

Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method

Shahram Saeidi

Sama Technical and Vocational Training College, Islamic Azad University, Tabriz Branch, Tabriz, Iran
sh_saeidi@iaut.ac.ir

Hakimeh Alemi Baktash

Sama Technical and Vocational Training College, Islamic Azad University, Tabriz Branch, Tabriz, Iran
Alemi_ha@yahoo.com

Abstract— The process scheduling, is one of the most important tasks of the operating system. One of the most common scheduling algorithms used by the most operating systems is the Round Robin method in which, the ready processes waiting in ready queue, seize the processor for a short period of time known as the quantum (or time slice) circularly. In this paper, a non-linear programming mathematical model is developed to determine the optimum value of the time quantum, in order to minimize the average waiting time of the processes. The model is implemented and solved by Lingo 8.0 software on four selected problems from the literature.

Index Terms— Time Quantum, Round Robin, Process Scheduling

I. Introduction

The operating system (OS), as the most important program needed for starting up and using the hardware, has different managing tasks each of which, is performed by one of the management units of the OS. The process management, as one of these management units, allocates the processor to the processes using several allocating algorithms.

One of the most common algorithms in processor allocation, is the Round Robin (RR) algorithm in which, the ready processes waiting in ready queue, are dispatched sequentially and allocate the processor for certain period of time known as time quantum (q) or time slice. If a process is finished during its time quantum, releases the processor, otherwise the processor is pre-empted by the OS and is allocated to the next ready process waiting in front of the ready queue and the current process will be moved to the end of the this queue. The algorithm continues until all the processes are terminated, or the system is switched off. This method is known as best dispatching algorithm in time sharing systems [1].

The most important issue in RR algorithm, which highly affects its efficiency, is determining the amount of the time quantum. Assigning very small and very large values for this parameter, will lead to performance decrease due to increasing context switching overhead, and degradation of the RR method to FCFS algorithm consequently. Actually the value of the quantum parameter is selected in 10-100 milliseconds range [2]. Each value will lead to a specific performance and will affect the algorithm's efficiency by affecting the processes' waiting time.

In this research, a mathematical model is developed to obtain the optimum value for time quantum parameter in order to minimize the average waiting time of the processes. The model is implemented in Lingo software and solved for four selected problems from the literature.

The content of the paper is organized as the follows: The problem definition and the literature review is discussed in section 2. The proposed mathematical model is completely discussed in section 3. Section 4 contains the implementation and computational results of the proposed model over four adopted problems from the literature. Conclusions are summarized in section 5.

II. Literature Review

2.1 Terminology

This research is related to one of the most important issues in principles of the operating system design. In This section, the more frequently used terms in this literature are defined. Some of these terms, are used as the efficiency criteria by some researchers [1], [2].

- Process: The program which is loaded into memory to be executed.
- Ready Queue: A waiting list holding ready processes according to incoming order. These processes are waiting for dispatch.

- Scheduling: Making decision about allocating policies of the resources to the processes.
- Long-term Scheduling: Making decision whether to accept a new job or not.
- Short-term Scheduling: Also known as dispatching, deals with the allocation or releasing the processor to the ready processes.
- Non pre-emptive Scheduling: A scheduling policy in which, the processor will not be pre-empted by the OS after allocation.
- Pre-emptive Scheduling: A scheduling policy in which, the processor can be pre-empted before the process is terminated normally.
- Time Quantum: Also known as time slice, the amount of time that a process captures the processor.
- Process State: The current situation of the a process in the system, including ready, waited(blocked), running, suspended, etc.
- Waiting time: A time period which a process waits to allocate a resource in the system.
- Burst time: A time period which a process needs for completion.
- Context Switching: Switching the processor from the current process to next ready, includes storing and retrieving the values of flags and some important registers.
- Turnaround Time: The time interval between dispatching and completing of a job.
- Response Time: The time taken until the first response of a typical request is emerged.
- Throughput: The number of completed(finished) jobs in a specific time period.

