New developments in Round Robin algorithms and their applications: a systematic mapping study

Daniela L. Freire, Rafael Z. Frantz, Fabricia Roos-Frantz

Department of Exact Sciences and Engineering, Unijuí University
Ijuí-RS, Brazil E-mail: {dsellaro, rzfrantz, frfrantz}@unijui.edu.br

Vitor Basto-Fernandes

Instituto Universitário de Lisboa (ISCTE-IUL) ISTAR-IUL
Lisbon, Portugal E-mail: vitor.basto.fernandes@iscte-iul.pt

Abstract:
Enterprises have a set of applications support that need to work synchronously and efficiently to respond quickly to business processes. To achieve high performance in the execution of the applications, software engineers suitably must task scheduling on the computational resources. The Round Robin algorithm is an efficient and effective scheduling technique for task scheduling in computing. This article provides a systematic mapping study that identifies the state-of-the-art in the research of the Round Robin algorithms to guide researchers and practitioners in the field of software engineering. The research regarding the improvement in the Round Robin algorithm continues active, indicating that it remains one of the more efficient scheduling techniques in the fields of packets, CPU and virtual machine scheduling.

Keywords: Round Robin Algorithm; Scheduling; Systematic Mapping Study; Performance Evaluation Metric; Simulation Tool.

Biographical notes: Daniela L. Freire received her PhD in Mathematical Modelling at Unijuí University. Her current research interest involved the use of artificial intelligence to solve problems in heath.
Rafael Z. Frantz is an Associate Professor who is with the Department of Exact Sciences and Engineering of Unijuí University and leads the Applied Computing Research Group since 2013. He was awarded a PhD degree in Software Engineering by the University of Seville, Spain. His current research interests focus on integration of enterprise applications, optimization, decentralised technologies including blockchain, smart contracts, and data integration.
Fabricia Roos-Frantz is an Associate Professor who is with the Department of Exact Sciences and Engineering of Unijuí University. She received her PhD in Software Engineering from the University of Seville, Spain. Her current research interests include data integration and analysis, optimization, petri nets, and learning analytics.
Vitor Basto-Fernandes graduated in information systems in 1995, pos-graduated in distributed systems in 1997 and got his PhD on multimedia transport protocols in 2006 from the University of Minho - Portugal, where he has been teaching assistant. From 2005 he has also been lecturing at the University of Tras-os-Montes e Alto Douro - Portugal, and invited assistant professor in the same university in 2007-2008. In 2008 he joined the Polytechnic Institute of Leiria (Portugal) in the Informatics Engineering Department, and in 2016 he moved to the University Institute of Lisbon (Portugal) where he is currently assistant professor at the Department of Information Science and Technology.

1 Introduction

To support their business processes, enterprises usually have a set of applications that compose the software ecosystem (Manikas, 2016). Advanced software engineering approaches support the composition of applications using computing resources with Infrastructure-as-a-Service, Service Level Agreements, deployment and run-time orchestration in multi-cloud environments (Kochovski et al., 2019). This composition of applications has left software ecosystems even more heterogeneous and dynamic. The applications and services need to be prepared to work together to provide fast and reliable answers for business processes of the company (Frantz et al., 2016). A heterogeneous software ecosystem demands compile-time and run-time aid for executing applications. The efficient task scheduling execution on the available computational resources is one of the main requirements for achieving high performance (Topcuoglu et al., 2002; Alsheikhy et al., 2015). Therefore, this problem has been widely studied and various algorithms have been proposed in the literature (Lakshmanan, 2017; Ghosh, 2017; Fataniya and Patel, 2018a,b). Some of the most popular scheduling algorithms are Shortest Job First, First Come First Served, Priority Scheduling, and Round Robin, being Round Robin the preferable choice amongst the scientific and technical community (Upadhyay and Hasija, 2016; Yasin et al., 2015). Round Robin algorithm gained popularity because it is a simple and straightforward algorithm, easy to implement and to understand from the users' perspective (Bryhni et al., 2000; Zhang et al., 2018). Besides, this algorithm has had much notoriety due to its optimal time-shared environment (Mohammed et al., 2016; Tajwar et al., 2017) and its ability to avoid starvation and to allow fair resource allocation to every task (Datta, 2015; Sabha, 2018) and it is regarded as one of the most suitable algorithm for multi-tasking real-time systems (Wason et al., 2016). The correctness of real-time scheduling is determined by both logical accuracy and timely execution (Baek and Lee, 2019).

There is a plethora of proposals of variants of the original Round Robin algorithm, which have been published in journals, conferences, and workshops, becoming hard to get a clear and complete overview of the state-of-the-art in the research field. Our literary review did not found a systematic mapping study on Round Robin algorithms in software engineering, instead, we only identified works that seek to evaluate and compare the performance of Round Robin-based algorithms (Negi and Kalra, 2014; Aslam and Shah, 2015; Musa et al., 2018). Systematic mapping is a method in software engineering research to identify gaps and clusters in the literature by classifying existing research (Petersen et al., 2015). It is seen as a scrupulous and unbiased method, in which the search can be easily repeated and, thereby, the results can be verified (Kitchenham and Stuart, 2007).

In this article, we conducted a systematic mapping study to overcome this gap in knowledge, identifying the state-of-the-art in Round Robin algorithms in software engineering. We follow the research protocol proposed by Kitchenham and Stuart (2007) and Petersen et al. (2015), in order to provide a valuable baseline to assist new research efforts (Kitchenham et al., 2010). We identify the main authors and venues of publications of Round Robin-based algorithm's proposals. Besides, we identified main variants of the Round Robin algorithm, their performance evaluation metrics and simulation tools. Regarding performance evaluation metrics, there is standardisation in the field of CPU scheduling, but this is not true in the field of packets scheduling and virtual machine scheduling. Regarding simulation tools, there is a gap of consolidated tools in the CPU scheduling field.

The rest of this article is organised as follows: Section 2 discusses the related work regarding evaluation and comparison of Round Robin algorithms; Section 3 provides background information on Round Robin algorithm; Section 4 introduces the research methodology that we used to conduct this systematic mapping study; Section 5 aggregates the results by means of a compact overview in graphics and tables; Section 6 discusses threats to validity; and, Section 7 presents our conclusions.

2 Related Work

Effective resource management and task scheduling mechanism, required in order to provide task completion time minimisation, are still open issues (Zeng et al., 2016). In this section, we discuss those proposals that examined or compared algorithms, methodologies, and techniques that address theses issues. The related works are summarised in Table 1, contemplating the research field, the approach type, and if the approach is Round Robin (RR) based or not.

Hu et al. (2010) presented a scheduling strategy on load balancing of virtual machines resources based on genetic algorithm and compared with the least-loaded scheduling and the rotating scheduling, regarding load balancing and reasonable resources utilisation. Ákos Szőke (2011) proposed a conceptual model for agile release scheduling by an optimisation algorithm. Release scheduling deals with the selection and assignment of deliverable features to a sequence of consecutive product deliveries while several constraints are fulfilled. Samal and Mishra (2013) evaluated the Round Robin algorithm and two of its variants: Modified Round Robin and Time Slice Priority Based Round Robin. They performed an experiment to compare the response time, average waiting time, and average turnaround time in the three algorithms. Negi and Kalra (2014) provided a comparative study of Round Robin algorithms to evaluate the performance of the variants: Self-Adjustment Time Quantum in Round Robin, Adaptive Round Robin, Ascending Quantum and Minimum-Maximum Round Robin, Mid Average Round Robin. The performance metrics were the average waiting time, the average turnaround time, and the number of context switches.

Aslam and Shah (2015) provided a literature review on load balancing algorithms in cloud computing, evaluating the performance of proposals that have been developed over the period of 2004-2015. Their work compared load balancing algorithms for cloud computing such as Round Robin, Min-Min, Max-Min, Ant colony, Carton, and Honey Bee. Maniyar and Kanani (2015) analysed seven variants of Round Robin algorithm regarding computational tasks scheduling in the cloud environment. They described the variants and appointed some benefits and challenges of Round Robin algorithms. Dash et al. (2015b) proposed a

Table 1 Summary of the related work.

Ref.	Research field	Approach	RR
Hu et al. (2010)	Cloud Computing	Performance evaluation	✗
Ákos Szőke (2011)	Management Project	Performance evaluation	✗
Samal and Mishra (2013)	CPU	Performance evaluation	✓
Negi and Kalra (2014)	CPU	Performance evaluation	✓
Aslam and Shah (2015)	Cloud Computing	Literature review	✓
Maniyar and Kanani (2015)	Cloud Computing	Literature review	✓
Dash et al. (2015b)	CPU	Performance evaluation	✓
Musa et al. (2018)	CPU	Performance evaluation	✓
Mahdavi-Hezavehi et al. (2017)	Self-adaptive System	Literature review	✗
Hanafi and RacaTodosijević (2017)	Heuristics	Literature review	✗
Magdich et al. (2018)	Real-time Embedded System	Literature review	✗
Blot et al. (2018)	Heuristics	Literature review	✗
Feld et al. (2018)	Real-time Embedded System	Literature review	✗
Corstjens et al. (2018)	Heuristics	Performance evaluation	✓
Srivastava and Kumar (2019)	CPU	Performance evaluation	✓
Kumar et al. (2019)	Cloud Computing	Literature review	✓
Qin et al. (2019)	Stream Processing Systems	Literature review	✗
Guerrero et al. (2019)	Fog Computing	Literature review	✗
Rezende et al. (2019)	Management Project	Literature review	✗
Campelo and Takahashi (2019)	Heuristics	Performance evaluation	✗
[Our Proposal]	EAI	Mapping systematic	✓

variant called Dynamic Average Burst Round Robin to increase the performance of CPU scheduling. They compared with other variants: Improved Round Robin with Varying Time Quantum, Mode Round Robin, Self-Adjustment Time Quantum in Round Robin, R.P-5, and Dynamic Quantum with Readjusted Round Robin. They performed an experiment with three scenarios: ascending, descending, and random burst time.

