

A Detail Analysis on AWT and ATT for Different CPU Scheduling Algorithms for Different Cases

Dr. Biswarup Samanta ^a, Arinjay Biswas ^b

a Assistant Professor (II), Department of Computer Science, Amity University Jharkhand (www.amity.edu/ranchi), Ranchi, India, biswarup.k@gmail.com; bsamanta@mc.amity.edu; ORCID Id: <https://orcid.org/0000-0002-9015-3136>

b. Graduate Trainee, Tata Consultancy Services (www.tcs.com), Chennai, India, biswas87280@gmail.com

Abstract: The sequence of execution of processes in multiprogramming plays a very crucial role to optimize the utilization of CPU. It has been found that CPU Scheduling algorithm serve as the basis of the multiprogramming. This paper presents the relationship between existing algorithms w.r.t A.W.T and A.T.A.T. through different cases. There are 5 cases where each case has 4 parts and each part represents different existing algorithms i.e. FCFS, SJF, RR, PB. Each part consists of 4 processes with different or same priority and arrival time depending upon the particular case, as designed in the section 4 of this paper. The goal of this paper is to find out the relationship w.r.t. A.W.T. and A.T.A.T., when different cases are implemented on the existing algorithms and find the fastest algorithm amongst the existing algorithms taken in this paper.

Case Formulation-

- 1) In case 1 the CPU burst taken are 4, 3, 5 and 2 for all the respective 4 processes.
- 2) In case 2 the CPU burst taken is an addition of 1 from the CPU burst of case 1 of each processes.
- 3) In case 3 the CPU burst taken is subtraction of 2 from the CPU burst of case 2 of each processes.
- 4) In case 4 the CPU burst taken is an addition of 3 from the CPU burst of case 3 of each processes.
- 5) In case 5 the CPU burst taken is the addition of 1 from the CPU burst of case 4 of each processes.

Keywords: CPU Scheduling, FCSF, SJF, RR, PB, Throughput

1. Introduction

The scheduling is used to allocate system resources to the active processes (**Biswarup Samanta and Biresh Kumar 2017**). Operating system performs variety of tasks in which scheduling of CPU is one of the basic task (**Rakesh Patel and Mrs. Milli. Patel 2013**). In early day's operating system, the user could execute a single process at a particular point of time. This process of execution of single process one by one, was time consuming and due to this, the CPU utilization was also not optimized. So, then the multiprogramming concept evolved, where the user could now run more than one program simultaneously, which saved the CPU time and the throughput of the system was also increased. The objective of the multiprogramming is to have some process running all times, to maximize CPU utilization (**Silberschatz, A. P.B. Galvin and P.B. Galvin and G.Gagne 2012**).

2. Short, Mid and Long Term Scheduler

Multiprogramming increases CPU utilization by running more than one processes. A process is defined as the run time instance of a program. Scheduler is the key module of the operating system to provide the multiprogramming environment in the computer system. The performance and throughput of the system can only be increased, if the CPU is scheduled intellectually. CPU scheduling is the task of selecting a process from the ready queue and allocating the CPU to it (**Amar Ranjan Dash, et. al., 2015**).

3. Review Of Algorithms

First come first serve - The most intuitive and simplest technique is to allow the first process submitted to run first (**Neetu Goel and Dr. R.B.Garg 2012**). In this algorithm processes arriving first are executed first.

Shortest job first- In this algorithm the processes are sorted in the ready queue in the increasing order of their CPU Burst time. Processes with least CPU Burst time are executed first and so on. If two processes having the same CPU Burst Time FCFS is used to break up the tie (**Navjot Singh, et. el., 2018**).

Round Robin – In this scheduling algorithm the first process in queue is executed first but a time slice is introduced in this algorithm. The time slice determine the maximum amount of time a process can hold the CPU at a stretch and after that the process need to wait for its turn again, if not finished it’s execution. The round-robin (RR) scheduling is designed especially for time-sharing systems (N. Mittal, et al., 2015).

Priority based- In this algorithm each process present in the queue have a priority assigned to it. The processes are executed as per their priority. The processes having high priority are executed first. Whenever any new process arrives in the ready queue having high priority as compared to the current process being executing, then the current process is interrupted and the high priority process is executed first and then again the process with next highest priority is executed. Waiting time and response time depend on the priority of the process (Ishwari Singh Rajput, et. al., 2012).