2.2 Round Robin Scheduling

The main goal of the CPU scheduling is distribution of the CPU time among the ready processes in a way that at least one of the system efficiency criteria is achieved. These criteria can be listed as below: [2]

- Acceptable response time.
- Performing the task (process) during the user pre-defined time period.
- Increasing the CPU utilization.
- Increasing the utilization of other system resources.
- Decreasing overall overhead.
- Decreasing the user waiting time.
- Decreasing turnaround time.
- Increasing the system throughput.

Some of the mentioned criteria are related together and will be affected by changing the others. Besides, these changes may not be at the same direction. That is, optimizing one of them, may lead to worst values for another and a unique optimum solution satisfying all these criteria at the same time does not exist necessarily [1,2].

The scheduling algorithms can be divided into two types, pre-emptive and non pre-emptive. The former type methods, allocate the processor to the processes exclusively until completion. The latter type methods, pre-empt the processor from the process at a specific time slice before completion and switch to the next processes.

Round Robin is one of the pre-emptive short-term scheduling policies in which, each of the ready processes waiting in the ready queue, get a time slice known as quantum to allocate the processor. If the amount of this quantum is enough for completion of the process, the process will release all allocated resources after termination, otherwise, the operating system re-takes the processor, and the process moves to the end of the ready queue to get dispatched again. So, a context switching occurs and the next ready process allocates the processor for the same quantum of time.

The most important parameter in this scheduling algorithm, is the amount of the time quantum [1]. If a small value is assigned to this parameter, the short processes will pass the system very soon, and the overhead of the system will increase due to the increasing the number of context switches. On the other hand, if the amount of the time quantum is larger than the maximum burst time of the ready processes, this policy will downgrade to FCFS scheduling algorithm. The value of this parameter at least should be considered as a little more than a specific transaction [2], or the time needed for a common conversation [1].

It's obvious that the amount of this parameter, highly affects the value of waiting time, turnaround time and response time of all the processes, and the number of context switches.

2.3 Previous Works

The importance of the problem, has already raised the attention of the researchers and study in this field still continues. Some more important works are listed below.

Albielmona [3] did an overall review over many scheduling algorithms. Proportional share scheduling algorithm proposed in [4] combines the small overhead of the RR method by protecting the short processes. The capability of re-adjusting the weights, enables the algorithm to have a more fair behaviour.

Nieh et al. [5] proposed a new scheduling algorithm known as Virtual Time Round-Robin (VTRR) with complexity of $O(1)$ by combining fair queuing algorithms with the RR scheduling algorithm.

Zahedi et al. [6] optimized the RR algorithm by using the concept of fuzzy rules and introduced it as Fuzzy rule based Round-Robin CPU Scheduling (FRRCS). They showed that the proposed algorithm, reduces the average waiting time.

The constraint of static time quantum is resolved in [7] by dynamically re-adjusting the value of the parameter repeatedly. Singh et al. [8] proposed a new RR algorithm and showed that their method, reduces the problems of the simple RR and increases the system throughput.

Mostafa et al. [9] introduced a new algorithm based on integer programming method to calculate the appropriate value of the time quantum. Mohanty et al. [10] also developed a new algorithm called Shortest Remaining Burst Round Robin (SRBRR) based on minimum remaining burst time and showed that this method is more effective than common RR. Behera et al. [11] developed a new algorithm called Dynamic Quantum with Re-adjusted Round Robin (DQRRR) and experimentally showed that this method is more efficient than the classic RR due to less number of context switches.

A new RR scheduling algorithm is proposed in [12] in which the time quantum is calculated intelligently and individually for every task. The researchers showed that the throughput of the system is increased due to the decreasing the total number of context switches.

Behera et al. [13] also proposed a method called Multi Dynamic Time Quantum Round Robin (MDTQRR) which dynamically calculated the value of the time quantum and leads to an increase in system throughput.

Hiranwal & Roy [14] proposed a priority driven scheduling algorithm based on the burst time of the processes. They showed that the use of this method increases the performance and stability of the operating system and supports building a self-adaptive OS.

Noon et al. [15] developed a new algorithm called AN, based on a new approach named dynamic time quantum which enables the operating system to adjust the time quantum according the burst time of the set of waiting processes in ready queue. Based on simulation and experiment, they showed that their proposed algorithm solves the static (fixed) time quantum problems and increases the performance of RR.