Musa et al. (2018) randomly selected three variants of Round Robin algorithms: Dynamic Average Burst Round Robin, Revamped Mean Round Robin, and Dynamic Time Slice Round Robin. They performed an experiment to compare the average waiting time, average turnaround time, and number of context switch in the three algorithms, besides the original Round Robin algorithm. Mahdavi-Hezavehi et al. (2017) reviewed the literature regarding architecture-based methods for handling multiple quality attributes (QAs) in self-adaptive systems. One of the QAs analysed was the performance regards to the efficiency of the software by using the appropriate amounts and types of resources under stated conditions and in a specific context of use. One of the QAs prioritisation mechanisms identified in the primary studies was polling scheduler. Hanafi and RacaTodosijević (2017) presented a survey of heuristics used for modelling combinatorial problems, such as logical design to scheduling and routing, and graph theory models for resource allocation and financial planning. Their goal was to identify the adequate use of mathematical programming techniques for approximate problem-solving.

Magdich et al. (2018) offered an overview and a comparison between some scheduling algorithms, namely: Proportionate Fair, Earliest Deadline until Zero Laxity, Least Local Remaining Execution First, Global Fair Lateness, and Global Earliest Deadline First. They also proposed a model-based approach which focuses on the use of the Model Driven Engineering and design patterns to support the automatic choice of scheduling approach and algorithm at a high-level of abstraction for real-time embedded systems. Blot et al. (2018) presented a survey regarding the use of local search techniques in multi-objective metaheuristics. They also proposed a standardised structure to represent and consolidate the subjacent components of multi-objective local search techniques and algorithms. Feld et al. (2018) presented a survey on schedulability analysis techniques for tasks with this rate-dependent behaviour. They reviewed methods for both fixed priority and earliest deadline first scheduling and, then, provided a taxonomy of the different analysis methods. Corstjens et al. (2018) studied methodologies that allow analysing the behaviour of heuristic algorithms through the elements operating within the algorithm correlate with performance. Their goal was to find benefits of combinations of these elements and to identify how the specific problem instance the algorithm is resolving impacts the combinations.

Srivastava and Kumar (2019) analysed six existing variants of traditional Round Robin algorithm: Optimised Round Robin algorithm, non-linear programming mathematical model, Mean Difference Round Robin, dynamic average burst Round Robin, self adjustment Round Robin, and Smart Optimised Round Robin. They compared their performance in different scenarios, by metrics: average of turnaround time, waiting time, and number of context switches. Kumar et al. (2019) reviewed the cloud computing literature regarding load balancing in order to achieve resource utilisation. Hence, then reduce energy consumption and carbon emission rate. They discussed static algorithms, such as Round Robin, Min-Max, and Min-Max, and dynamic algorithms, such as Honeybee Foraging Behaviour, Throttled, Equally Spread Current Execution, Ant Colony, Biased Random Sampling, and Modified Throttled. Qin et al. (2019) performed a systematic literature review in order to identify and characterise different approaches regarding the enactment of run-time adaptation in stream processing. One of the inclusion criteria was articles that discuss scheduling approaches that cause a change of the processing in the execution phase. As resource-related enactment approaches, they identified resource scheduling and processing scheduling.

Guerrero et al. (2019) presented a brief survey of optimisation proposals for the fog service placement problem. The proposals explored optimisation techniques, such as heuristics, greedy algorithms, linear programming, or genetic algorithms; and defined features of the fog resources, such as placement, scheduling, allocation, provisioning, or mapping for services, resources, clients, tasks, virtual machines, or even fog colonies. Rezende et al. (2019) presented a systematic review on software project scheduling problem, concerning the context of search-based software engineering. The authors classified the articles selected by main elements of the software project scheduling such as the model used, search-based techniques, instances used to validate the solution and evaluation aspects. Campelo and Takahashi (2019) presented a methodology for the definition of the required sample sizes for experiments concerning the performance of optimisation algorithms to achieve desired statistical properties. The methodology also executes the algorithms on each tested problem instance to obtain the precision of the estimated differences in performance is controlled under a predefined level.

3 Background

Round Robin is one of the oldest, simplest, and most widely used scheduling algorithms, mainly for time-sharing systems (Gope et al., 2011). It uses a proper quantum time to achieve fairness and starvation free towards the tasks (Agha and Jassbi, 2013). A quantum time is a small unit of time and a task is a piece of executable code. The tasks are kept in a circular ready queue and new tasks are always added to the tail of ready queue. A scheduler goes around this ready queue, allocating the available computational resources to each task for a time interval of one quantum. The scheduler selects the first task from the ready queue, which will be executed during one quantum and will have its execution interrupted when this time expires. If one quantum is not enough to completely execute the task (burst time), the task is added to the tail of the queue and the scheduler stores the context of this task in a stack or registers and follows the execution of the next task in the ready queue. This action performed by the scheduler is called context switch. If the execution of the task finishes before the end of the quantum, the task itself releases the computational resource voluntarily. In either case, the scheduler assigns the computational resource to the next task in the ready queue. The pseudo code of the Round Robin is shown in Algorithm 1.

Algorithm 1 *Round Robin Algorithm*

1: Present every task in to ready queue
2: Check if there are tasks in the ready queue
3: Compute $T_{quantum}$ **time quantum**
4: Assign $T_{quantum}$ to tasks $t[j] \leftarrow T_{quantum}$, $j++$
5: **if** $j <$ number of tasks **then**
6:     go to jump to step-line 4
7:     Compute the remaining burst time of the every tasks and jump to step-line 3
8:     **if** a new task has arrived **then**
9:         Update ready queue and jump to step-line 3
10:     **end if**
11: **end if**

The task execution time (T_{Exec}) is incremented by the context switch overhead. Ideally, task execution time is defined by Equation 1, where N is the number of tasks in the ready queue, $T_{quantum}$ is the quantum time, and $T_{context\_switch}$ is the context switch overhead. Thus, each task would not wait for longer than $N \cdot T_{quantum}$ time units until its next quantum.

$$T_{Exec} = N \cdot (T_{quantum} + T_{context\_switch}) \tag{1}$$

Several metrics are used to evaluate the performance of scheduling algorithms, such as computational resource utilisation, throughput, response time, turnaround time, waiting time, correctness, overhead, predictability, fairness, and fault tolerance. These metrics are described as following.

Computational Resource Utilisation: measures the percentage of time that the computational resources are kept busy. For best scheduling, it is desirable that the computational resource utilisation should be about 100 per cent.

Throughput: measures the number of tasks that are completed per time unit. For best scheduling, it is desirable that the throughput should be high.

Response Time: measures the time from the submission of a task until the first response is received from the algorithm. For best scheduling, it is desirable that the response time should be low.

Turnaround Time: measures the time interval from the time of task submission to the task completion time. This metric includes the waiting time spent by a task to get into memory, the waiting time in the ready queue, its time executing in a computational resource, and its time doing I/O. For best scheduling, it is desirable that the turnaround time should be low.

Waiting Time: measures the time spent waiting in a ready queue to submit a task for the computational resource. It does not include the execution time nor the time that the task is doing I/O completion. For best scheduling, it is desirable that the waiting time should be low.

Correctness: measures the proportion of time or proportion of total scheduling time unit when a task fails to meet its deadline. For best scheduling, it is desirable that the correctness should be high.

Overhead: measures the percentage of wasted time by the algorithm, while new tasks are arriving. Such as time spent scheduling planning, context switching, and re-configuring. For best scheduling, it is desirable that the overhead should be low.

Predictability: measures the percentage of time spent by task execution above predicted time to execute this task, irrespective of stochastic factors such as system load variations a or arrival rate. For best scheduling, it is desirable that the predictability should be low.

Fairness: measures the uniformity in task wait times to be executed. In the absence of predefined criteria for selection, the scheduler should allocate a fair amount of the resource to each task in order to ensure homogeneity of available resources for all tasks. For best scheduling,

Fault Tolerance: measures the uniformity in the number of tasks assigned for every resource. In the absence of predefined criteria for selection, the scheduler should avoid overload of a computational resource, considering that this overload can cause execution faults. For best scheduling, it is desirable that the fault tolerance should be high.

Usually, Round Robin has high throughput, good response time and it is less complex than other algorithms. We list the main strength of Round Robin algorithm as follows:

- Easy to understand.
- Fairness.
- Performs better for short time burst.
- Priority (can use running time and arrival time).

The quantum time heavily impacts in the performance of Round Robin algorithm (Aslam and Shah, 2015). When quantum time is very large causes less context switch and lesser turnaround time but it can increase response time and waiting time. In this case, the Round Robin algorithm plays like the no preemptive First-Come, First-Served scheduling, which selects and executes a task until completion according to First-In-First-Out policy. On the other hand, when the quantum time is extremely too small it causes less waiting time and response time but can increase the turnaround time and the context switch number (Mohammed et al., 2016). A high context switch number leads to the wastage of time, of memory and scheduler overhead (Ramakrishna and Rao, 2013). In this case, the Round Robin algorithm is called Processor Sharing Algorithm that is a scheduling policy where the tasks are all executed simultaneously, each receiving an equal fraction of the computational resource available (Hiranwal and Roy, 2011; Fataniya and Patel, 2018a). These weaknesses have made it unsuitable for real-time systems, motivating researchers to propose new variants of the original Round Robin algorithm. We list the main weaknesses of Round Robin algorithm as follows:

- Large waiting time.
- Large response time.
- Increased context switches.
- Large turnaround time.
- Less throughput.

4 Research Method

In this section, we detailed the research method used in our mapping study. We followed the guidelines proposed by Kitchenham and Stuart (2007), Petersen et al. (2015), and Brings et al. (2018). The next section, we present the study design by means of its objective, research questions, and search protocol.

