A Survey on Scheduling Algorithms in Real-Time Systems

Wafa Hantom, Alanoud Aldweesh, Reem Alzaher and Atta-ur-Rahman
Department of Computer Science, College of Computer Science and Information Technology,
Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
Email: {2220500182, 22220500188, 22220500186, *aaurrahman}@iau.edu.sa

Abstract

The scheduling algorithm is an essential part of real-time systems, and there are many different scheduling algorithms due to the changing needs and requirements of different real-time systems. The choice of algorithm is critical in any real-time system and is greatly influenced by the type of system such as preemptive or non-preemptive operating systems. Also, the number of the processors (i.e., uniprocessor or multiprocessor) is an essential factor in the choice of the scheduling algorithms. In this paper, we classify scheduling algorithms that are used in Real-time systems into two categories, Uniprocessor scheduling algorithms, and Multiprocessor scheduling algorithms.

Keywords: *Scheduling Algorithms; Real-time Systems; Priority-driven Algorithms; EDF (Earliest Deadline First); RM (Rate Monotonic)*

I. INTRODUCTION

Real-time systems are defined as those systems in which the accuracy of the system depends not only on the logical result of the computation but also on the production of results in the specified time. Real-time systems are used in a variety of applications such as critical safety systems, control units in power plants, satellite controllers, command systems, and flight control systems. Real-time systems can be categorized into hard real-time systems and soft real-time systems. In hard real-time systems, the responses must occur within the required deadline. Otherwise, missing the deadline may result in huge losses and dangerous consequences. For example, missile control systems. Soft real-time are those systems where deadlines are important but will still function properly if deadlines are not met because the task can be rescheduled or can be completed after the specified time. For example, multimedia and gaming systems [1]. This paper discusses the most used algorithms in Real-Time systems, which are Rate Monotonic and Earliest Deadline First algorithms as well as an explanation of different scheduling algorithms with their respective pros and cons and suitability towards the nature of real time system like whether it is soft real time, hard real time.

The remaining part of this work is organized as follows. Section II contains a review of related literature. Section III discusses the scheduling algorithms for uniprocessor systems, such as RM and EDF. Section IV represents an overview of scheduling algorithms for multiprocessor systems, as well as a comparison between uniprocessor and multi-processors scheduling algorithms. Section V contains the conclusion and recommendation emanating from this work.

Manuscript received April 5, 2022
Manuscript revised April 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.4.80

I. REVIEW OF RELATED WORK

Many research works have been done on scheduling algorithms in real-time systems. The researchers aim to find the optimal algorithm for such systems. This section represents some of these works.

This paper [2] discusses the scheduling algorithm in the real-time system in terms of the effect of the quality of the real-time scheduling algorithm on the real-time system throughput capacity, response time, and this paper also discusses the features and performance of the real-time system according to the system environment, splitting the real-time system into single processor scheduling, multiprocessor scheduling, distributed scheduling, real-time scheduling algorithms RMS, EDF, and LLF in a single processor.

This paper [3] presents a real-time domain summary in scheduling and operating systems Where four scheduling models are discussed: static scheduling, pre-emptive scheduling with fixed priority, dynamic scheduling, and dynamic scheduling for best effort. operating systems in real-time.

In [4], the authors focus on making some improvements to Earliest First Deadline (EDF) Algorithms in order to reduce the number of relay tasks in addition to the ability to predict their behavior. The earliest first deadline (EFDF) is known. Displays algorithms at the very least complexity by Performance analysis. Based on the results of the experiment, it was found that the earliest deadline first (EFDF) algorithm reduced the complexity time in older tasks, deadline first (EDF) scheduling algorithm in a real-time system. In a multiprocessor system.

In [5], They concluded That the EDF scheduling algorithm is an optimal scheduling algorithm for single processors, but it has received little attention from the industry. Fixed Priority, on the other hand, is relatively popular with many commercial real-time operating systems despite offering lower theoretical schedulable processor utilization.

In [6], they presented an optimal real-time scheduling algorithm for multiprocessors, which is not based on time quanta called LLREF designed based on a technique of using the T-L plane abstraction for reasoning about multiprocessor scheduling. It showed that scheduling for multiprocessors can be viewed as repeatedly occurring T-L planes, and correct scheduling on a single T-L plane leads to the optimal solution for all times.

