Journal of Asian Scientific Research
ISSN(e): 2223-1331/ISSN(p): 2226-5724
journal homepage: http://www.aessweb.com/journals/5003

TIME SHARING ALGORITHM WITH DYNAMIC WEIGHTED HARMONIC ROUND ROBIN

Mohammad Saber Iraji[1]

[1]*Faculty Member of Department of Computer Engineering and Information Technology, Payame Noor University (PNU), I.R. of Iran*

ABSTRACT

In operating system is an important issue, Allocates processor time to processes, to be fair. We presented in This paper proposed methods to reduce Average waiting time, average response time, Switch Counts process, through the design of a time sharing non-exclusive scheduling algorithm based on the round Robin and determined accurate time quantum . Among the proposed approaches, weight harmonic- dynamic weight and Subtraction Method are achieved better and more accurate results.

Keywords: Round robin, Time quantum, Scheduling, Operation systems, Waiting time, Response time.

Contribution/ Originality

This study contributes in scheduling time sharing for multi-processing environments. The paper's primary contribution is finding a method Increase Productivity non-exclusive scheduling algorithm.

1. INTRODUCTION

The operating system is a program which controls application program run and operates as an interface, between user and computer hardware.

One of the key functions of the operating system is the management of available resources (main memory), input / output devices, processors and Scheduling of their use by different active processes. In multitasking and multiprocessing environment the way the processes are assigned to run on the available CPUs is called scheduling. In multiprogramming systems which several applications are executed simultaneously there should be a fairly scheduling mechanism to share the CPU between processes [1]. The operating system must allocate computer resources among multiple potential rival processes . In the case of CPU allocated resource is the run time of CPU

DOI: 10.18488/journal.2/2015.5.3/2.3.131.142
ISSN(e): 2223-1331/ISSN(p): 2226-5724

and The allocation tool is scheduling. Practice scheduling must be designed in such a way that so many purposes, such as fairness, non-hungry processes , efficient use of processor time and little overhead cover. Sometimes, different levels of priority or real-time deadlines for the start or completion of some processes in real-time scheduling may be required. The Purpose of CPU scheduling, is process allocation for processors or processors over time so that it meet the system goals such as response time, throughput and CPU performance. The kinds of scheduling consist of the following cases.

1 - Long Term scheduling: to decide about adding to a set of processes to run.

2 - medium term scheduling: to decide about adding to a number of processes that some parts of them or all of them are in main memory.

3 - Short-term scheduling: to decide about which one of the processes in main memory is selected for implementation by the processor.

4 – Scheduling input /output : to decide about which one of the processes `s input / output requests Is done by an existing input / output device.

The scheduling basically consists of queue management to minimize queuing delays and optimize the performance in the queuing environment.

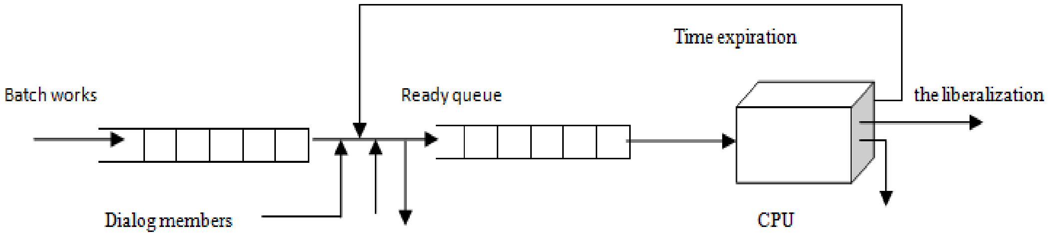

Fig-1. The Short-term scheduling diagram

Decision mode specifies the time that the selection function runs .We have two general classification: exclusive and non-exclusive. Exclusive: in this case, since a process is in implementation state, it continues to run until it ends or it becomes blocked itself for input / output waiting or requesting a service from the operating system .Non–exclusive: the running process can be stopped by the operating system and be transferred to the standby mode. Decision to be non-exclusive can be done when a new process enters, or when an interrupt takes a blocked process to ready state or periodically based on hour interrupt. Non-exclusive politics toward monopoly policies, applies more overhead but they provide better service for processes set Because they prevent CPU`s long monopolizing by a process. Besides, the non-exclusive price can be decreased by employing efficient text replacement strategies (with the maximum use of hardware) and sharing the large main memory, to keep up a large part of programs in main memory.