4.1 Research questions

This study has been conducted by the following research question:

> *Which are the state-of-the-art of Round Robin algorithms in software engineering?*

To help answering this research question, we split it into seven specific questions. Such responses allow to classify and tabulate the articles for further analysis. These specific questions are listed as follows.

RQ1: When and where were published the Round Robin algorithms proposals?

RQ2: Which research fields where the Round Robin algorithms are applied?

RQ4: Which are the metrics used to evaluate the Round Robin algorithms?

RQ5: Which are the tools used to evaluate the Round Robin algorithms?

RQ6: What are the most recent variants of the Round Robin algorithm?

RQ7: Which are the most cited Round Robin algorithms proposals?

4.2 Search protocol

In this section, we present the search protocol, which has first been developed by the first author, and later reviewed by the second and third author. The scope of this study included the search terms indicated by Kitchenham and Stuart (2007): Population, Intervention, Comparison and Outcomes (PICO). These terms were used to identify keywords and formulate search strings from research questions.

Population: proposals of Round Robin algorithms applied to task scheduling.

Intervention: strategies, methods, technical or patterns adopted by Round Robin algorithms.

Comparison: different proposals of Round Robin algorithms.

Outcomes: applicability, features, and performance of the Round Robin algorithms.

We performed a search using the following base search string: (“round robin algorithm ” or “rr algorithm”). This study searched for published articles from 2014 to 2018, written in English, in the subject area of Computer Science. The search process for this study is based on an automated search of the indexing systems and digital libraries. The search strings are shown in Table 2 and the correspondence between search string and database used is shown in Table 3.

Table 2 Search strings of the initial collect of articles.

ID	Search string
str_1	*“Round Robin algorithm” “Round Robin policy” “RR algorithm” “RR policy”*
str_2	*TI ((“Round Robin algorithm” OR “Round Robin policy” OR “RR algorithm” OR “RR policy”)) OR AB ((“Round Robin algorithm” OR “Round Robin policy” OR “RR algorithm” OR “RR policy”)) OR KW ((“Round Robin algorithm” OR “Round Robin policy” OR “RR algorithm” OR “RR policy”)))*
str_3	*“Round Robin algorithm” OR “Round Robin policy” OR “RR algorithm” OR “RR policy”*
str_4	*TITLE-ABS-KEY (“Round Robin algorithm”) OR TITLE-ABS-KEY (“Round Robin policy”) OR TITLE-ABS-KEY (“RR algorithm”) OR TITLE-ABS-KEY (“RR policy”)*

We used a preliminary search string that retrieved an initial list of articles for each database considered. After, we used an online tool, called Parsifal [12], to import the references of the collected articles and to remove eventual duplication. In total, 590 articles were returned, being 189 duplicated, remaining 401 unique articles. The number of articles in every database is shown in Table 4.

Table 3 Search strings used by databases.

ID	Database
str_1	ACM [1]
str_2	Ebsco [2]
str_3	Emerald [3], IEEE [4], Wiley [5], Google Scholar [6], Web of Science [7] ProQuest [8], Scielo [9], ScienceDirect [10]
str_4	Scopus [11]

Table 4 Result of the initial collect of articles.

Database	N° of collected
ACM	20
Ebsco	12
Emerald	1
Google Scholar	44
IEEE	93
ProQuest	23
Scielo	3
ScienceDirect	160
Scopus	136
Web of Science	94
Wiley	4

Because of the large volume of returned articles, we decided to use two filters. The first filter intended to separate those in which the Robin Robin algorithm was one of the main subjects addressed in the article. In this filter, the first author read the title, the keywords and the abstract of articles, and applied inclusion and exclusion criteria. The publication year criterion, English language, and subject area of Computer Science were applied again because some database does not allow these constraints in their searches. These criteria are shown as follows.

- Inclusion criteria
  - Articles were published in the last five years (2014-2018).
  - Ensure that the articles evaluate Round Robin algorithms.
- Exclusion criteria
  - Articles are books or belongs to gray literature.
  - Articles not presented in English.
  - Articles not accessible in full-text.
  - Articles do not in the field of computer science.
  - Articles do not approach the Round Robin algorithm.
  - Articles propose algorithm not based in Round Robin algorithm.
  - Articles that only to compare Round Robin algorithms, but did not present Round Robin algorithm proposals.

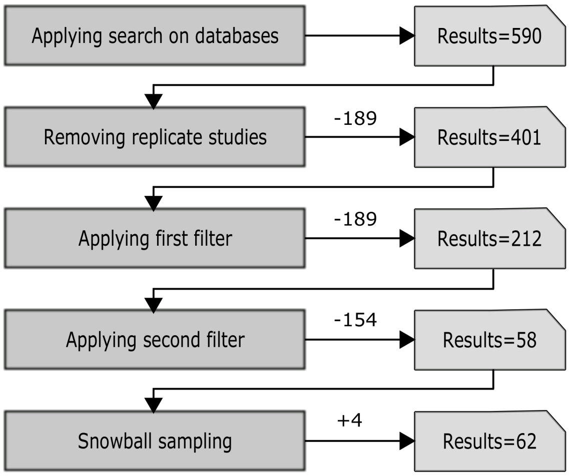

Figure 1 Articles selection process.

The articles that met at least one of the inclusion criteria were included. The articles that met at least one of the exclusion criteria were excluded. There were 212 remained articles in the first filter which were given as input to the second filter. The second filter intend to separate those in which there is a Round Robin algorithm variation proposal. For this filter, the first author read the title, keywords, abstract, introduction and conclusion of articles and applied the following inclusion and exclusion criteria:

- Inclusion criteria
  - Articles propose a Round Robin algorithm variation.
- Exclusion criteria
  - Articles present the same Round Robin algorithm as previous articles.
  - Articles simply apply one of the Round Robin algorithms.
  - Articles simply compare Round Robin algorithms.

The outcome of the second filter is 58 primary articles and there were used to conduct backward snowball sampling, which led to the addition of 4 articles. The backward snowball technique was recursively applied, allowing the retrieval of 4 articles, which were cited by the authors of the collected articles and that are relevant for our research. The steps of the articles selection process are shown in Fig. 1.

The quality assessment for the set of 62 primary articles was based on full-text reading of these articles by the first author. To assess the quality of these articles, we formulated six questions:

- Does the article clearly define its goal?
- Does the article present the research field where the algorithm is applied?
- Does the article present evaluation metrics for the algorithm?
- Does the article present a simulation or execution of the algorithm?
- Does the article present the pseudo-code or code of the algorithm?
- Does the article present the tools used to evaluate the algorithm?

The questions were answered with the following values: `yes`, `partial`, and `no`. An answer `yes` weights 1, `partial` weights 0.5, and `no` weights 0. Therefore, every article can achieve from 0 to 6 points in maximal for the total of six questions. Additionally, we registered what is the research field, metrics, and tools from those articles that provide this information. At the end of this third step, all six question were answered with their respective weight for every 62 articles. Finally, the second and third author reviewed each activity of the analysis to ensure consistency in the analysis and consolidation of the results. The final list of the mapped articles is presented in Table 5, Table 6, and Table 7 .

5 Results

This section presents the results from the analysis of the primary articles, based on the research questions previously mentioned. We presented the main results: publication venues, publication years, authors, research institutions, research fields, tools and metrics used in the articles.

Table 5 Mapped articles.

Ref.	Title	Score
Hamouda et al. (2017)	A downlink resources allocation scheme for multimedia applications in OFDMA wireless systems	6.0
Xu and Wang (2014)	A modified round-robin load-balancing algorithm for cluster-based web servers	4.5
Abdulrahim et al. (2014a)	A new improved round robin (NIRR) CPU scheduling algorithm	6.0
Sabha (2018)	A novel and efficient Round Robin algorithm with intelligent time slice and shortest remaining time first	5.0
Mittal et al. (2015)	A paper on modified round robin algorithm	6.0
Phorncharoen and Sa-Ngiamvibool (2018)	A proposed round robin scheduling algorithm for enhancing performance of CPU utilization	6.0
Kathuria et al. (2016)	A revamped mean round robin (rmrr) cpu scheduling algorithm	6.0
Verma et al. (2014)	A round robin algorithm using mode dispersion for effective measure	4.5
Abdulrahim et al. (2014b)	An additional improvement in round robin (AAIRR) CPU scheduling algorithm	4.5
Banerjee et al. (2017c)	An approach towards development of an intelligent cloudlet scheduling mechanism for Cloud QoS improvement.	6.0
Srinivasu et al. (2015)	An augmented dynamic round robin cpu scheduling algorithm	4.5
Farooq et al. (2017)	An efficient dynamic round robin algorithm for CPU scheduling	5.0
Rao et al. (2014)	An enhanced dynamic round robin CPU scheduling algorithm	2.5
Khatri (2016)	An enhanced round robin CPU scheduling algorithm	5.0
Basker et al. (2014)	An enhanced scheduling in weighted round robin for the cloud infrastructure services	5.0
Kumar et al. (2014)	An improved approach to minimize context switching in round robin scheduling algorithm using optimization techniques	4.0
Alsheikhy et al. (2015)	An improved dynamic round robin scheduling algorithm based on a variant quantum time	5.0
Mishra and Rashid (2014)	An improved round robin CPU scheduling algorithm with varying time quantum	4.5
Xu et al. (2014)	An improving algorithm for combined input-crosspoint-queued switches	4.0
Dash et al. (2015a)	An optimized round robin CPU scheduling algorithm with dynamic time quantum	5.0
Hafeez and Rasheed (2016)	An optimum dynamic time slicing scheduling algorithm using round robin approach	5.0
Khokhar and Kaushik (2017)	Best time quantum round robin CPU scheduling algorithm	5.0
Dash et al. (2015b)	Characteristic specific prioritized dynamic average burst round robin scheduling for uniprocessor and multiprocessor environment	5.0
Banerjee and Padhy (2014)	Comparative analysis of maximum performance round robin (MPRR) by dynamic time quantum with static time quantum	5.0
Khokhar and Kaushik (2017)	Best time quantum round robin CPU scheduling algorithm	5.0
Dash et al. (2015b)	Characteristic specific prioritized dynamic average burst round robin scheduling for uniprocessor and multiprocessor environment	5.0
Banerjee and Padhy (2014)	Comparative analysis of maximum performance round robin (MPRR) by dynamic time quantum with static time quantum	5.0

Table 6 Mapped articles.