4. Design and Study Of Different Cases

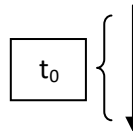
We have considered the ready queue with four processes P1, P2, P3 and P4 with respective CPU Burst time and priority as per the cases. These four processes are scheduled using FCFS, SJF, RR and PB algorithms. The average WT (AWT), average T.A time (ATAT) has been calculated and the results have been analysed at the end. We have taken 5 different cases and applied all the 4 algorithms to them, for our study as follows:

Table.1. Different Cases of CPU Scheduling

Case	Part	Algorithm	Time Quantum	Priority Limit	Priority Range
1	1	FCFS	NA	1 to 3	2
	2	SJF	NA	1 to 3	2
	3	RR	2	1 to 3	2
	4	PB	NA	1 to 3	2
2	1	FCFS	NA	1 to 3	2
	2	SJF	NA	1 to 3	2
	3	RR	3	1 to 3	2
	4	PB	NA	1 to 3	2
3	1	FCFS	NA	1 to 3	2
	2	SJF	NA	1 to 3	2
	3	RR	2	1 to 3	2
	4	PB	NA	1 to 3	2
4	1	FCFS	NA	1 to 3	2
	2	SJF	NA	1 to 3	2
	3	RR	3	1 to 3	2
	4	PB	NA	1 to 3	2
5	1	FCFS	NA	1 to 4	3
	2	SJF	NA	1 to 4	3
	3	RR	3	1 to 4	3
	4	PB	NA	1 to 4	3

CASE 1:

Table.2. Input for the processor for CASE 1



Process ID	CPU burst time	Priority
P1	4	2
P2	3	1
P3	5	3
P4	2	1

Part 1: FCFS and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 5.75 ms, Avg. Turnaround Time: 9.25 ms

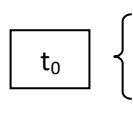
Part 2: SJF and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 4 ms, Avg. Turnaround Time: 7.5 ms

Part 3: Time quantum 2 RR and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 7.25 ms, Avg. Turnaround Time: 10.75 ms

Part 4: PB and Priority Range 3 i.e. from 1 to 3
 Avg. Waiting Time: 4.25 ms, Avg. Turnaround Time: 7.75 ms

CASE 2:

Table.3. Input for the processor for CASE 2



Process ID	CPU burst time	Priority
P1	5	2
P2	4	1
P3	6	2
P4	3	3

Part 1: FCFS and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 7.25 ms, Avg. Turnaround Time: 11.75 ms

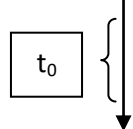
Part 2: SJF and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 5.5 ms, Avg. Turnaround Time: 10 ms

Part 3: Time quantum 3 ms RR and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 10.25 ms, Avg. Turnaround Time: 14.75 ms

Part 4: PB and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 7 ms, Avg. Turnaround Time: 11.5 ms

CASE 3:

Table.4. Input for the processor for CASE 3



Process ID	CPU burst time	Priority
P1	3	1
P2	2	3
P3	4	3
P4	1	2

Part 1: FCFS and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 4.25 ms, Avg. Turnaround Time: 5.5 ms

Part 2: SJF and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 2.5 ms, Avg. Turnaround Time: 5 ms

Part 3: Time quantum 2 ms RR and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 4.75 ms, Avg. Turnaround Time: 6.25 ms

Part 4: PB and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 3.25 ms, Avg. Turnaround Time: 5.75 ms

CASE 4:

Table.5. Input for the processor for CASE 4

	Process ID	CPU burst time	Priority
t_0	P1	6	4
	P2	5	2
	P3	7	3
t_1	P4	4	1

Part 1: FCFS and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 8.5 ms, Avg. Turnaround Time: 14 ms

Part 2: SJF and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 7 ms, Avg. Turnaround Time: 12.5 ms

Part 3: Time quantum 3ms RR and Priority Range 3 i.e. from 1 to 3
 Avg. Waiting Time: 13 ms, Avg. Turnaround Time: 18.5 ms

Part 4: PB and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 7.25 ms, Avg. Turnaround Time: 12.75 ms

CASE 5:

Table.6. Input for the processor for CASE 5

	Process ID	CPU burst time	Priority
t_0	P1	7	2
	P2	6	4
t_1	P3	8	3
t_4	P4	5	1

Part 1: FCFS and Priority Range 3 i.e. from 1 to 4
 Avg. Waiting Time: 9 ms, Avg. Turnaround Time: 15.5 ms

Part 2: SJF and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 7.5 ms, Avg. Turnaround Time: 14 ms

Part 3: Time quantum 3 ms RR and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 16.25 ms, Avg. Turnaround Time: 21.5 ms

Part 4: PB and Priority Range 2 i.e. from 1 to 3
 Avg. Waiting Time: 9 ms, Avg. Turnaround Time: 15.5 ms

5. ANALYSIS

In this section we analyze the AWT and average TAT for all the cases mentioned above.

