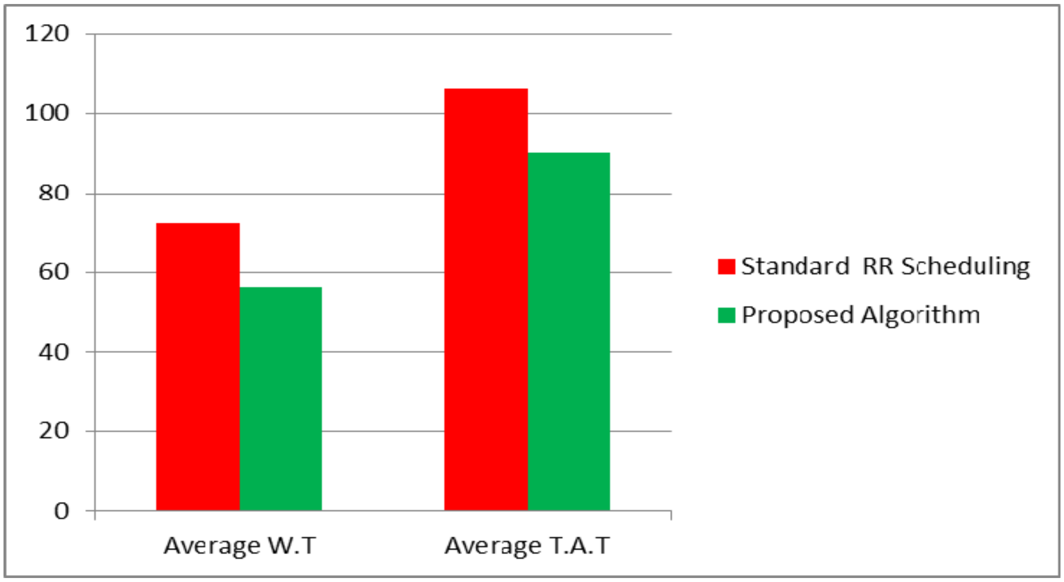

Figure 4. Performance comparison for Case 1

Table 5. Tasks Specifications

Name of Tasks	Arrival Time	Burst Time	Priority
T1	1	24	3
T2	3	29	2
T3	5	86	4
T4	2	12	0
T5	0	19	1

In the second case, all tasks have been repeated again with different arrival time which are 1, 0, 4, 2, 3, and 5 respectively to confirm efficiently of the proposed algorithm. Table 5 shows the specification of five tasks at a ready queue.

The execute of five tasks with their period time for standard round-robin in Table 6 and Gantt chart is shown Figure 5.

Table 6. Tasks Specifications After Executed with Different Arrival

Name of Tasks	CPU Time	Arrival Time	W.T	T.A.T
T5	19	0	0	19
T1	24	1	70	94
T4	12	2	37	49
T2	29	3	72	101
T3	86	4	79	165

Average of W.T: 51.6ms.

Average of T.A.T: 85.6ms.

Figure 7 shows the enactment comparison of tasks between the standard round-robin algorithm and the proposed algorithm at ready queue with the different arrival time.

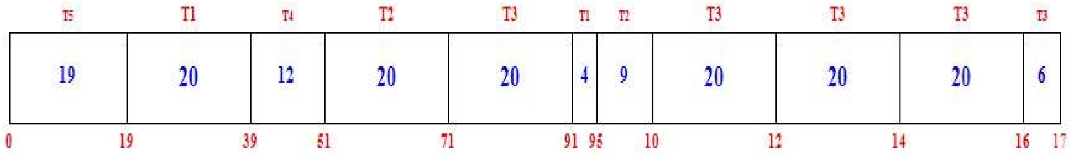

Figure 5. Gantt Chart of Standard Round-Robin Algorithm

The execute of five tasks with their period time for the proposed algorithm Table 7 and Gantt chart is shown Figure 6.

we need to comparative between the proposed algorithm and the most recent of previous work as [12][10][8][7] to evaluate the efficiency percentage of this proposed. Its applied the same values of previous work in our proposed algorithm.

Table 7. Tasks After Executed with Priority Feature

Name of Tasks	CPU Time	Arrival Time	Priority	W.T	T.A.T
T5	19	0	1	0	19
T4	12	2	0	17	29
T5	29	3	2	68	97
T1	24	1	3	79	103
T3	86	4	4	79	165

Average of W.T: 48.6ms.

Average of T.A.T: 82.2ms.