Ref.	Title	Score
Banerjee et al. (2015)	Comparative performance analysis of best performance round robin scheduling algorithm (BPRR) using dynamic time quantum with priority based round robin (PBRR) CPU scheduling algorithm in real time systems	5.0
Banerjee et al. (2017b)	Design and analysis of an efficient QoS improvement policy in cloud computing	6.0
Simon et al. (2014)	Dynamic round robin with controlled preemption (DRRCP)	6.0
Rao et al. (2015a)	Dynamic time slice calculation for round robin process scheduling using NOC	5.0
Muraleedharan et al. (2016)	Dynamic time slice round robin scheduling algorithm with unknown burst time	5.0
Fayyaz et al. (2017)	Efficient dual nature round robin CPU scheduling algorithm: a comparative analysis	5.0
Datta (2015)	Efficient round robin scheduling algorithm with dynamic time slice	5.0
Rao et al. (2015b)	Enhanced precedence scheduling algorithm with dynamic time quantum (EPSADTQ)	4.5
Indusree and Prabadevi (2017)	Enhanced round robin CPU scheduling with burst time based time quantum	5.0
Masood and Khader (2015)	Enhanced round robin packet scheduling technique to support multimedia applications in MANET	5.0
Alnowiser et al. (2014)	Enhanced weighted round robin (EWRR) scheduling with DVFS technology in cloud energy-aware	6.0
Bhoi et al. (2014)	Enhancing CPU performance using subcontrary mean dynamic round robin (SMDRR) scheduling algorithm	5.0
Mijinyawa and Abdullahi (2017)	Improved group based time quantum (IGBTQ) CPU scheduling algorithm	6.0
Shyam and Nandal (2014)	Improved mean round robin with shortest job first scheduling	6.0
Ullah (2017)	Improved optimum dynamic time slicing round robin algorithm	4.0
Dorgham and Nassar (2016)	Improved round robin algorithm: proposed method to apply SJF using geometric mean	5.0
Li et al. (2017)	Emulating round-robin in wireless networks	5.0
Parekh and Chaudhari (2016)	Improved round robin CPU scheduling algorithm: round robin, shortest job first and priority algorithm coupled to increase throughput and decrease waiting time and turnaround time	5.0
Ahmed and Muquit (2016)	Improved round robin scheduling algorithm with best possible time quantum and comparison and analysis with the RR algorithm	5.0
Shyam and Kumar (2015)	Improved round robin with shortest job first scheduling	4.5
Goel and Garg (2015)	Improvised optimum multilevel dynamic round robin algorithm for optimizing CPU scheduling	5.5
Roshan and Rao (2016)	Least-mean difference round robin (LMDRR) cpu scheduling algorithm	5.0
Banerjee et al. (2017a)	Mixed round robin scheduling for real time systems	5.0

Table 7 Mapped articles.

Ref.	Title	Score
Mora et al. (2017)	Modified median round robin algorithm (MMRRA)	6.0
Singh and Agrawal (2017)	Modified round robin algorithm (MRR)	5.0
Pradhan et al. (2016)	Modified round robin algorithm for resource allocation in cloud computing	6.0
Merwyn et al. (2014)	Optimal round robin CPU scheduling algorithm using euclidean distance	4.5
Srilatha et al. (2017)	Optimal round robin CPU scheduling algorithm using manhattan distance	4.0
Upadhyay and Hasija (2016)	Optimization in round robin process scheduling algorithm	4.5
Dave et al. (2017)	Optimize task scheduling act by modified round robin scheme with vigorous time quantum	5.0
Damera et al. (2016)	Optimized MCE scheduling algorithm to allocate radio resources using evolved round robin scheduling	5.5
Kiran et al. (2014)	Optimizing CPU scheduling for real time applications using mean-difference round robin (MDRR) algorithm	5.0
Yasin et al. (2015)	Prioritized fair round robin algorithm with variable time quantum	6.0
Riaz et al. (2018)	Randomized dynamic quantum CPU scheduling algorithm	4.5
Garcia-Carballeira and Calderon (2017)	Reducing randomization in the power of two choices load balancing algorithm	5.0
Torjemen et al. (2014)	Scheduling in multi-radio multi-channel mesh networks: brief review and novel approach	5.0
Joshi and Tyagi (2015)	Smart optimized round rRobin (SORR) CPU scheduling algorithm	5.0
McGuire and Lee (2015)	The Adaptive Round Robin Scheduling Algorithm	5.0

5.1 Publication frequency

In this section, we answer the research question RQ1, indicating the number of mapped articles identified from year 2014. The articles were also separated by journals and conferences, c.f. Fig. 2. The year that had more publications was 2014, but it continued intense until 2017. Unlike previous years, in 2017, the number of articles presented in international conferences was higher than the articles published in scientific journals. Despite the proposals of novel variants of the Round Robin algorithm decreased in 2018, they continued to be proposed until current days.

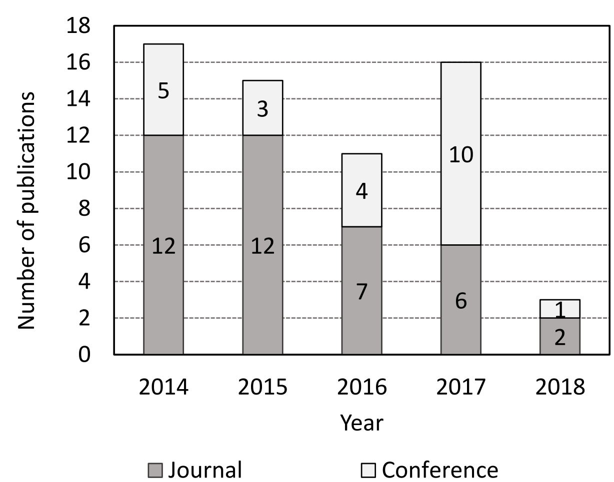

Figure 2 Number of mapping articles by year.

5.2 Publication authors and venues

In this section, we answer the research question RQ2, indicating the main authors and venues of publication regarding Round Robin algorithms proposals. The top three authors found in our mapping study are stated in Table 8. The majority of these authors are linked in Nigeria and India institutions. Saleh E. Abdullahi, researcher by Nile University in Nigeria, was the author that has most mapped articles, reaching five articles between 2014 and 2017. N. Srinivasu, researcher by Koneru Lakshmaiah Education Foundation in India, has four articles in 2014-2015. G. Rama Koteswara Rao also is linked in K L University in India and was author of three of these articles.

The more frequent venues of publications are stated in Table 9. International Journal of Computer Applications and International Journal of Advanced Research in Computer Science and Software Engineering, which have published more articles from 2014 to 2015. The International Conference on Electronics, Computer and Computation (ICECCO) was the conference that has presented more articles, which happened in 2017.

Table 8 Higher frequency of article publications by author.

Author	Instituition	Reference
Saleh E. Abdullahi	Nile University of Nigeria, Nigeria	Simon et al. (2014), Abdulrahim et al. (2014b), Abdulrahim et al. (2014a), Mora et al. (2017), Mijinyawa and Abdullahi (2017)
N. Srinivasu	K L University, India	Rao et al. (2014) Rao et al. (2015a), Srinivasu et al. (2015), Rao et al. (2015b)
G. Rama Koteswara Rao	K L University, India	Rao et al. (2014) , Rao et al. (2015a), Rao et al. (2015b)

Table 9 Higher frequency of article publications by source.

Source Title	Reference
International Journal of Computer Applications	Merwyn et al. (2014), Abdulrahim et al. (2014a), Goel and Garg (2015)
International Journal of Advanced Research in Computer Science and Software Engineering	Abdulrahim et al. (2014b), Shyam and Nandal (2014), Joshi and Tyagi (2015)
International Journal of Electrical and Computer Engineering	Rao et al. (2015a), Srilatha et al. (2017)
Indian Journal of Science and Technology	Banerjee et al. (2015), Muraleedharan et al. (2016)
International Journal of Applied Engineering Research	Rao et al. (2014), Masood and Khader (2015)
International Journal of Computer Science, Engineering and Applications	Mishra and Rashid (2014), Dash et al. (2015b)
Journal of Computer Engineering	Shyam and Kumar (2015), Khatri (2016)
Journal of Global Research in Computer Science	Mishra and Rashid (2014), Bhoi et al. (2014)
Journal of Theoretical and Applied Information Technology	Srinivasu et al. (2015), Roshan and Rao (2016)
International Conference on Electronics, Computer and Computation	Mijinyawa and Abdullahi (2017), Mora et al. (2017)

5.3 Publication quality

In this section, we answer the research questions RQ3, RQ4, and RQ5, indicating the research fields where the Round Robin algorithms were applied and the metrics and tools used to evaluate these algorithms. First, we present an overview of the quality of the articles according to our quality assessment checklist, described in Section 4.2. After, we present the mapped articles grouped by research field and the metrics and tools found.

Table 10 Frequency of answers in the quality assessment checklist.