The scheduler attempts in order to minimize following items: Time of return, response time, waiting time for the processes and the number of context switching [2]

The round robin scheduling algorithm is one of the oldest non-exclusive algorithms .this algorithm, uses a small time unit called a time quantum or time slice that primary challenge in this algorithm is selection of proper time quantum.

It assigns a time interval for each process in the CPU `s ready queue as much as the time quantum. If the time quantum is large, the response time of the processes will be very large, that interactive environment can not tolerate and if the time quantum is small due to unnecessary duplication, switch contact size will be too much and totally a negative result is will obtain. The time quantum is being used in a static way in round robin algorithm that use of dynamic time quantum to improve the algorithm is suggested. 50 percent of the processes finish through the first round and time quantum is being calculated frequently for each round. then 50 percent of the remaining processes enter the second round and It will be calculated in the same way, also the third round ,the forth round .during the second round with .which means the maximum number of rounds will be less than or equal to 6. So processes or their burst time finish in the sixth round [3] An example of using Dynamic Time Quantum is Agha and Jassbi [4] that it has been provided in irregular processes and time Quantum in each cycle is calculated using the harmonic average.

The aim of this study is to examine and discuss about the short-term scheduling algorithms, this paper attempts to optimize the round robin algorithm. The paper is organized in six sections. After the introduction in Section 1, Section 2 which also introduces the related works of scheduling .Section 2 continues with short-term scheduling definitions in section 3. in Section 4 is describe scheduling algorithms types. It continues with Mathematical models for proposed algorithms and examples. Section 6 and 7 presents the results, conclusions of the research. The paper ends with a list of references.

2. LITERATURE

In recent years, many great works are done to improve the round Robin scheduling algorithm and to obtain response times and waiting times and the number of switches.

Examples are as follows:

Mr Ajit Singh, et al. [5] has used the defined static quantity “q” in the first cycle in the simple Round Robin and after completion of the first cycle he has doubled the amount of q and has sorted the burst's time remaining from small to large. then he completed The second cycle with the new Q, and these steps are repeated. Neete Goel, et all in Goel and Garg [6] to improve Round Robin emphasizes on making Q dynamic ,so first sorts burst's time amounts, then like [5] uses doubling the Q quantity. But the difference is Goel compares the burst time remaining and q2/2. if it is lesser, he uses completely, otherwise these steps will be repeated. Sanjaya Kumar Panda, et all in Panda and Bhoi [7] introduced the Min-Max Round Robin (MMRR) algorithm which is calculated as dynamic Q .he used The difference between the largest and smallest brush time. Sanjaya Kumar Panda ,et all in KumarPanda, et al. [8] introduces Group Based Time Quantum (GBTQ) algorithm .which calculates Q as like as Panda and Bhoi [7]. but The difference is the number of processes divide into 4 parts and each part should define a Q. Abbas Noon,et all in Noon, et al. [9] called his algorithm AN to find dynamic Q and used arithmetic mean in order to improve the simple Round Robin algorithm . Debashree Nayake,et all in Nayak, et al. [10] introduced Improved Round Robin(IRR)to calculate dynamic Q.

The Saroj Hiranwal , et all in Hiranwal and Roy [11] used two ways to improve the Round Robin algorithm to obtain the smart Time slice. In this paper, in the case that the burst time number is odd, the mid process burst time is being used, and when it is even the Average burst time is being used.