Alam [16] used fuzzy logic to determine the best quantum value in RR scheduling algorithm and claimed that the throughput of the system is not decreased due to unnecessarily context switches.

Suresh et al. [17] developed an algorithm to increase the quality of service in WiMax networks by proposing a Round Robin based scheduling mechanism which maintains the fairness among the users.

Raheja et al. [18] also developed an algorithm to determine the optimum time quantum value in Round Robin scheduling, based on vague and non-precise parameters using Linguistic variables and Mamdani Fuzzy logic and called it LRRTQ. The authors used Matlab software to implement the proposed algorithm and fuzzy interface system.

Nayak et al. [19] developed a new scheduling algorithm called Improved Round Robin (IRR) by dynamically calculating the time quantum value in order to decrease the average waiting time of the processes and total number of context switches.

Siregar [20] used Genetic Algorithm to determine optimum time quantum value in Round Robin CPU scheduling algorithm to minimize the average waiting time of the processes.

III. The Proposed Model

In this section, a non-linear mathematical model for optimizing the time quantum value in RR scheduling algorithm is proposed.

3.1 Assumptions

In the proposed model, it is assumed that:

- There are n number of ready processes waiting in the ready queue to be dispatched circularly for the time being.
- The burst time of each process is known and available.
- The hardware system is uni-processor.
- The context switching time is equal for all the processes.

3.2 Parameters

The parameters of the model can be listed as follows:

- n : Total number of ready processes at the beginning time.
- S_i : Burst (service) time of the i^{th} process.
- TAT_i : Turnaround time of the i^{th} process.
- W_i : Waiting time of the i^{th} process.
- R_i : Total number of the times which the i^{th} process will seize the processor.
- Lq_i : The amount of the last time quantum which the i^{th} process needs.

- PP_{ij} : Total burst time of the processes like j , that are waiting in the ready queue before the i^{th} process, at the beginning time.
- PS_{ij} : Total burst time of the processes like j , that are waiting in the ready queue after the i^{th} process, at the beginning time.
- CT : Time needed for context switching.

$$\text{Min } \overline{W} = \frac{\sum_{i=1}^n W_i}{n} \quad (1)$$

s.t.

$$TAT_i = (R_i - 1)(q + CT) + Lq_i + \sum_{j < i} PP_{ij} + \sum_{i < j} PS_{ij} \quad (2)$$

$$W_i = TAT_i - S_i \quad (3)$$

$$R_i = \left\lceil \frac{S_i}{q} \right\rceil \quad (4)$$

$$Lq_i = S_i - \left\lceil \frac{S_i}{q} \right\rceil \cdot q \quad (5)$$

$$PP_{ij} = \begin{cases} R_i * (q + CT) & \text{if } R_i < R_j \\ (R_j - 1) * (q + CT) + Lq_j + CT & \text{otherwise} \end{cases} \quad \forall j < i \quad (6)$$

$$PS_{ij} = \begin{cases} (R_i - 1) * (q + CT) & \text{if } R_i \leq R_j \\ (R_j - 1) * (q + CT) + Lq_j + CT & \text{otherwise} \end{cases} \quad \forall j > i \quad (7)$$

$$q : \text{integer} > 0; \quad (8)$$

where:

The main objective of the model, is the minimization of the average waiting time of the processes which is given in (1). The equation (2) calculates the turnaround time of the i^{th} process which equals to number of the times that the process receives a complete quantum from the processor, including the context switching time, plus the amount of the last time quantum, plus total sum of execution times of the predecessor and successor processes of the i^{th} process. The equation (3), calculates the waiting time of the i^{th} process. Equations (4) and (5), calculate the total number of the times which the i^{th} process will capture the processor, and the amount of the last needed time quantum respectively. The equations (6) and (7) calculate total execution time of the predecessor and successor

3.3 Decision Variables

The proposed model includes only one decision variable:

- q : The amount of time quantum.

3.4 The Mathematical Model

The model is proposed to minimize average waiting time of the processes as follows:

processes respectively. The constraint of having an integer value for the decision variable q , is added in (8).

IV. Implementation and Computational Results

The proposed model is implemented in Lingo 8.0 optimization software and solved over four selected problems from the literature. The problems' details and the computational results are discussed in the following subsections.