Question	Nº of answers		
	yes	partial	no
Does the article clearly define its goal?	62	3	0
Does the article present the research field where the algorithm is applied?	65	0	0
Does the article present evaluation metrics for the algorithm?	58	7	0
Does the article present a simulation or an execution of the algorithm?	57	8	1
Does the article present the pseudo-code or code of the algorithm?	52	5	8
Does the article present the tools used to evaluate the algorithm?	16	5	44

According to Table 10, approximately 95% of the mapped articles present clearly their goals; all mapped articles present clearly their research fields; approximately 89% of them present their evaluation metrics; approximately 88% present a simulation of their algorithms; approximately 80% present clearly their pseudo-code; but, only, approximately, 25% articles present their simulation tools to evaluate the proposed algorithm.

5.3.1 Research fields

The number of mapped articles in each research field is shown in Fig. 3. Packet scheduling is the field that has nearly uniform frequency of mapped articles until 2017. CPU scheduling is the field in which there are more articles mapped, been more frequent from 2014-2015. Virtual machine (VM) scheduling is the field in which the publishing frequency articles more varied, did not have mapped articles in the years 2015 nor 2018.

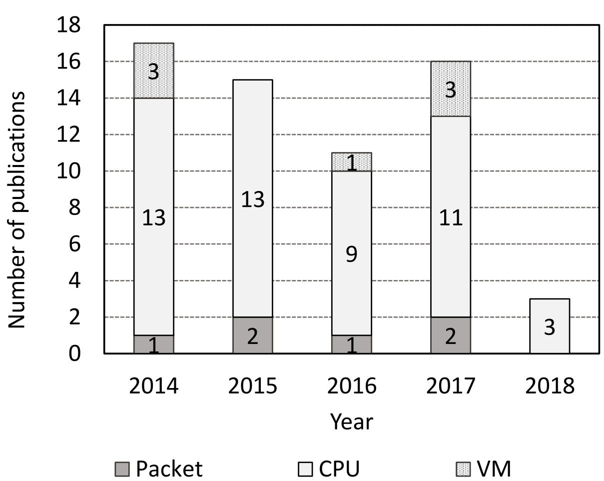

Figure 3 Number of mapping articles in each research field by year.

5.3.2 Metrics

In the packet scheduling research field, there is no standardisation of the metrics used, whereas there are many different metrics, being the most frequent the Average File Delay and the Lost Packet Rate, c.f. Table 11. In CPU scheduling, the Average Turnaround Time, the Average Waiting Time, and the Number Context Switch are metrics used in almost all mapped articles. These last three metrics are also widely used in mapped articles in the field of virtual machine scheduling.

Table 11 Frequency of used metrics in the mapped articles.

Metric	Research Field		
	Packet	CPU	VM
Average File Delay	3	-	-
Average Queue Length	1	-	-
Average Response Time	-	4	1
Average Slowdown	1	-	-
Average Turnaround Time	1	50	3
Average Waiting Time	1	50	3
CPU Utilisation	-	1	1
Data Dropped	1	-	-
End-to-end Delay	-	-	2
Gini Fairness Index	1	-	-
Jair's Fairness Index	1	-	-
Latency Bound	1	-	-
Load Range	-	-	1
Load Variance	-	-	1
Lost Packet Rate	2	-	-
Number Context Switch	-	44	2
Power	-	-	1
Satisfaction User	1	-	-
Scheduling Overhead	-	1	-
Throughput	1	-	1
Variance Inter service Time	1	-	-

5.3.3 Tools

The tools used to simulate the algorithms are rarely mentioned in the mapped articles. In the fields of packet scheduling and virtual machine scheduling, there are specific purpose tools, whereas CPU scheduling there are no cited tools. The mapped articles in CPU scheduling developed simulator using programming languages, being the most frequent Java, C++, Visual Basic, and MatLab, c.f. Table 12.

Amongst the tools used in the algorithms in the field of packet scheduling, we found Network Simulator Version 2 (NS2). NS2 is one of the most used event-driven simulation tool for articleing the dynamic nature

Table 12 Frequency of used tools in the mapped articles.

Tool	**Research Field**		
	Packet	CPU	VM
Simulator Java	1	3	-
Simulator C++	-	2	-
Simulator Visual Basic	-	2	-
Simulator MatLab	2	1	2
Simulator C#	-	1	-
NS2 Network Simulator	2	-	-
CloudSim	-	-	4
SimGrid	-	-	1

of communication networks. NS2 was developed in C++ programming language and Object-oriented Tool Command Language (OTcl) (Issariyakul and Hossain, 2012). As an additional contribution, we revised the technical and scientific literature and found Optimised Network Engineering Tool (OPNET). OPNET is a software simulation package specialised in discrete-event simulation of communication systems. It has an object-oriented and graphics environment that allows the interpretation and synthesises of output data (Chang, 1999).

In field of CPU scheduling, the mapped articles cite simulation tools that were developed by the authors in theirs proposals. The MatLab was the most used programming language to build the simulators. As an additional contribution, we found few simulators for Round Robin algorithms, namely CPU Scheduling Simulator (CPUSS), Modern Operating Systems Simulators (MOSS), Simulator of Operating System Job Scheduling (SOSJS), Optimum Multilevel Dynamic Round Robin Scheduling (OMDRRS), and Interactive Systems Scheduling Simulator (I3S). CPUSS [13] is a framework that allows users to swiftly and easily devise and collects metrics for custom CPU scheduling strategies including First-Come, First-Served, Round Robin, Shortest Job First (SJF), Priority First, and SJF with Priority Elevation rule. Its environment is Windows-DOS. MOSS [14] was developed in Java programming language. SOSJS [15] was developed in Visual Basic 6.0 programming language, implemented the Round Robin and uses the metrics: Average Completion Time and Average Turnaround Times. OMDRRS was also developed in Visual Basic 6.0 programming language, calculates intelligent time slice after every round of execution (Goel and Garg, 2013). I3S is an open educational resource for simulation of the structure, functionality and performance of tasks scheduling in interactive systems, implemented in Python and PHP programming language (Fioravanti et al., 2016).

In the field of virtual machine scheduling, CloudSim and SimGrid are the most widely used tools. CloudSim seems to dominate this field. It enables modelling and simulation of cloud computing systems and application provisioning environments. It is implemented with the Java programming language by Melbourne University, Australia (Goyal et al., 2012). SimGrid is a simulation toolkit for the article of scheduling algorithms for distributed application. It provides a set of core abstractions and functionalities to build simulators for specific application domains (Casanova, 2001). As an additional contribution, we found Eucalyptus that is an open-source software framework for cloud computing that allows users to run and control entire virtual machine instances deployed across a variety of physical resources (Nurmi et al., 2009).

5.4 Variants of Round Robin

In this section, we answer the research question RQ6, by indicating the latest variants of the Round Robin algorithm found in the mapped articles. We list the names followed by a brief description of the algorithm, as follows.

`Best time quantum Round Robin (BQRR)` arranges the tasks in increasing order of their given burst times and chooses best time quantum depending on the mean and median of these burst times (Khokhar and Kaushik, 2017).

`Characteristic Specific Prioritized Dynamic Average Burst Round Robin (CSPDABRR)` uses predefined priority features for calculating priority of tasks and uses dynamic time quantum (Dash et al., 2015b).

`Standard deviation-based quantum Round Robin (DevRR)` calculates the dynamic time quantum based on the standard deviation and average burst time of each task in a queue (Phorncharoen and Sa-Ngiamvibool, 2018).

`Dynamic Average Burst Round Robin (DABRR)` uses dynamic time quantum instead of static time quantum used in Round Robin (Dash et al., 2015a).

`Dynamic Quantum With Re-Adjusted Round Robin (DQRARR)` arranges the tasks in ascending order according to their burst time present in the ready queue and calculated the optimal time quantum (Behera et al., 2011).

`Efficient Dynamic Round Robin (EDRR)` selects a most optimal dynamic time quantum to gain advantage of Round Robin algorithm (Farooq et al., 2017).

`Efficient Dual Nature Round Robin (EDNRR)` uses Round Robin by setting the time quantum in the increasing order and decreasing order (Fayyaz et al., 2017).

`Enhanced Round Robin with Burst-time based Time Quantum (ERRBTQ)` calculates time quantum as per the burst time of tasks already in ready queue (Indusree and Prabadevi, 2017).

`Enhanced Precedence Scheduling Algorithm with Dynamic Time Quantum (EPSADTQ)` suggests that a priority should be assigned to each task based on balanced precedence factor. The method also uses the average as a time quantum (Rao et al., 2015b).

`evolved Round Robin (eRR)` customises dynamic quantum for each service in a real-time environment (Damera et al., 2016).

`Group Based Time Quantum Round Robin (GBTQRR)` focuses on task only, defining a time quantum for every group of tasks (Mijinyawa and Abdullahi, 2017).

`Improved Optimum Dynamic Time Slicing Round Robin Algorithm (IODTSRR)` calculates the time quantum according to the state of queue along with the capability of executing ready tasks arriving at the same or different time (Ullah, 2017).

`Improved Round Robin Cloudlet Scheduling Algorithm (IRRCSA)` is a scheduling algorithm for the field of virtual machine. It uses several task queues and changes the quantum time according to the remaining time to complete the task (Banerjee et al., 2017a).

`Mixed Round Robin Scheduling (MIXED)` is the mix-up of three Scheduling algorithm: min-max, avg-max, and Best Performance Round Robin (Banerjee et al., 2017c).

`Mode Round Robin (MRR)` adjusts the time quantum repeatedly using Mode dispersion measure in accordance with remaining CPU burst time (Verma et al., 2014).

`Modified Round Robin (MRR)` calculates time quantum based on number of tasks and their priority (Singh and Agrawal, 2017).

`Modified Median Round Robin Algorithm (MMRRA)` calculates time quantum based on middle number of the sorted burst time of tasks and on highest burst time (Mora et al., 2017).