Authors in [7] talked about that EDF algorithm schedules real-time tasks are based on their deadlines plus that EDF is widely studied as a dynamic priority-driven scheduling scheme because of its optimality for periodic, aperiodic, and sporadic preemptive tasks, optimality for sporadic non-preemptive tasks, and acceptable performance for periodic and aperiodic non-preemptive tasks.

EDF can achieve the highest possible processor utilization for preemptive tasks. Although finding an optimal schedule for periodic and aperiodic non-preemptive tasks is NP-hard [8]. In [9], experiments show that EDF can achieve very good results even when the system is lightly loaded. When the processor is overloaded (i.e., the combined requirements of pending tasks exceed the system's capabilities), EDF performs poorly. Researchers have proposed several adaptive techniques for dealing with heavily loaded situations, but they all require the detection of the overload condition. [10]

Scheduling algorithms play an effective role in the environments where the there is a greater number of users compared to the available resources. Other than that, several constraints are imposed that must be satisfied alongside [16][17][18]. These constraints can be priority of the user or its processes, power consumption constraints [19][20].

These issues are further maximized in the cloud environment [21][22]. Where there are several resources must be optimally utilized by the several number of users. To appropriately define the scope several layers are introduced in the mobile cloud computing paradigm like the internet of things (IoT), mist computing, edge computing and fog computing layers etc. [23][24][25]. The optimum resource allocation results in enhanced quality of service and maximizing the benefits obtained by the resources with users' satisfaction [26][27][28].

II. Scheduling Algorithms for Uniprocessors

Real-time systems that used a single processor have various scheduling algorithms. As shown in Fig. 1, These algorithms can be classified into static, and priority-driven algorithms. The static category involves many algorithms such as Round Robin (RR) in which the processor time is divided equally among the tasks. The other category is priority-driven algorithms, and it is the focus of this section.

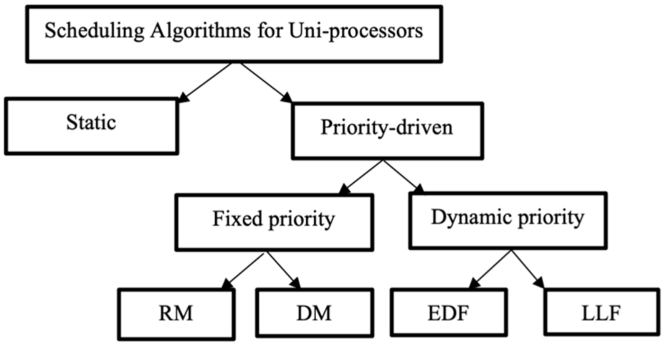

Fig. 1. Classification of Scheduling Algorithms for Uniprocessor [11]

A. Priority-driven Scheduling Algorithms

As represented in Fig. 1, the priority-driven scheduling algorithms are divided into fixed and dynamic. This classification is based on the priority assignment whether the priority is static or changed at running time. This section represents an overview of the most widely used priority-driven algorithms in real-time systems which are EDF and RM.

The Rate Monotonic Algorithm is another name for the RM Scheduling Algorithm. The RM algorithm is a fixed or static priority scheduling algorithm. Tasks are preferred by RM based on their period. The disadvantage of this algorithm is that it does not provide a perfect result in a low-load situation. When compared to dynamic scheduling, RM performs better in overloaded situations. In the RM algorithm, the shortest period gives the most chances to execute. [12]

The earliest deadline first scheduling algorithm is also known as the nearest deadline first scheduling algorithm. The EDF algorithm is a dynamic scheduling algorithm. The task must be completed as soon as possible. The task with the earliest deadline has the highest priority. EDF Scheduling provides 100 percent task utilization under loaded conditions or when the utilization is less than or equal to 1. In contrast, when task utilization is more than the cross-load factor or slightly overloaded, the utilization of the processor decreases exponentially [12]. Table I shows the advantages and disadvantages of RM and EDF.

TABLE I. RM and EDF Advantages and Disadvantages

Algorithms	Advantages	Disadvantages
RM (Rate Monotonic)	• Simple to implement. • Commonly used algorithm.	• Waste CPU utilization
EDF (Earliest Deadline First)	• Full process utilization	• Difficult implantation • Misbehave in overloaded conditions

Fig. 2 represents a case study of how RM and EDF behave on the same task set. Suppose a task set consists of three tasks where each task Ti is represented by its computation time and the period, Ci and Pi, respectively. The tasks are T1(2,6), T2(3,8), T3(2,12). As it is shown in Figure 2(a), the priority in RM is assigned based on the period. So, the task with the lowest period has the highest priority. In Figure 2(b), the EDF's priority is changed based on the task deadline. So, the task with the shortest deadline at each time interval has the highest priority.