4.1 Example 1

The first example is adopted from Mohanty et al. [10]. The details are listed in Table 1.

Table 1: Process specification of example 1

Process#	Arrival Time	Burst Time
1	0	13
2	0	35
3	0	46
4	0	63
5	0	97

The researchers in this paper, have proposed a method called SRBRR to calculate the value of the time quantum, and have compared the results with the classic RR algorithm with static time quantum equal to 25. The compared results in the original paper, along

with the results obtained by the proposed algorithm, are listed in Table 2. The results show that the proposed model has obtained better values. The value of the context switching is assumed to be zero.

Table 2: Computational results of example 1

Used Method	Time Quantum (q)	Average Turnaround Time	Average Waiting Time
Static RR	25	148.2	97.4
SRBRR	17, 34, 46	122.4	71.6
Proposed Model	63	113.2	62.4

4.2 Example 2

The second example is adopted from Hiranval and Roy [14] and the details are given in Table 3. The

amount of context switching in the original paper is assumed to be zero.

Table 3: Process specification of example 2

Process#	Arrival Time	Burst Time
1	0	14
2	0	34
3	0	45
4	0	62
5	0	77

The researchers of the reference, have proposed a new algorithm called Adaptive Round Robin (ARR) and have compared the obtained results with the classic RR method. This comparison along with the obtained

values by the proposed method in this paper is given in Table 4. The results in this example show the superiority of the proposed model.

Table 4: Computational results of example 2

Used Method	Time Quantum (q)	Average Turnaround Time	Average Waiting Time
Classic RR	25	143.4	97
ARR	45	117.4	71
Proposed Model	62	108.2	62

4.3 Example 3

The third example is also adopted from the research by Hiranval and Roy [14] as specified in Table 5. The

processes are sorted decreasingly based on the burst time and context switching time is assumed to be zero.

Table 5: Process specification of example 3

Process#	Arrival Time	Burst Time
1	0	83
2	0	54
3	0	30
4	0	19
5	0	8

The comparison of the obtained results in the original paper besides the results given by the proposed algorithm is shown in Table 6. As the values show, the

proposed method has a better performance than previous methods over this example.

Table 6: Computational results of example 3

Used Method	Time Quantum(q)	Average Turnaround Time	Average Waiting Time
Classic RR	26	149.2	110.4
ARR	45	157	118.2
Proposed Model	10	128	84.6

4.4 Example 4

The fourth example is adopted from Noon [15] paper. As mentioned in section 2, the researchers in this paper have proposed a new method called AN to calculate

the value of the time quantum in RR method of process scheduling and have compared it with dynamic RR algorithm. The problem specification and the computational results are given in Table 7 and Table 8.

Table 7: Process specification of example 4

Process#	Arrival Time	Burst Time
1	0	20
2	0	40
3	0	60
4	0	80

As the results show in Table 8, the proposed method has obtained the same values obtained by AN method, but better than that of dynamic RR.

Table 8: Computational results of example 4

Used Method	Time Quantum (q)	Average Turnaround Time	Average Waiting Time
AN Method	60	100	50
Dynamic RR	20	120	70
Proposed Model	60	100	50

V. Conclusions

In this paper, a new non-linear mathematical model is developed to calculate the optimum value of the time quantum in Round Robin process scheduling algorithm to minimize the average waiting time of the processes. The model is implemented in Lingo 8.0 software and four selected problems adopted from the literature are implemented and solved. In all the selected examples, the proposed method obtained utterly better or equal performance than the previous algorithms in the references.

Acknowledgement

The authors would like to thank Sama Technical and Vocational Training College, Tabriz Branch, Islamic Azad University for the financial support of this research, which is based on a research project contract.