`Modified Round Robin Scheme with Vigorous Time Quantum (MRRSVTQ)` repeatedly elects an optimal quantum time during the execution of task, set time quantum actively on the base of task burst time that remains in ready queue (Dave et al., 2017).

`Optimum Dynamic Time Slicing Using Round Robin (ODTSRR)` is proposed for time shared systems and is based upon dynamic time quantum (Hafeez and Rasheed, 2016).

`Optimal Round Robin using Manhattan distance (ORRSM)` determines the time quantum by taking account the similarity or differences of the burst times tasks present in the ready queue (Srilatha et al., 2017).

`Priority Dynamic Quantum Time Scheduling Algorithm (PDQT)` prioritises the tasks in the ready queue and changing the quantum time of each round changing the quantum time for each task in each round depending on its priority (Mohammed et al., 2016).

`Randomized Dynamic Quantum (RDQ)` generates random time quantum for each task (Riaz et al., 2018).

`Shortest Queue of d with Randomisation and Round Robin Policies (SQ-RR(d))` combines randomisation techniques and static local balancing based on Round Robin policy (Garcia-Carballeira and Calderon, 2017).

`Time-Since-Last-Service (TSLS-based)` emulates Round Robin algorithm for wireless networks taking into account wireless interference and channel fading (Li et al., 2017).

As an additional contribution, we revised the technical and scientific literature in current year until April 2019 and found four new proposals, indicating that Round Robin algorithm remains to be a subject of interest of the scientific community.

- `Enhanced Round-Robin-Based Job Scheduling (ERRJS)` calculates time quantum considering the execution time of the tasks in ready queue (Sahu et al., 2019).
- `Fittest Job First Dynamic Round Robin (FJFDRR)` uses dual ready queue and uses a priority factor that considers the arrival time of processes (Manuel et al., 2019).
- `Priority based round robin (PBRR)` prioritises the tasks in the ready queue without changing the quantum time for each task (Zouaoui et al., 2019).
- `Amended Dynamic Round Robinn (ADRR)` selects the tasks that have lower burst times to be executed first (Shafi et al., 2019).

5.5 Citations rank

In this section, we answer the research question RQ7, indicating the ten most cited publications found amongst the mapped articles, c.f. Table 13. We opted to use the same data source to obtain this information for every mapped article. We used the Google Scholar database to have a verification system independent from the single data source (Fregnan et al., 2019). The numbers of citation of the articles were collected in April 2019.

Table 13 Higher frequency of citations by mapped article.

Nº of citations	Title
31	An improved round robin CPU scheduling algorithm with varying time quantum Mishra and Rashid (2014)
22	Enhanced weighted round robin (EWRR) scheduling with DVFS technology in cloud Alnowiser et al. (2014)
21	A new improved round robin (NIRR) CPU scheduling algorithm Abdulrahim et al. (2014a)
19	Modified round robin algorithm for resource allocation in cloud computing Pradhan et al. (2016)
14	A modified round-robin load-balancing algorithm for cluster-based web servers Xu and Wang (2014)
13	An enhanced scheduling in weighted round robin for the cloud infrastructure services Basker et al. (2014)
13	Optimizing CPU scheduling for real time applications using mean-difference round robin (MDRR) algorithm Kiran et al. (2014)
13	An optimized round Robin CPU scheduling algorithm with dynamic time quantum Dash et al. (2015a)
11	Efficient round robin scheduling algorithm with dynamic time slice Datta (2015)

6 Discussion and Threats to Validity

In this section, we evaluated the factors that influence the obtained results and the main limitations of this systematic mapping. We used 11 databases chosen amongst the leading ones of scholarly impact, offering significantly journal coverage and the authors that have higher h-indexes (Powell and Peterson, 2017). However, it is possible that our mapping did not find all existing variants of the Round Robin algorithms, because some authors used others terms and did not include the more general terms "Round Robin algorithm". It is important to note that many of the excluded studies applied some of the variants of the original Round Robin algorithm, but they did not present proposal of improvement or modification of the Round Robin algorithm. We use a single data source to collect the number of citations of the mapped studies, then it is possible that different information is found by other scientific databases. Besides, we recognise the potential limits of this approach because a high number of citations is not necessarily guarantees the important article on the subject (Fregnan et al., 2019). Our research procedure is based on the subjective evaluation of the studies by a single team, so it may not be representative for other research groups. In order to mitigate the possibilities of faults, the group was formed by researchers from different fields of software engineering and from different institutions.

7 Conclusion

Round Robin scheduling algorithm is one of the most popular algorithms. Over the years, new proposals of this algorithm have appeared, seeking to improve it or adapt it to different applications. The systematic mapping presented in this article provides an overview of the proposals of the Round Robin Algorithm, applied to software engineering. To answer our main research question, seven specific questions were defined encompassing publications regarding Round Robin algorithms, panorama, research fields, metrics, and tools used to validate the proposals. We sum up the main conclusions from the results found in our mapping:

RQ1: The articles that proposed variants of the Round Robin algorithm have been published from 2014 to 2018. Between 2014 and 2017, there is a higher frequency of proposals, with an average above 15 new algorithms by year. Until 2015, these articles used to be published in scientific journals, but as off 2016, they have been published in similar quantities, both scientific journals and conferences.

RQ2: The main mapped researchers were: Saleh E. Abdullah, by Nile University in Nigeria; N. Srinivasu and G. Rama Koteswara Rao, both researchers by Koneru Lakshmaiah Education Foundation in India. The more frequent mapped scientific journals were: International Journal of Computer Application and International Journal of Advanced Research in Computer Science and Software Engineering. Both journals published two articles in 2014 and one in 2015. The more frequent mapped conference was the International Conference on Electronics, Computer and Computation (ICECCO), which had two articles presented in 2017.

RQ3: The field of CPU scheduling was that with more proposals mapped, with an average of 10 articles per year. The field of network packet scheduling and the field of virtual machines and grid scheduling had an average of 1 articles per year.

RQ4: In the packet scheduling research field, there is no standardisation of the metrics used, being the most frequent the Average File Delay and the

Lost Packet Rate. In CPU scheduling, the Average Turnaround Time, the Average Waiting Time, and the Number Context Switch are metrics used in almost all mapped articles. These last three metrics are also widely used in mapped articles in the field of virtual machine scheduling.

RQ5: The research fields of packet scheduling and virtual machine scheduling are more advanced in terms of simulation support tools than the field of CPU scheduling. Network Simulator Version 2 (NS2) is the most simulator used in the field of packet scheduling. In the field of CPU scheduling, their researchers have developed its owns tools. In the field of virtual machine scheduling, CloudSim and SimGrid are the most widely used tools.

RQ6: The recent articles regarding Round Robin algorithms reveal that its use on task scheduling is a hot research theme in the world of real-time systems with articles published until current year.

RQ7: The proposal of Mishra and Rashid (2014) were the most cited, with 31 citations. Both the proposal of Alnowiser et al. (2014) and of Abdulrahim et al. (2014a) had more of 20 citations. There are still others with more of 10 citations.

The exhaustive work for the selection of the articles does not ensure that all the variants of the Round Robin algorithm were included in this systematic mapping. However, it ensures a sampling of good quality that can help researchers who wish to know the current proposals of this algorithm. This article showed that the research regarding the improvement in the Round Robin algorithm continues active, shown that Round Robin remains one of the more efficient of scheduling techniques in fields of researches of packets, CPU and virtual machine scheduling. Besides, this mapping study showed that there is a lack of standardisation in relation to metrics in the field of packets scheduling and virtual machine scheduling and of tools for the field of CPU scheduling.

Acknowledgements

This work was supported by the Brazilian Coordination Board for the Improvement of University Personnel (CAPES) and the Research Support Foundation of the State of Rio Grande do Sul (FAPERGS) under grant 17/2551-0001206-2.

References

Abdulrahim A, Abdullahi SE, Sahalu JB (2014a) A new improved round robin (NIRR) CPU scheduling algorithm. International Journal of Computer Applications 90(4):27–33

Abdulrahim A, Aliyu S, Mustapha AM, Abdullahi SE (2014b) An additional improvement in round robin (AAIRR) CPU scheduling algorithm. International Journal of Advanced Research in Computer Science and Software Engineering 4(2):601–610

Agha AEA, Jassbi SJ (2013) A new method to improve round Robin scheduling algorithm with quantum time based on harmonic-arithmetic mean (HARM). International Journal of Information Technology and Computer Science 5(7):56–62

Ahmed WF, Muquit SP (2016) Improved round robin scheduling algorithm with best possible time quantum and comparison and analysis with the RR algorithm. International Research Journal of Engineering and Technology 3(3):1357–1361

Alnowiser A, Aldhahri E, Alahmadi A, Zhu MM (2014) Enhanced weighted round robin (EWRR) with DVFS technology in cloud energy-aware. In: International Conference on Computational Science and Computational Intelligence (CSCI), vol 1, pp 320–326

Alsheikhy A, Ammar R, Elfouly R (2015) An improved dynamic round robin scheduling algorithm based on a variant quantum time. In: International Computer Engineering Conference (ICENCO), pp 98–104

Aslam S, Shah MA (2015) Load balancing algorithms in cloud computing: A survey of modern techniques. In: National Software Engineering Conference (NSEC), pp 30–35

Baek H, Lee J (2019) Improved schedulability analysis of the contention-free policy for real-time systems. Journal of Systems and Software 154:112–124

Banerjee P, Padhy PDLN (2014) Comparative analysis of maximum performance round robin (MPRR) by dynamic time quantum with static time quantum. International Journal of Advanced Research in Computer Science and Software Engineering 4(11):372–377

Banerjee P, Zabin T, ShwetaKumai, Kumari P (2015) Comparative performance analysis of best performance round robin scheduling algorithm (BPRR) using dynamic time quantum with priority based round robin (PBRR) CPU scheduling algorithm in real time systems. Indian Journal of Science and Technology 4(2):151–159