Fig. 3 represents another case study of how RM misbehaves in some conditions. Suppose a task set consists of two tasks T1(2,5), T2(4,7). As it is shown in Figure 3(a), since T1 has a higher priority than T2, T1 will preempt every instance of T2, and sometimes it may cause a deadline missing. In contrast, in Figure 3(b), the EDF's is able to schedule this task set because it doesn't cause any deadline missing. As stated by [13], "For larger task sets, the number of preemptions caused by RM increases, thus the overhead due to the context switch time is higher under RM than EDF".

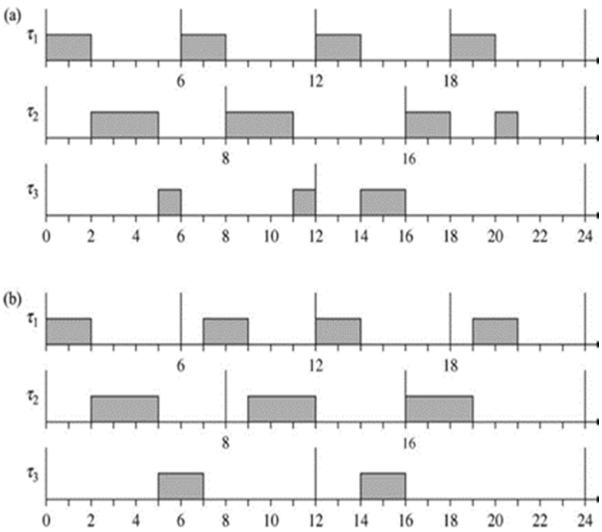

Fig. 2. (a) RM and (b) EDF scheduling comparison [13]

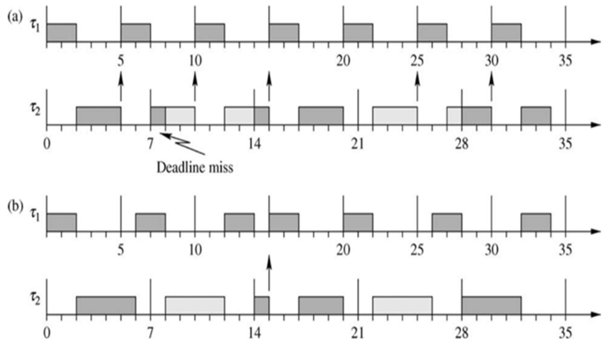

Fig. 3. (a) RM and (b) EDF scheduling comparison [13]

III. Scheduling Algorithms for Multiprocessors

As time goes, the need for more than one processor is increased to perform more complex and heavier computations. Multiprocessor systems require a different scheduling scheme than uniprocessor. Many research works have been done in this field to obtain the best scheduling algorithm. Fig. 4 represents the algorithm's classification of multiprocessor systems. They are divided into classic and heuristic and the evolutionary algorithms. In the classic category, most algorithms are not exclusively created to be used in multiprocessor environments, however, they achieve less time complexity in multiprocessor systems compared to other categories. One drawback of classic algorithms is that they don't guarantee an optimal solution. The other category is heuristic & evolutionary algorithms, which achieve a near-optimal solution but with more running time.

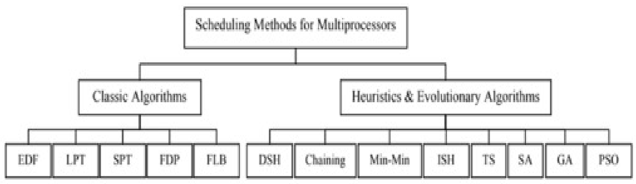

Fig. 4. Scheduling algorithms for multiprocessor systems [14]

TABLE II. Comparison between Uniprocessor and Multiprocessor Algorithms [15]