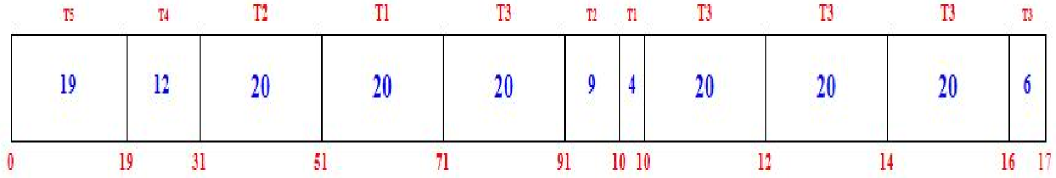

Figure 6. Gantt Chart for Proposed Algorithm

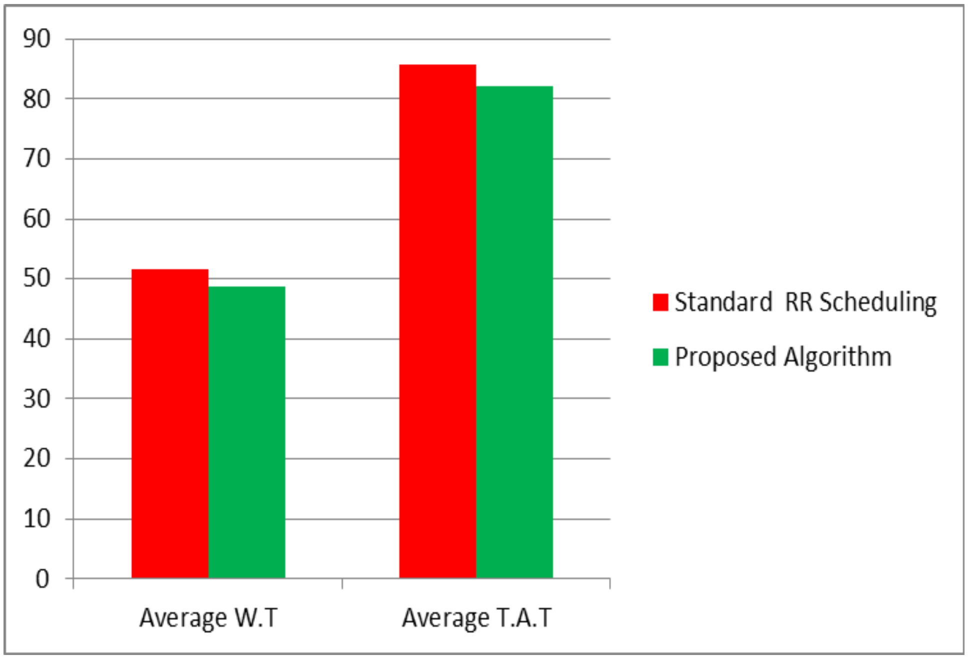

Figure 7. Performance comparison Case II

- Optimal RR CPU Scheduling Algorithm By Manhattan Distance[12].

Average of WT: 44.8ms;

Average of TAT: 64.4ms.

Gantt Chart

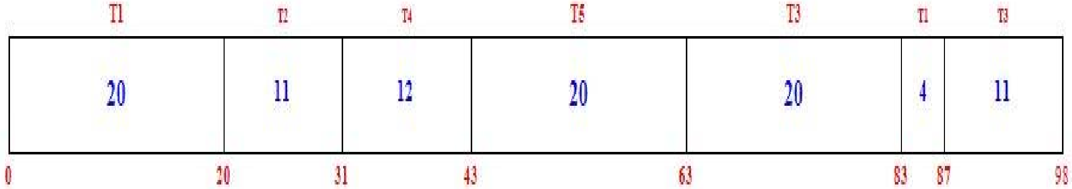

Figure 8. Optimal Round-Robin Algorithm

- Improved round-robin scheduling [10].

Gantt Chart

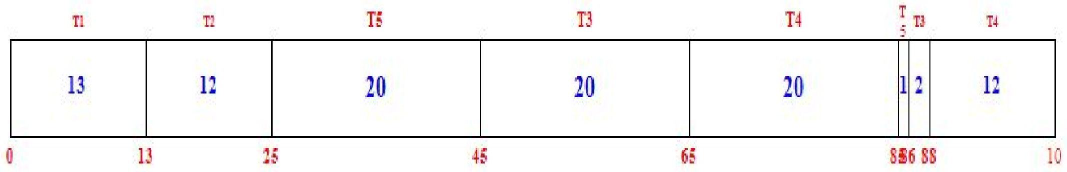

Figure 9. Gantt chart of improved round-robin algorithm

Average of WT: 42.4ms.