Table.7. Analyzing all the cases with respect to the FCFS, SJF, RR and PB

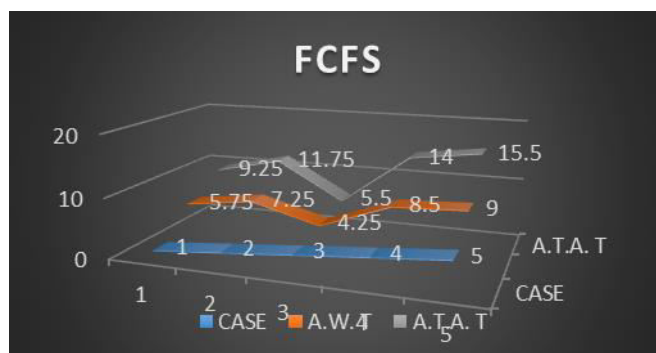
SCHEDULER	CASES																			
	CASE 1				CASE 2				CASE 3				CASE 4				CASE 5			
	C.P.U. BURST	PRIORITY	A.W.T	A.T.A.T	C.P.U. BURST	PRIORITY	A.W.T	A.T.A.T	C.P.U. BURST	PRIORITY	A.W.T	A.T.A.T	C.P.U. BURST	PRIORITY	A.W.T	A.T.A.T	C.P.U. BURST	PRIORITY	A.W.T	A.T.A.T
FCFS	4	2	5.75	9.25	5	2	7.25	11.75	3	1	4.25	5.5	6	4	8.5	14	7	2	9	15.5
SJF	3	1	4	7.5	4	1	5.5	10	2	3	2.5	5	5	2	7	12.5	6	4	7.5	14
RR	5	3	7.25	10.75	6	2	10.25	14.75	4	3	4.75	6.25	7	3	13	18.5	8	3	16.25	21.5
PB	2	1	4.25	7.75	3	3	7	11.5	1	2	3.25	5.75	4	1	7.25	12.75	5	1	9	15.5