Performance Metric \ Algorithms	Uniprocessor Algorithms				Multiprocessor Algorithms			
	EDF	**LLF**	**MUF**	**IUF**	**EDZL**	**ILLF**	**MMUF**	**MIUF**
Priority	D	D	Hybrid	D	D	D	Hybrid	D
CPU Utilization	High	High	High	High	High	High	High	High
No of Context Switching	Less	High	High	High	Very less	Less	Less	Less
Optimal	Yes	Yes	for critical tasks	Yes	No	Yes	Yes	Yes
Deadline miss chances	Average	Average	Less	Less	Less	Less	Less	Very Less
Response Time	High	Average	Low	High	Low	High	Average	Low
Predictability	Not Predictable	Not Predictable	Predictable under transient load	Dynamic predictability	More Predictable than EDF	More Predictable	Predictable under transient load	Dynamic predictability
Effectiveness	Optimal, Easy to implement	Takes execution time into cosideration	Work in transient overload	maximize utilization bound of schedule	Context switching overhead is low	Less context switching	Optimal for noncritical tasks	Improves context switching, response time and CPU utilization
Limitations	Not Work in overload, not optimal for pro.>1	In laxity tie, more context switches occurs	Non critical task may miss deadline	Context switching is very high	chances of deadline miss of the critical tasks	Execut-ion time is more	Only consider static utilization of task set	

In Table II, authors in [15] represent a comparison of some uniprocessor and multiprocessor scheduling algorithms and compare them from different metrics such as priority, CPU utilization, number of contexts switching, optimality, deadline miss chances, response time, predictability, effectiveness, and limitations. In conclusion, the Instantaneous utilization factor (IUF) scheduling Algorithm performs better than other uniprocessor algorithms, where the Modified instantaneous utilization factor (MIUF) gives better response time, CPU utilization, and context switching compared to other multiprocessor algorithms [29][30].

IV. CONCLUSION AND RECOMMENDATION

This paper provides an overview of scheduling algorithms in real-time systems. The earlier studies have been reviewed and discussed in the field of scheduling algorithms in real-time systems. This paper is also discussed the most used algorithms in uniprocessor systems, which are RM and EDF. In addition, an overview of multiprocessor scheduling algorithms has been illustrated. In conclusion, the choice of scheduling algorithm is affected by many factors and there is no way to state an optimal algorithm for all systems, as each system is different in its structure and needs. As future work, the application of evolutionary algorithms in the scheduling represents an area for future work and research.

REFERENCES

[1] G. Bernat, A. Burns, and A. Llamosí, "Weakly hard real-time systems," IEEE Transactions on Computers, vol. 50, no. 4, pp. 308–321, Apr. 2001, doi: 10.1109/12.919277.

[2] L. Jie, G. Ruifeng, and S. Zhixiang, "The research of scheduling algorithms in real-time system," in CCTAE 2010 - 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering, 2010, vol. 1, pp. 333–336. doi: 10.1109/CCTAE.2010.5544771.

[3] K. Ramamritham and J. A. Stankovic, "Scheduling Algorithms and Operating Systems Support for Real-Time Systems."

[4] J. Singh, "An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in Real-Time System." [Online]. Available: https://www.researchgate.net/publication/48176486

[5] W. Lunniss, S. Altmeyer, and R. I. Davis, "A Comparison between Fixed Priority and EDF Scheduling accounting for Cache Related Pre-emption Delays", doi: 10.4230/LITES-v001-i001-a001.

[6] H. Cho and E. Douglas Jensen, "An Optimal Real-Time Scheduling Algorithm for Multiprocessors," 2006.

[7] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of periodic and sporadic tasks," in Proceedings - Real-Time Systems Symposium, 1991, pp. 129–139. doi: 10.1109/real.1991.160366.

[8] F. Brauer, M. R. Garey, D. S. J. W. H. Freeman, and S. Francisco, "Computers and Intractability: A Guide to the Theory o[NP-Completeness."

[9] D. Locke, "Best-effort decision making for real-time scheduling." [Online]. Available: https://.researchgate.net/publication/238690426

[10] L. Georges, P. Mühlethaler, and N. Rivierre, "A Few Results on Non-Preemptive Real Time Scheduling A Few Results on Non-Preemptive Real Time Scheduling. [Research Report] RR-3926, INRIA. 2000.