In Yaashuwanth and Ramesh [12] C.Yaashuwanth ,et all has used the Otc , Pc, Sc, Csc parameters to calculate the Intelligent time slice beside using the preference And he have achieved a Better result in comparison to the simple Round Rabin , and the Intelligent time slice for Round Rabin .In Behera, et al. [13] The Round Robin algorithm was modified by using dynamic ITS, and Shortest Remaining Time Next algorithm(SRTN) they achieved to the the time quantum quantity by calculating the ITC and OTS Parameters. In Mohanty, et al. [14] PROF. RakeshMohanty ,et all Provided Priority Based Dynamic Round Robin (PBDRR) algorithm ,in this algorithm unique intelligent time slice of processes is being calculated and then it changes every round run. In Srivastav, et al. [15] Asst.Proff.M.K.Srivastar, et all Has used Fair Priority Round Robin with Dynamic time Quantum(FPRRDQ) algorithm which used the defined priorities by the user and the weight ,to calculate the time quantum and he has Improved the priority based simple round robin (PBSRR) algorithm, and the Shortest Execution First Dynamic Round Robin algorithm (SEFDRR) .In Rajput and Gupta [16] Ishwari Singh Rajput ,et all use the defined priority and define anew priority to improve the round Robin algorithm by burst time remaining after the completion of first cycle .

H.S.Behera , et all in Behera, et al. [17] used the Precedence based Round Robin with Dynamic Time Quantum algorithm (PRRDTQ) . PFi parameter is defined to determine the priority and MRR and PBDRR algorithms are improved. In Al-Hagery [18] the author introduces Selective-Round Robin Quantum algorithm (SRRQT),which calculates switch counts, waiting time and time quantum for each process ion , and selects the best case finally so they improved the AN algorithm .

Many people have used different tools to improve the round robin algorithm such as fuzzy logic [19], [20]. neural networks [21],[22]., genetic algorithm and artificial intelligence [23],[24] is an example of fuzzy logic, they have made the Round Robin algorithm fuzzy by defining linguistic variables LNOP,LABT and they proved that this manner is better than the simple Round Robin algorithm by sorting the burst times.

In algorithms reviewed above, non-exclusive algorithms with dynamic time quantum are useful in optimize the short term scheduling parameters (Average waiting time, average response time, Switch Count).

3. THE SHORT-TERM SCHEDULING DEFINITIONS

The usual criteria's are divided in two dimensions: Criteria's of the user`s view point and Criteria's of the system `s view point. Criteria of the user`s view point, refers to the system `s behaviour as it seems to a user or a single process. Scheduling policy must be in such a way that provides good services to different users. Criteria of the system `s view point and relevant to the efficiency including: the total time, response time and the deadline.

3.1. Total Time

(Turn around –time) is the time range from accepting a process to the completion of it, this

3.2. Time Includes On

Real run time and the time spent to wait for resources (including CPUs). The total time is a proper criterion for batch jobs.

3.3. Response Time

The response time for an interactive process is the time interval from declaring a request until the start of getting the answer. The process often, begins to produce output when processing of that demand continues. So in the user's viewpoint, this Criterion is better than total time Criterion. Scheduling system should try to achieve the low time response and to maximize the number of interactive users, according to the acceptable response time.

3.4. Deadline

(Deadline) when the dead line for the completion of the process can be specified, the scheduling system must emphasize the other goals lesser in order to maximize the deadline satisfy percent.

3.5. Throughput

(Throughput)the Scheduling policy should try to maximize the number of completed processes at the same time .this criteria expresses how much work is done and it clearly depends on the average of process length but it is affected by Scheduling policy that can be impressive in efficiency .

3.6. CPU Usage

Indicates the percentage of time that the CPU is busy and its an important criterion for common expensive systems. In single-user systems and some other systems too, such as real-time systems, this criterion is less important.

3.7. Justice

(Justice) when user guides are absent or in the absence of guidance provided by the system, the processes should be treated equally and no process should suffer from hunger.

3.8. Priorities Apply

When processes have priorities, scheduling policy should prefer processes with higher priorities and perform them sooner than the other processes.

3.9. Resource Balancing

(Resource balancing) Scheduling policy should keep the system resources busy. The Processes that use very busy resources lesser should be prior to the others. This criteria includes on medium and long-term scheduling.