Average of TAT: 62.4ms.

Integer Programming [8].

Gantt Chart

Average of WT: 36 ms

Average of TAT: 84.6 ms

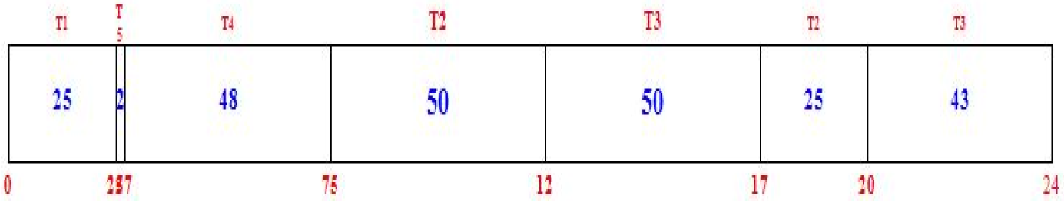

Figure10. Gantt chart of Integer Programming

- An Optimized Round Robin Algorithm [7].

Gantt Chart

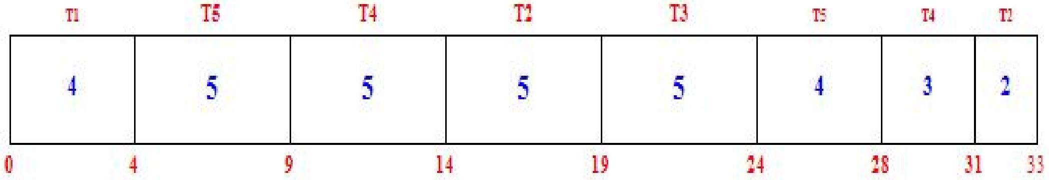

Figure 11. An Optimized Round Robin Algorithm

Table 8 shows summarized of the most recent previous works as we can see they are organized according to the published date.

Table 8. Comparison Between Previous Studies Results with Proposed Algorithm

Performance Feature	Average W.T	Average W.T When Applied in Proposed Algorithm	Average T.A.T	Average T.A.T When Applied in Proposed Algorithm
Srilatha et al. [12]	52.8ms	44.8ms	72.4ms	64.4ms
Sangwan et al. [10]	44.8	42.4	64.8	62.4
Kumar et al.[8]	71.6ms	36ms	119.8ms	84.6ms
Singh et al.[7]	17ms	13.2ms	23.2ms	19.8ms

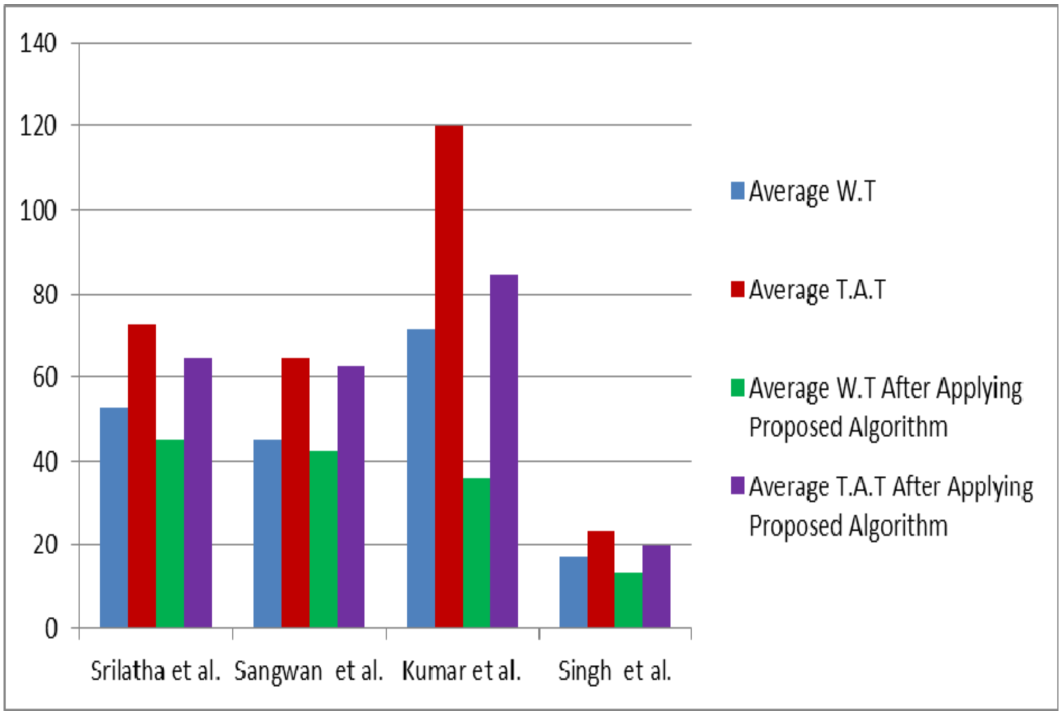

Figure 12. Comparison Between Previous Studies with Proposed Algorithm

According to the end results as shown in previous Table we can see that the proposed algorithm more efficient from the these previous work. Figure 12 show the summary of comparison between our proposed algorithm with the previous proposed algorithms.