Banerjee P, Kumar B, Banerjee P (2017a) Mixed round robin scheduling for real time systems. International Journal of Computer Trends and Technology 49(3):189–195

Banerjee S, Adhikari M, Biswas U (2017b) Design and analysis of an efficient QoS improvement policy in cloud computing. Service Oriented Computing and Applications 11(1):65–73

Banerjee S, Chowdhury A, Mukherjee S, Biswas U (2017c) An approach towards development of an intelligent cloudlet scheduling mechanism for cloud QoS improvement. International Journal of Hybrid Intelligent Systems 14(1-2):21–30

Basker R, Uthariaraj VR, Devi DC (2014) An enhanced scheduling in weighted round robin for the cloud infrastructure services. International Journal of Recent Advance in Engineering & Technology 2(3):81–86

Behera HS, Mohanty R, Nayak D (2011) A new proposed dynamic quantum with re-adjusted round robin scheduling algorithm and its performance analysis. International Journal of Computer Applications 5(5):10–15

Bhoi SK, Panda SK, Tarai D (2014) Enhancing CPU performance using subcontrary mean dynamic round robin (SMDRR) scheduling algorithm. Journal of Global Research in Computer Science 2(12):17–21

Blot A, Éléonore Kessaci M, Jourdan L (2018) Survey and unification of local search techniques in metaheuristics for multi-objective combinatorial optimisation. Journal of Heuristics 24(6):853–877

Brings J, Daun M, Kempe M, Weyer T (2018) On different search methods for systematic literature reviews and maps: Experiences from a literature search on validation and verification of emergent behavior. In: International Conference on Evaluation and Assessment in Software Engineering, pp 35–45

Bryhni H, Klovning E, Kure O (2000) A comparison of load balancing techniques for scalable web servers. IEEE Network 14(4):58–64

Campelo F, Takahashi F (2019) Sample size estimation for power and accuracy in the experimental comparison of algorithms. Journal of Heuristics 25(2):305–338

Casanova H (2001) Simgrid: a toolkit for the simulation of application scheduling. In: International Symposium on Cluster Computing and the Grid (CCGrid), pp 430–437

Chang X (1999) Network simulations with OPNET. In: Winter Simulation Conference (WSC), vol 1, pp 307–314

Corstjens J, Dang N, Depaire B, Caris A, Causmaecker PD (2018) A combined approach for analysing heuristic algorithms. Journal of Heuristics pp 1–38

Damera B, Babu PC, Mohamed JS (2016) Optimized MCE scheduling algorithm to allocate radio resources using evolved round robin scheduling. In: International Conference on Applied and Theoretical Computing and Communication Technology (ICATCCT), pp 770–775

Dash AR, Sahu SK, Samantra SK (2015a) An optimized round robin CPU scheduling algorithm with dynamic time quantum. International Journal of Computer Science, Engineering and Information Technology 5(1):7–26

Dash AR, Sahu SK, Samantra SK, Sabat S (2015b) Characteristic specific prioritized dynamic average burst round robin scheduling for uniprocessor and multiprocessor environment. International Journal of Computer Science, Engineering and Applications 5(4-5):1–20

Datta L (2015) Efficient round robin scheduling algorithm with dynamic time slice. International Journal of Education and Management Engineering 5(2):10–19

Dave B, Yadev S, Mathuria M, Sharma YM (2017) Optimize task scheduling act by modified round robin scheme with vigorous time quantum. In: International Conference on Intelligent Sustainable Systems (ICISS), pp 905–910

Dorgham OHM, Nassar MO (2016) Improved round robin algorithm: Proposed method to apply SJF using geometric mean. International Journal of Advanced Studies in Computers, Science and Engineering 5(11):112–119

Farooq MU, Shakoor A, Siddique AB (2017) An efficient dynamic round robin algorithm for CPU scheduling. In: International Conference on Communication, Computing and Digital Systems (C-CODE), pp 244–248

Fataniya B, Patel M (2018a) Dynamic time quantum approach to improve round robin scheduling algorithm in cloud environment. International Journal of Scientific Research in Science, Engineering and Technology 4(4):709–711

Fataniya B, Patel M (2018b) Survey on different method to improve performance of the round robin scheduling algorithm. International Journal of Scientific Research in Science, Engineering and Technology 4(2):69–77

Fayyaz S, Hamza HA, Din SMU, Iqra (2017) Efficient dual nature round robin CPU scheduling algorithm: A comparative analysis. International Journal of Multidisciplinary Sciences and Engineering 8(3):21–26

Feld T, Biondi A, Davis RI, Buttazzo G, Slomka F (2018) A survey of schedulability analysis techniques for rate-dependent tasks. Journal of Systems and Software 138:100–107

Fioravanti ML, Kamada MK, Barbosa EF (2016) I3S simulator: An open educational resource for teaching scheduling in interactive systems. Nuevas Ideas en Informática Educativa 12(1):704–709

Frantz RZ, Corchuelo R, Roos-Frantz F (2016) On the design of a maintainable software development kit to implement integration solutions. Journal of Systems and Software 111:89–104

Fregnan E, Baum T, Palomba F, Bacchelli A (2019) A survey on software coupling relations and tools. Information and Software Technology 107:159–178

Garcia-Carballeira F, Calderon A (2017) Reducing randomization in the power of two choices load balancing algorithm. In: International Conference on High-Performance Computing & Simulation (HPCS), pp 365–372

Ghosh A (2017) Balanced path selection in cloud computing using round robin algorithm. International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2(4):963–969

Goel N, Garg RB (2013) Simulation of an optimum multilevel dynamic round robin scheduling algorithm. International Journal of Computer Applications 76(7):42–46

Goel N, Garg RB (2015) Improvised optimum multilevel dynamic round robin algorithm for optimizing CPU scheduling. International Journal of Computer Applications 123(16):1–5

Gope P, Singh A, Sharma A, Pahwa N (2011) An enhanced round robin policy for real time traffic. In: International Conference on Advances in Communication, Network, and Computing (CNC), pp 44–50

Goyal T, Singh A, Agrawal A (2012) Cloudsim: simulator for cloud computing infrastructure and modeling. Procedia Engineering 38(4):3566–3572

Guerrero C, Lera I, Juiz C (2019) Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures. Future Generation Computer Systems 97:131–144

Hafeez MS, Rasheed F (2016) An optimum dynamic time slicing scheduling algorithm using round robin approach. International Journal of Computer Science and Information Security 14(6):778–798

Hamouda H, Kabaou MO, Bouhlel MS (2017) A downlink resources allocation scheme for multimedia applications in OFDMA wireless systems. In: International Conference on Advanced Information Networking and Applications (AINA), pp 31–38

Hanafi S, RacaTodosijević (2017) Mathematical programming based heuristics for the 0–1 MIP: a survey. Journal of Heuristics 23(4):165–206

Hiranwal S, Roy KC (2011) Adaptive round robin scheduling using shortest burst approach based on smart time slice. International Journal of Computer Science and Communication 2(2):319–323

Hu J, Gu J, Sun G, Zhao T (2010) A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In: International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), pp 89–96

Indusree JR, Prabadevi B (2017) Enhanced round robin CPU scheduling with burst time based time quantum. In: IOP Conference Series: Materials Science and Engineering (ICSET), vol 263, pp 1–8

Issariyakul T, Hossain E (2012) Introduction to network simulator 2 (NS2). In: Introduction to Network Simulator NS2, pp 21–40

Joshi R, Tyagi SB (2015) Smart optimized round robin (SORR) CPU scheduling algorithm. International Journal of Advanced Research in Computer Science and Software Engineering 5(7):568–574

Kathuria S, Singh PP, Tiwari P, Prashant (2016) A revamped mean round robin (RMRR) CPU scheduling algorithm. International Journal of Innovative Research in Computer and Communication Engineering 4(4):6684–6691

Khatri J (2016) An enhanced round robin CPU scheduling algorithm. Journal of Computer Engineering 18(4):20–24

Khokhar D, Kaushik A (2017) Best time quantum round robin CPU scheduling algorithm. International Journal of Scientific Engineering and Applied Science 3(5):213–217

Kiran RNDSS, Babu PV, Krishna BBM (2014) Optimizing CPU scheduling for real time applications using mean-difference round robin (MDRR) algorithm. In: Convention of Computer Society of India (CSI), vol 1, pp 713–721

Kitchenham B, Stuart C (2007) Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE-2007-01, School of Computer Science and Mathematics, Keele University, UK

Kitchenham B, Pretorius R, Budgen D, Brereton OP, Turner M, Niazi M, Linkman S (2010) Systematic literature reviews in software engineering – a tertiary study. Information and Software Technology 52(8):792–805

Kochovski P, Drobintsev PD, Stankovski V (2019) Formal quality of service assurances, ranking and verification of cloud deployment options with a probabilistic model checking method. Information and Software Technology 109:14–25

Kumar A, Pandey DS, Prakash V (2019) A survey: Load balancing algorithm in cloud computing. In: International Conference on Advanced Computing and Software Engineering (ICACSE), pp 408–412

Kumar MRM, Rajendra BR, Sreenatha M, Niranjan CK (2014) An improved approach to minimize context switching in round robin scheduling algorithm using optimization techniques. International Journal of Research in Engineering and Technology 3(4):262–271

Lakshmanan S (2017) A survey on heterogeneous resource scheduling algorithm in cloud computing. International Journal of Scientific Research in Science, Engineering and Technology 3(8):709–716

Li B, Eryilmaz A, Srikant R (2017) Emulating round-robin in wireless networks. In: International Symposium on Mobile Ad Hoc Networking and Computing (MOBHOC), pp 1–10

Magdich A, Kacem YH, Kerboeuf M, Mahfoudhi A, Abid M (2018) A design pattern-based approach for automatic choice of semi-partitioned and global scheduling algorithms. Information and Software Technology 97:83–98