4. SCHEDULING ALGORITHMS TYPES

Different scheduling algorithms have been defined a number of them are in below:

4.1. The Entry Serving (FCFS)

The easiest Scheduling politic is Serving entry sequence or to exit in an entry sequence. Every process joins to the ready queue preparing. When the current process ceases to execute, the oldest process in the ready queue is being selected to run. FCFS acts much better than the short processes.

This scheduling algorithm is a proprietary algorithm.

4.2. Shortest Process (SPN)

This policy is proprietary. The Process which has the shortest expected processing time, is being selected for execution. So the short process passes long works and it is in the head of the queue.

4.3. Shortest Remaining Time (SRT)

It`s a kind of non-exclusive SPN, the scheduler always chooses the process that is expected to have the shortest remaining process time.

4.4. Round Robin (RR)

The simplest algorithm to reduce the penalty incurring short jobs in FCFS is to use round Robin. An hour interrupt is being generated in periodical time intervals. If the Interrupt occur the running process is put in the ready queue and the next ready work is chosen based on the FCFS. This method is also known as a period because each process gives a period before it is taken over a given. The main design problem in the Round Robin is the time quantum quantity. If this quantity is too short, the short processes pass the system is almost quickly .on the other hand, the processing overhead in clock managing the and the Schedule run and distribution operation of the processes can be observed. So the very low quantity should be avoided. A useful guideline is that the period is should be a little longer than the required time for a conventional dialog. Note that the Q quantity is more than the largest (longest) greater process the Round Robin algorithm declines to FCFS. One of the notable features of the Round Robin algorithm is that it is non-exclusive.

5. MATHEMATICAL MODELS

Proposing the mathematical model of some offered methods in studied papers with proposed techniques:

5.1. To Calculate Dynamic Q by Harmonic Mean of the and Irregular Data. (HARM Algorithm)

QT= (n/ ((1/t_1)+(1/t_2)+…(1/t_n)) (1)

5.2. To Calculate the Dynamic q by the Arithmetic Mean (AN algorithm) [9]

QT=AVG ($\sum_{i=1}^{n} BT(p_i)$) (2)

5.3. To Calculate the Dynamic Q by the Arithmetic Mean (Proposed)

QT=$\sqrt[n]{BT(p_1) * BT(p_2) * \ldots * BT(p_n)}$ (3)

5.4. To Calculate the Dynamic Q by the Median [3]

$$QT=\begin{cases} Y_{(n+1)/2} & \text{if n is odd} \\ 1/2(y_{\frac{n}{2}} + y_{1+(\frac{n}{2})}) & \text{if n is even} \end{cases} \quad (4)$$

5.5. To Calculate the Dynamic q by Three Quarters. (MDTQRR Algorithm) [25]

$$MQ =\begin{cases} y_{(n+1)/2} & \text{if n is odd} \\ 1/2\ (y_{n/2} + y_{(1+n)/2}) & \text{if n is even} \end{cases} \quad (5)$$

5.6. To Calculate the Dynamic q by Averaging the Middle and the Biggest Burst Time (IRR Algorithm) [10]

$$Median(x) =\begin{cases} y_{(n+1)/2} & \text{if n is odd} \\ 1/2\ (y_{n/2} + y_{(1+n)/2}) & \text{if n is even} \end{cases} \quad (6)$$

QT= (highest (BT) + median(M))/2 (7)

5.7. To Calculate the Dynamic Q by the Average, Harmonic Mean, and the Biggest Burst Time. (Proposed)

M=1/(1/BT(P_1))+(1/BT(P_2))+…+(1/BT(P_n)) (8)

QT= (highest BT+ M)/2 (9)

5.8. To Calculate the Dynamic Q by Weight Mean (Proposed)

QT=($\sum_{i=1}^{n} BT(P_i) W_i$)/$\sum_{i=1}^{n} W_i$ (10)

5.9. To Calculate the Dynamic Q by the Average, the Weight Average, and the Biggest Burst Time. (Proposed)

M=($\sum_{i=1}^{n} BT(P_i) W_i$)/$\sum_{i=1}^{n} W_i$ (11)