CASE 1: FCFS

Table.8. The A.W.T and A.T.A.T of the FCFS scheduler

CASE	A.W. T	A.T.A. T
1	5.75	9.25
2	7.25	11.75
3	4.25	5.5
4	8.5	14
5	9	15.5

Figure.1. 3-D lines representing the above FCFS table

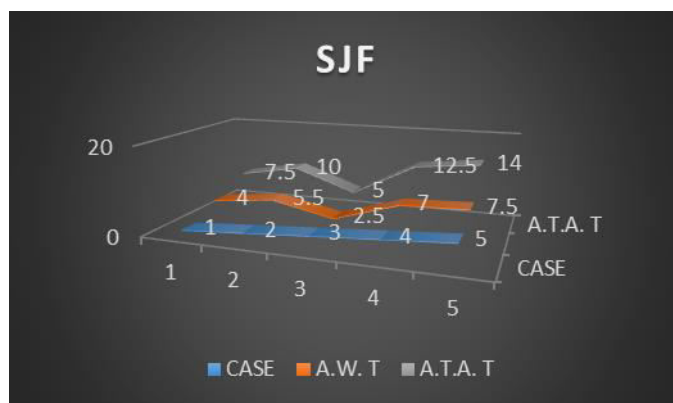


CASE 2: SJF

Table.9. The A.W.T and A.T.A.T of the SJF scheduler

CASE	A.W. T	A.T.A. T
1	4	7.5
2	5.5	10
3	2.5	5
4	7	12.5
5	7.5	14

Figure.2. 3- D lines representing the above SJF table

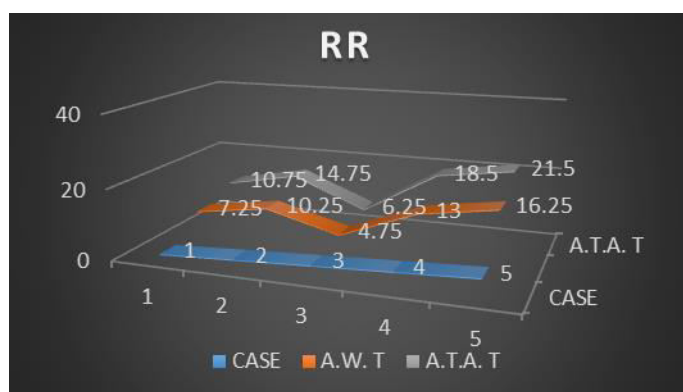


CASE 3: RR

Table.10. The A.W.T and A.T.A.T of the RR scheduler

CASE	A.W. T	A.T.A. T
1	7.25	10.75
2	10.25	14.75
3	4.75	6.25
4	13	18.5
5	16.25	21.5

Figure.3. 3-D lines representing the above RR table

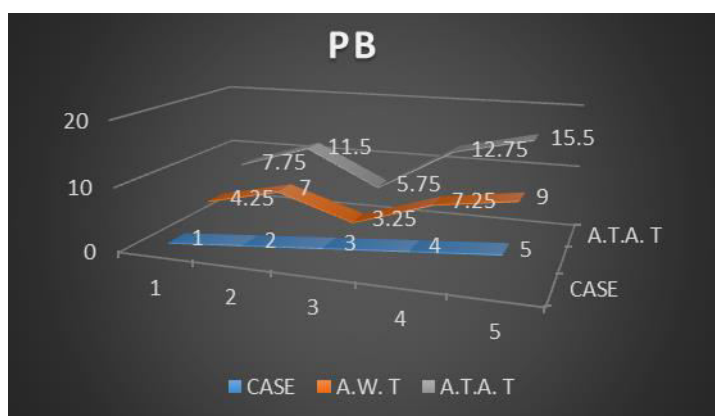


CASE 4: PB

Table.11. The A.W.T and A.T.A.T of the PB scheduler

CASE	A.W. T	A.T.A. T
1	4.25	7.75
2	7	11.5
3	3.25	5.75
4	7.25	12.75
5	9	15.5

Figure.4. 3-D lines representing the above PB table



6. Conclusion

While going through our study in this paper, it has been observed that for all the cases presented here in this paper that, the average waiting time (A.W.T.) and the average turnaround time (A.T.A.T.) is increasing when the CPU burst time is increased and then decreasing when the CPU burst time is decreased in each case. And hence it can be said that CPU burst time is directly proportional to the average waiting time and average turnaround time for all the given cases. Among the 4 given algorithms, it has been observed that for every case the SJF algorithm is having the lowest average waiting time and average turnaround time. Hence, it is the most efficient algorithm as per the results received.

References

- Amar Ranjan Dash, Sandipta kumar Sahu and Sanjay Kumar Samantra (2015), "AN OPTIMIZED ROUND ROBIN CPU SCHEDULING ALGORITHM WITH DYNAMIC TIME QUANTAM", International Journal of Computer Science, Engineering and Information Technology (IJCEIT), Vol. 5, No.1, February 2015.
- Biswarup Samanta and Bires Kumar (2017), "A Comparative Study on PBRR CPU Scheduling Algorithm for Different Time Quantum and Priority", International Journals of Advanced Research in Computer Science and Software Engineering, ISSN: 2277-128X (Volume 7, Issue-6), June 2017.
- Ishwari Singh Rajput and Deepa Gupta (2012), "A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems", International Journal of Innovation in Engineering and Technology(IJiet), Vol. 1 Issue 3 Oct 2012, ISSN: 2319 – 1058.
- Navjot Singh, Prabhjeet Kaur, Baljeet Kaur and Amandeep Kaur (2018), "A Comparative Analysis of Various CPU Scheduling Algorithms", Navjot Singh Int. Journal of Engineering Research and Application, ISSN : 2248-9622, Vol. 8, Issue4 (Part-III) April 2018, pp33-35.
- Neetu Goel and Dr. R.B.Garg (2012), "A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics & Image Processing, Vol. 2, Issue 4, November 2012.
- Neha Mittal, Khusbu Garg and Ashish America (2015), "A Paper on Modified Round Robin Algorithm", International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), Volume IV, Issue XI, November 2015.
- Rakesh Patel and Mrs. Milli. Patel (2013), "SJRR CPU Scheduling Algorithm", International Journal Of Engineering And Computer Science, ISSN:2319-7242, Volume 2 Issue 12, December.2013 Page No. 3396-3399.
- Silberschatz, A. P.B. Galvin and P.B. Galvin and G.Gagne (2012), Operating System Concepts, 8th Edition, Wiley India.