Mahdavi-Hezavehi S, Durelli VH, Weyns D, Avgeriou P (2017) A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems. Information and Software Technology 90:1–26

Manikas K (2016) Revisiting software ecosystems research: A longitudinal literature study. Journal of Systems and Software 117:84–103

Maniyar B, Kanani B (2015) Review on round robin algorithm for task scheduling in cloud computing. Journal of Emerging Technologies and Innovative Research 2(2):788–793

Manuel JIC, Baquirin RBM, Guevara KS, Tandingan DR (2019) Fittest job first dynamic round robin (FJFDRR) scheduling algorithm using dual queue and arrival time factor: a comparison. IOP Conference Series: Materials Science and Engineering 482:1–9

Masood MS, Khader PSA (2015) Enhanced round robin packet scheduling technique to support multimedia applications in MANET. International Journal of Applied Engineering Research 10(1):8699–8713

McGuire C, Lee J (2015) The adaptive80 round robin scheduling algorithm. In: World Congress on Engineering and Computer Science (WCECS), pp 243–258

Merwyn D, Caiero F, Surlakar S (2014) Optimal round robin CPU scheduling algorithm using euclidean distance. International Journal of Computer Applications 96(18):8–11

Mijinyawa HM, Abdullahi SE (2017) Improved group based time quantum (IGBTQ) CPU scheduling algorithm. In: International Conference on Electronics, Computer and Computation (ICECCO), pp 1–6

Mishra MK, Rashid F (2014) An improved round robin CPU scheduling algorithm with varying time quantum. International Journal of Computer Science, Engineering and Applications 4(4):1–8

Mittal N, Garg K, Ameria A (2015) A paper on modified round robin algorithm. International Journal of Latest Technology in Engineering, Management & Applied Science 4(11):93–98

Mohammed MA, AbdulMajid M, Mustafa BA (2016) A proposed priority dynamic quantum time algorithm to enhance varying time quantum round robin algorithm. International Journal of Computer Applications in Technology 54(3):184–191

Mora H, Abdullahi SE, Junaidu SB (2017) Modified median round robin algorithm (MMRRA). In: International Conference on Electronics, Computer and Computation (ICECCO), pp 1–7

Muraleedharan A, Antony N, Nandakumar R (2016) Dynamic time slice round robin scheduling algorithm with unknown burst time. Indian Journal of Science and Technology 9(8):1–6

Musa KI, Lasisi KE, Gokir J (2018) Comparative performance and analysis of some improved round robin CPU scheduling. International Journal of Scientific & Engineering Research 8(8):1494–1502

Negi S, Kalra P (2014) A Comparative Performance Analysis of Various Variants of Round Robin Scheduling Algorithm. International Journal of Information & Computation Technology 4(7):765–772

Nurmi D, Wolski R, Grzegorczyk C, Obertelli G, Soman S, Youseff L, Zagorodnov D (2009) The eucalyptus open-source cloud-computing system. In: International Symposium on Cluster Computing and the Grid (CCGrid), pp 124–131

Parekh HB, Chaudhari S (2016) Improved round robin CPU scheduling algorithm: Round robin, shortest job first and priority algorithm coupled to increase throughput and decrease waiting time and turnaround time. In: International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), pp 184–187

Petersen K, Vakkalanka S, Kuzniarz L (2015) Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64:1–18

Phorncharoen S, Sa-Ngiamvibool W (2018) A proposed round robin scheduling algorithm for enhancing performance of CPU utilization. Przeglad Elektrotechniczny 94(4):26–29

Powell KR, Peterson SR (2017) Coverage and quality: A comparison of Web of Science and Scopus databases for reporting faculty nursing publication metrics. Nursing Outlook 65(5):572–578

Pradhan P, Behera PK, Ray BNB (2016) Modified round robin algorithm for resource allocation in cloud computing. In: International Conference on Computational Modeling and Security (CMS), vol 85, pp 878–890

Qin C, Eichelberger H, Schmid K (2019) Enactment of adaptation in data stream processing with latency implications—a systematic literature review. Information and Software Technology 1–21

Ramakrishna M, Rao GPR (2013) Efficient round robin CPU scheduling algorithm for operating systems. International Journal of Innovative Technology and Research 1(1):103–109

Rao G, Srinivasu N, Kumar KG, Abhishek B (2014) An enhanced dynamic round robin CPU scheduling algorithm. International Journal of Applied Engineering Research 9(15):3085–3098

Rao GSN, Srinivasu N, Srinivasu SVN, Rao GRK (2015a) Dynamic time slice calculation for round robin process scheduling using NOC. International Journal of Electrical and Computer Engineering 5(6):1480–1485

Rao GSN, Srinivasu SVN, Srinivasu N, Rao GR (2015b) Enhanced precedence scheduling algorithm with dynamic time quantum (EPSADTQ). Research Journal of Applied Sciences, Engineering and Technology 10(8):938–941

Rezende AV, Silva L, Britto A, Amaral R (2019) Software project scheduling problem in the context of search-based software engineering: A systematic review. Journal of Systems and Software 155:43–56

Riaz R, Kazmi SH, Kazmi ZH, Shah SA (2018) Randomized dynamic quantum CPU scheduling algorithm. Journal of Information Communication Technologies and Robotic Applications 9(2):19–23

Roshan DR, Rao KS (2016) Least-mean difference round robin (LMDRR) CPU scheduling algorithm. Journal of Theoretical & Applied Information Technology 88(1):51–56

Sabha SU (2018) A novel and efficient round robin algorithm with intelligent time slice and shortest remaining time first. Materials Today: Proceedings 5(5):12009–12015

Sahu T, Verma SK, Shakya M, Pandey R (2019) An enhanced round-robin-based job scheduling algorithm in Grid computing. In: International Conference on Computer Networks and Communication Technologies (ICCNCT), pp 799–807

Samal P, Mishra P (2013) Analysis of variants in round robin algorithms for load balancing in cloud computing. International Journal of computer science and Information Technologies 4(3):416–419

Shafi U, Shah M, Wahid A, Kamran M, Javaid Q, Asghar M, Haider M (2019) A novel amended dynamic round robin scheduling algorithm for timeshared systems. International Arab Journal of Information Technology 16(4):1–9

Shyam R, Kumar P (2015) Improved round robin with shortest job first scheduling. International Journal of Advanced Research in Computer Science and Software Engineering 5(3):156–162

Shyam R, Nandal SK (2014) Improved mean round robin with shortest job first scheduling. International Journal of Advanced Research in Computer Science and Software Engineering 4(7):170–179

Simon A, Abdullahi S, Junaidu S (2014) Dynamic round robin with controlled preemption (DRRCP). International Journal of Computer Science Issues 11(3):109–117

Singh M, Agrawal R (2017) Modified round robin algorithm (MRR). In: International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp 2832–2839

Srilatha N, Sravani M, Divya Y (2017) Optimal round robin CPU scheduling algorithm using manhattan distance. International Journal of Electrical and Computer Engineering 7(6):3664–3668

Srinivasu N, Balakrishna ASV, Lakshmi RD (2015) An augmented dynamic round robin CPU scheduling algorithm. Journal of Theoretical & Applied Information Technology 76(1):118–126

Srivastava A, Kumar D (2019) Analysis of round robin scheduling algorithms in CPU scheduling. In: International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM), pp 378–384

Tajwar MM, Pathan MN, Hussaini L, Abubakar A (2017) CPU scheduling with a round robin algorithm based on an effective time slice. Journal of Information Processing Systems 13(4):941–950

Topcuoglu HR, Hariri S, Wu MY (2002) Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Transactions on Parallel and Distributed Systems 13(3):260–274

Torjemen N, Tabbane N, Baklouti H, Tabbane S (2014) Scheduling in multi-radio multi-channel mesh networks: Brief review and novel approach. In: International Symposium on Networks, Computers and Communications (ISNCC), pp 1–5

Ullah S (2017) Improved optimum dynamic time slicing round robin algorithm. In: International Conference on Electrical Information and Communication Technology (EICT), pp 1–6

Upadhyay A, Hasija H (2016) Optimization in round robin process scheduling algorithm. In: Information Systems Design and Intelligent Applications (INDIA), pp 457–467

Verma R, Mittal S, Singh V (2014) A round robin algorithm using mode dispersion for effective measure. International Journal for Research in Applied Science and Engineering Technology 2(3):166–174

Wason R, Dagaur S, Prashar A, Bansal C, Kaur M, Maini S (2016) Novel simulation of computer memory management. In: International Conference on Computing for Sustainable Global Development (INDIACom), pp 945–950

Xu L, Chen Y, Gong Z (2014) An improving algorithm for combined input-crosspoint-queued switches. In: International Conference on Trustworthy Computing and Services (ISCTCS), pp 281–287

Xu Z, Wang X (2014) A modified round-robin load-balancing algorithm for cluster-based web servers. In: Chinese Control Conference (CCC), pp 3580–3584

Yasin A, Faraz A, Rehman S (2015) Prioritized fair round robin algorithm with variable time quantum. In: International Conference on Frontiers of Information Technology (FIT), pp 314–319

Zeng D, Gu L, Guo S, Cheng Z, Yu S (2016) Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Transactions on Computers 65(12):3702–3712

Zhang T, Pota H, Chu CC, Gadh R (2018) Real-time renewable energy incentive system for electric vehicles using prioritization and cryptocurrency. Applied Energy 226:582–594

Zouaoui S, Boussaid L, Mtibaa A (2019) Priority based round robin (PBRR) CPU scheduling algorithm. International Journal of Electrical and Computer Engineering 9(1):19–202

Ákos Szőke (2011) Conceptual scheduling model and optimized release scheduling for agile environments. Information and Software Technology 53(6):574–591