QT= (highest BT+ M)/2 (12)

5.10. To Calculate the Dynamic Q with $\sqrt[2]{(x_1{}^2 + \cdots + x_n{}^2)/n}$ Relation. (Proposed)

QT= $\sqrt[2]{\left(BT(p_1)\right)^2 + \cdots + (BT(p_n))^2)/n}$ (13)

5.11. To Calculate the Dynamic Q by the Average, Arithmetic Mean and the Biggest Burst Time. (Proposed)

M=(BT(P_1)+BT(P_2)+….+BT(P_n))/n (14)

QT= (highest BT+ M)/2 (15)

5.12. The Subtraction of Max and Min. (Proposed)

M=max burst time –min burst time (16)

5.13. To Calculate the Dynamic Q by Subtracting Average of Min and Max Biggest Burst Time. (Proposed)

M=max burst time –min burst time (17)

QT= (highest BT- M)/2 (18)

5.14. To Calculate the Dynamic Q by Harmonic Weight (Proposed)

QT= $\sum_{i=1}^{n} W_i / (\sum_{i=1}^{n} W_i / BT(P_i))$ (19)

5.15. To Calculate the Dynamic Q by the Average, the Geometric Mean and the Biggest Burst Time (Proposed).

M= $\sqrt[n]{BT(P_1) * BT(P_2) * \ldots * BT(P_n)}$ (20)

QT= (highest BT- M)/2 (21)

6. EXPERIMENTAL RESULTS

We solved all mentioned items in our proposed methods, during this search for Example 3 at Behera, et al. [25] and we offered the results in a table for a better comparison.

according to the table`s results: The proposed methods are “Geometric mean” and “Geometric mean with the biggest burst time “. the waiting time average quantity ,response time average and the context switching count in the” geometric mean with the biggest burst time” method, from right to left are 26.2,58.6,and 5, and for “Geometric mean method”,32.4,64.8,and 7 are calculated. although the waiting time average and the response time average in “Geometric mean with the biggest burst time “method are lesser than “Geometric mean” method but this algorithm is near to be exclusive because its switching count is lesser. so the” Geometric mean” method have more efficiency in multiple program environments because it is Non-exclusive. then ,according to the Considerations and the obtained results in table1 ,it showed that proposed methods :{1) Geometric mean 2)ordered harmonic 3) Weight average with subtraction weight and dynamic weight 4)weight harmonic with subtraction weight and dynamic weight and also 5) Weight average with dynamic weight with priority defining without considering the entry time} have more optimal parameters for waiting time average and response time average .(table 2)

After study of the Summarized results in table 2, weight harmonic algorithm ,dynamic weight, and Subtraction, with waiting time average quantity, [25]the response time average ,and 7,61,4,29 switch count have the best efficiency.