5. CONCLUSIONS

In this paper we have proposed a smart enhancement round robin scheduling for operation system.

The diagrams and effect of proposed round robin algorithm provides better results in parameters of average waiting time(A.W.T), turnaround time (T.A.T) in entirely cases of tasks classifications than the standard Round Robin algorithm. The main problem in round robin scheduling algorithm is the select of quantum time. In this proposed algorithm the performance of a novel method based on priority for enhancement round-robin scheduling algorithm is higher than that of Round Robin algorithms when compare between them. The result show the average waiting time in proposed work 56.4ms while the standard 72.4ms also turnaround time 90.4 ms when standard 106.4ms these result appearance the efficient of the proposed with 16 unites to saved CPU time. On other hand when compared the proposed work with a set of previous study the result show the maximum unites saved CPU time of the average waiting time and turnaround time in proposed work 35 ms while the minimum unites saved is 2ms, these led to present the performance of the proposed more than these studies. In a future work can be improvement and developed the smart enhancement Round Robin scheduling for more accurate and performance for all criteria of CPU scheduling.

REFERENCES

[1] Allkoci, A.; Dhima, E.; Tafa, I. Comparing Priority and Round Robin Scheduling Algorithms. *International Journal of Computer Scienc*. 2014, 11,175-181.

[2] Joshi, R.; Tyag, S. Smart Optimized Round Robin (SORR) CPU Scheduling Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering. 2015, 7, 568-574.

[3] Abdalkafor, A. S. (2017). Designing Offline Arabic Handwritten Isolated Character Recognition System using Artificial Neural Network Approach. *International Journal of Technology*, *8*(3), 528-538.

[4] Abdalkafor, A. S., Aiman , M.N & Mustafa, N.O. Artificial neural network To predict the success rates of schools. *Journal of Theoretical and Applied Information Technology* 12, no. 96 (2018): 1-11. (In Preparation).

[5] Abdalkafor, A. S & Sadeq, A. Arabic Offline Handwritten Isolated Character Recognition System Using Neural Network. *International Journal of Business and ICT, 2*(3), 2016, pp. 41-50.

[6] Belwal, M.; Kumar, S. A Priority Based Round Robin CPU Scheduling Algorithm. *International Journal of Computer Science and Information Technologies*. 2017, 8, 0975-9646.

[7] Singh, A;, Goyal, P.; Batra, S. An optimized round robin scheduling algorithm for CPU scheduling. *International Journal on Computer Science and Engineering*. 2010, *2*, 2383-2385.

[8] Kumar, D. P.; Reddy, T. S.; Reddy, A. Y. Finding Best Time Quantum for Round Robin Scheduling Algorithm to avoid Frequent Context Switch. *International Journal of Computer Science and Information Technologies*. 2014, 5.

[9] Kumar, M.; Rashid, F. An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum. *International Journal of Computer Science, Engineering and Applications*. 2014, 4, 1-8.

[10] Sangwan, P.; Sharma, M.; Kumar, A. Improved Round Robin Scheduling in Cloud Computing. *Advances in Computational Sciences and Technology*. 2017, 4, 639-644.

[11] Pati, M. N. S. ; Korde, P.; Dey, P. An advanced approach to traditional round robin CPU scheduling algorithm to prioritize processes with residual burst time nearest to the specified time quantum. *Materials Science and Engineering*. 2017, 263.

[12] Srilatha, N. ; Sravani, M. ;Divya, Y. Optimal Round Robin CPU Scheduling Algorithm Using Manhattan Distance. *International Journal of Electrical and Computer Engineering (IJECE)*. 2017, *7*, 3664-3668.

[13] Tajwar, M. M. ; Pathan, M. ; Hussaini, L. ; Abubakar, A. CPU Scheduling with a Round Robin Algorithm Based on an Effective Time Slice. *Journal of Information Processing Systems. 2017,* 13 , 941-950.