References

- [1] W. Stallings, *Operating Systems: Internals and Design Principles*. 6th Edition, Prentice Hall, 2008.
- [2] M. Fahimi, *Operating Systems*, vol. 1, 1st Edition, Tehran: Jelve Publishing, 1992.
- [3] R. Abielmona, "Scheduling Algorithmic Research", Department of Electrical and Computer Engineering, Ottawa-Carleton Institute, 2000.
- [4] T. Helmy, and A. Dekdouk, "Burst Round Robin: As a Proportional-Share Scheduling Algorithm", *IEEE, Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations*, pp. 424-428, 2007.
- [5] J. Nieh, Ch. Vaill, and H. Zhong, "Virtual-Time Round-Robin: An O(1) Proportional Share Scheduler." *Proceeding of the 2001 USENIX Annual Technical Conference*, USA, 2001.

- [6] M.H. Zahedi, M. Ghazizadeh, and M. Naghibzadeh, "Fuzzy Round Robin CPU Scheduling (FRRCS) Algorithm", *Advances in Computer and Information Sciences and Engineering*. 348-353, 2008.
- [7] 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*, 6(10), 1831- 1837, 2009.
- [8] A. Singh, P. Goyal, and S. Batra, "An Optimized Round Robin Scheduling Algorithm for CPU Scheduling", *International Journal on Computer Science and Engineering*. 2(7), 2382-2385, 2010.
- [9] S.M. Mostafa, S.Z. Rida, and S.H. Hamad, "Finding time quantum of Round Robin CPU scheduling in general computing systems using integer programming", *International Journal of Research and Review in Applied Science*. 5(1), 64-71. 2010.
- [10] R. Mohanty, H.S. Behera, K. Patwari, M.R. Das, M. Dash, and Sudhasree, "Design and Performance Analysis of a new proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm", *Proceedings of the International Symposium on Computer Engineering and Technology (ISCET)*, 2010.
- [11] H.S. Behera, R. Mohanty, and D. Nayak, "A new proposed dynamic quantum with re-adjusted Round Robin scheduling algorithm and its performance", *International Journal of Computer Applications*, 5(5), 10-15, 2010.
- [12] C. Yaashuwanth, and R. Ramesh, "Intelligent Time Slice for Round Robin in Real Time Operating Systems", *IJRRAS*, 2(2), 126-131, 2010.
- [13] 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", *Indian Journal of Science and Engineering*, 2(2), 262-271, 2011.
- [14] 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*, 2(2), 219-326, 2011.
- [15] A. Noon, A. Kalakech, and S. Kadry, "A new Round Robin Based scheduling algorithm for Operating Systems: Dynamic Quantum time Mean Average", *International Journal of Computer Science Issues*. 8(3), 224-229, 2011.
- [16] B. Alam, "Finding Time Quantum of Round Robin CPU Scheduling Algorithm Using Fuzzy Logic", International conference on Computer and Electrical Engineering, ICCEE 2008.
- [17] B. Suresh, P.V.G.D. Prasad Reddy, and C.K. Chakravarthy, "Variable Quantum Deficit Round Robin Scheduling for Improved Fairness in MULTIHOP Networks", *International Journal of Distributed and Parallel Systems*, Vol. 2., No. 1, 2011.
- [18] 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. No. 1, 2012.
- [19] D. Nayak, S. K. Malla, and D. Debadarshini, "Improved Round Robin Scheduling using Dynamic Time Quantum", *International Journal of Computer Applications*, Vol. 38, No. 5, 2012.
- [20] M. U. Siregar, "A New Approach to CPU Scheduling Algorithm: Genetic Round Robin", *International Journal of Computer Applications*, Vol. 47, No. 19., 2012.

Authors' Profile:



Shahram Saeidi, Born in Tabriz, Iran, 1973. Received Industrial Engineering (Ph.D. and M.Sc.), Computer Software Engineering (B.Sc.). Professor of Computer Engineering Department, Islamic Azad University, Tabriz Branch, Tabriz, Iran.

Hakimeh Alemi Baktash, Born in Ghom, Iran, 1978. Received Computer Software Engineering (M.Sc. And B.Sc.). Teacher Assistant, Sama Technical and Vocational Training College, Islamic Azad University, Tabriz Branch, Tabriz, Iran

How to cite this paper: Shahram Saeidi, Hakimeh Alemi Baktash, "Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method", *International Journal of Information Technology and Computer Science (IJITCS)*, vol.4, no.10, pp.67-73, 2012. DOI: 10.5815/ijitcs.2012.10.08