Table-1. Result comparison table for existence, proposed methods

Algorithm	Time quantum	waiting time average	response time average	Switch Count
Proposed : Geometric mean	7,15,31,25,34	32.4	64.8	7
Proposed : Mean, geometric mean, with the largest burst time	7,15,60,30	26.2	58.6	5
AN [9], arithmetic mean [9]	7,15,47,43	26.2	58.6	5
Proposed : Mean, arithmetic mean, with the largest burst time	7,15,68,22	26.2	58.6	5
Max-Min [7]	7,15,82,8	26.2	58.6	5
Proposed : Min - Max timeout Average with the largest burst time	7,15,86,4	26.2	58.2	5
Irregular harmony [4] HARM algorithm	7,18,36,36	44.2	76.6	7
Proposed : ordered harmonic	7,15,19,35,36	30	62.4	7
Proposed : : ordered harmonic with the largest burst time	7,15,54,36	26.2	58.6	5
Middle or middle average [3]	7,15,42,48	26.2	58.6	5
Mean , middle with the largest burst time , IRR [10]	7,15,66,24	26.2	58.6	5
The middle and ¾ [25] MDTQRR algorithm	7,15,42,90	26.2	58.6	4
Proposed : $\sqrt[2]{(x_1{}^2+\cdots+x_n{}^2)/n}$	7,15,58,32	26.2	58.6	5
Mean with the largest burst time	7,15,74,16	26.2	58.6	5
Proposed : Weight mean with Subtraction weight and dynamic	7,15,35,12,43	33.2	65.6	7
Proposed: Average, weight mean difference of with the largest burst time and Subtraction weight and dynamic.	7,15,62,28	26.2	58.6	5
Proposed : Weighted mean with Subtraction weight and fixed	7,15,35,23,32	33.2	65.6	7
Proposed : Weight mean of dynamic weight ,(1/ entry time)+ (Subtraction/ total burst time)	7,15,43,47	26.2	58.6	5
Proposed : Weight mean of fixed weight ,(1/ entry time) + (Subtraction/ total burst time)	7,15,43,47	26.2	58.2	5
Proposed : weight harmonic, dynamic weight and Subtraction	7,15,14,34,42	29	61.4	7
Proposed : weight harmonic , fixed weight and Subtraction	7,15,14,36,40	29	61.4	7
Proposed : weight harmonic, dynamic weight and (1/ entry time)+(Subtraction/ total burst time)	7,15,15,34,41	29.2	61.6	7
Proposed : Weight mean of dynamic Weight, (1/ entry time) +(Subtraction/ total burst time) with Priority defining	7,15,43,47	26.2	58.6	5
Weight mean of dynamic Weight (1/ entry time) + (Subtraction/ total burst time) with Priority defining without considering the entry	7,31,22,37	31	63.4	7
Simple round robin	25	46.2	78.6	8

Table-2. Result comparison table for Optimum proposed methods

Algorithm	Time quantum	waiting time average	response time average	Switch Count
Geometric mean	7,15,31,25,34	32.4	64.8	7
ordered harmonic	7,15,19,35,36	30	62.4	7
Weight mean with Subtraction weight and dynamic	7,15,35,12,43	33.2	65.6	7
weight harmonic, dynamic weight and Subtraction	7,15,14,34,42	29	61.4	7
Weight mean of dynamic Weight,(1/ entry time)+(Subtraction/ total burst time) with Priority defining without considering the entry time for entry order.	7,31,22,37	31	63.4	7

7. CONCLUSION

Our purpose in this research is determining the exact amount of burst time, time quantum ,for round robin algorithm and creating an exclusive algorithm in multiprocessing environments .for this purpose, as yet ,many mentioned algorithms are offered in literature review ,and we suggested some methods too. between these methods in" weight harmonic, dynamic weight and Subtraction" offered algorithm, the amount of parameters: waiting time average, response time average ,and the switch count (with observing the exclusive condition) are lesser than the existing round robin algorithms and the performed calculations in experimental results in table1 ,prove this claim .in order to perform future works, the proposed algorithm in round robin can be developed with Multiple queues , and also Neural methods can be used in order to determine the exact amount of time quantum in multiprocessing environments.

REFERENCES

[1] A. S. Tanenbaum and A. Tannenbaum, *Modern operating systems* vol. 2. Englewood Cliffs: Prentice Hall, 1992.

[2] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating system concepts* vol. 8: Wiley, 2013.

[3] R. J. Matarneh, "Self-adjustment time quantum in round robin algorithm depending on burst time of the now running processes," *American Journal of Applied Sciences,* vol. 6, pp. 1831-1837, 2009.

[4] A. E. A. Agha and S. J. Jassbi, "A new method to improve round robin scheduling algorithm with quantum time based on harmonic-arithmetic mean (HARM)," *International Journal of Information Technology & Computer Science,* vol. 5, pp. 56-62, 2013.

[5] A. Singh, P. Goyal, and S. Batra, "An optimized round robin scheduling algorithm for CPU scheduling," *International Journal on Computer Science & Engineering,* vol. 02, pp. 2383-2385, 2010.

[6] N. Goel and R. B. Garg, "An optimum multilevel dynamic round robin scheduling algorithm," *International Journal of Computer Applications,* vol. 76, pp. 42-46, 2013.

[7] S. K. Panda and S. K. Bhoi, "An effective round robin algorithm using min-max dispersion measure," *International Journal on Computer Science & Engineering,* vol. 4, pp. 45-53, 2012.

[8] S. KumarPanda, D. Dash, and J. Kumar Rout, "A group based time quantum round robin algorithm using min-max spread measure," *International Journal of Computer Applications,* vol. 64, pp. 1-7, 2013.

[9] A. Noon, A. Kalakech, and S. Kadry, "A new round robin based scheduling algorithm for operating systems: Dynamic quantum using the mean average," *International Journal of Computer Science Issues (IJCSI),* vol. 8, pp. 224-229, 2011.

[10] D. Nayak, S. K. Malla, and D. Debadarshini, "Improved round robin scheduling using dynamic time quantum," *International Journal of Computer Applications,* vol. 38, pp. 34-38, 2012.

[11] S. Hiranwal and 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, pp. 319-323, 2011.

[12] C. Yaashuwanth and D. R. Ramesh, "A new scheduling algorithms for real time tasks," *International Journal of Computer Science and Information Security (IJCSIS),* vol. 6, pp. 61-66, 2009.

[13] H. S. Behera, S. Patel, and B. Panda, "A new dynamic round robin and SRTN algorithm with variable original time slice and intelligent time slice for soft real time systems," *International Journal of Computer Applications,* vol. 16, pp. 54-60, 2011.

[14] R. Mohanty, H. S. Behera, K. Patwari, M. Dash, and M. L. Prasanna, "Priority based dynamic round robin (PBDRR) agorithm with intelligent time slice for soft real time systems," *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 2, pp. 46-50, 2011.

[15] A. P. M. Srivastav, S. Pandey, I. Gahoi, and N. K. Namdev, "Fair priority round robin with dynamic time quantum: FPRRDQ," *International Journal of Modern Engineering Research (IJMER),* vol. 2, pp. 876-881, 2012

[16] I. S. Rajput and D. Gupta, "A priority based round robin CPU scheduling algorithm for real time systems," *International Journal of Innovations in Engineering and Technology,* vol. 1, pp. 1-11, 2012.

[17] H. S. Behera, B. K. Swain, A. K. Parida, and G. Sahu, "A new proposed round robin with highest response ratio next (RRHRRN) scheduling algorithm for soft real time systems," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 1, pp. 200-206, 2012.

[18] M. A. H. Al-Hagery, "A selective quantum of time for round robin algorithm to increase CPU utilization," *International Journal of Computer Information Systems,* vol. 3, pp. 54-59, 2011.

[19] C. S. McCahon and E. S. Lee, "Job sequencing with fuzzy processing times," *Computers & Mathematics with Applications,* vol. 19, pp. 31-41, 1990.

[20] M. Hapke and R. Slowinski, "Fuzzy priority heuristics for project scheduling," *Fuzzy Sets and Systems,* vol. 83, pp. 291-299, 1996.

[21] E. O. Oyetunji and A. E. Oluleye, "Performance assessment of some CPU scheduling algorithms," *Research Journal of Information and Technology,* vol. 1, pp. 22-26, 2009.

[22] D. Chillet, A. Eiche, S. Pillement, and O. Sentieys, "Real-time scheduling on heterogeneous system-on-chip architectures using an optimised artificial neural network," *Journal of Systems Architecture,* vol. 57, pp. 340-353, 2011.

[23] P. Surekha and S. Sumathi, "PSO and ACO based approach for solving combinatorial fuzzy job shop scheduling problem," *International Journal of Computer Technology and Applications,* vol. 2, pp. 112-120, 2011.

[24] S. Raheja, R. Dhadich, and S. Rajpal, "AN optimum time quantum using linguistic synthesis for round robin cpu scheduling algorithm," *International Journal on Soft Computing,* vol. 3, pp. 57-66, 2012.

[25] H. S. Behera, R. Mohanty, S. Sahu, and S. K. Bhoi, "Comparative performance analysis of multi dynamic time quantum round robin (MDTQRR) algorithm with arrival time," vol. 2, pp. 262-271, 2011.