[11] D. G. Harkut, M. S. Ali, M. Poonam Lohiya, B. Principal, and nd yr, "Real-time Scheduler For Wireless Sensor Network : A Review." [Online]. Available: www.ijert.org

[12] H. Thakar, "Comparison between EDF_RM and EDF_DM in dynamic scheduling algorithm with sporadic task," vol. 1, 2016, [Online]. Available: www.ijsdr.org49

[13] G. C. Buttazzo, "Rate Monotonic vs. EDF: Judgment Day," 2005.

[14] M. Rouhifar and R. Ravanmehr, "A Survey on Scheduling Approaches for Hard Real-Time Systems," International Journal of Computer Applications, vol. 131, no. 17, pp. 41–48, Dec. 2015, doi: 10.5120/ijca2015907656.

[15] S. Pandit and R. Shedge, "Survey of Real Time Scheduling Algorithms." [Online]. Available: www.iosrjournals.orgwww.iosrjournals.org.

[16] A. Rahman, "GRBF-NN based ambient aware realtime adaptive communication in DVB-S2." J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-02174-w.

[17] A. Rahman, S. Dash, & A.K. Luhach, "Dynamic MODCOD and power allocation in DVB-S2: a hybrid intelligent approach." Telecommun Syst, vol. 76, pp. 49–61, 2021.

[18] A. Rahman, S. Dash, A.K. Luhach, N. Chilamkurti, S. Baek, Y. Nam, "A Neuro-Fuzzy Approach for User Behavior Classification and Prediction", Journal of Cloud Computing, 8(17), 2019.

[19] A. Rahman, "Memetic Computing based Numerical Solution to Troesch Problem", Journal of Intelligent and Fuzzy Systems, 37(1):1545-1554, 2019.

[20] A. Rahman, "Optimum Information Embedding in Digital Watermarking", Journal of Intelligent and Fuzzy Systems, 37(1):553-564, 2019.

[21] S. Dash, S. Biswas, D. Banerjee, A. Rahman, "Edge and Fog Computing in Healthcare – A Review", Scalable Computing 20(2):191-206, May 2019.

[22] A. Rahman, S. Dash, M. Ahmad, T. Iqbal, "Mobile Cloud Computing: A Green Perspective," Intelligent Systems, Lecture Notes in Networks and Systems book series (LNNS, volume 185), pp. 523-533, 2021.

[23] F. Alhaidari, S. H. Almotiri, M. A. Ghamdi, M. A. Khan, A. Rehman, S. Abbas, K.M. Khan, A. Rahman, "Intelligent software-defined network for cognitive routing optimization using deep extreme learning machine approach," Computers, Materials & Continua, vol. 67, no.1, pp. 1269–1285, 2021.

[24] A. Rahman, Sultan K., Aldhafferi N., Alqahtani A. (2018) Differential Evolution Assisted MUD for MC-CDMA Systems Using Non-Orthogonal Spreading Codes. In: Abraham A., Haqiq A., Muda A., Gandhi N. (eds) Innovations in Bio-Inspired Computing and Applications. IBICA 2017. Advances in Intelligent Systems and Computing, vol 735. Springer, Cham.

[25] A. Rahman, Qureshi I.M., Malik A.N., Naseem M.T., "QoS and Rate Enhancement in DVB-S2 using Fuzzy Rule Base System", Journal of Intelligent & Fuzzy Systems (JIFS), vol. 30 (1), pp. 801-810, 2016.

[26] A. Rahman, D.N. Zaidi, M.H. Salam, S. Jamil, "User Behavior Classification using Fuzzy Rule Based System", 13th International Conference on Hybrid Intelligent Systems (HIS'13), pp. 118-123, December 04-06, Tunisia, 2013.

[27] A. Rahman, S.A. Alrashed, A. Abraham, "User Behavior Classification and Prediction using FRBS and Linear Regression" Journal of Information Assurance and Security, vol. 12, no. 3, pp. 86-93, 2017.

[28] A. Rahman, I.M. Qureshi, A.N. Malik, M.T. Naseem, "A Real Time Adaptive Resource Allocation Scheme for OFDM Systems using GRBF-Neural Networks and Fuzzy Rule Base System", International Arab Journal of Information Technology (IAJIT), Vol. 11, no. 6, pp. 593-601, 2014.

[29] A. Rahman, I.M. Qureshi, A.N. Malik, M.T. Naseem, "Dynamic Resource allocation for OFDM Systems using DE and Fuzzy Rule Base System", Journal of Intelligent & Fuzzy Systems (JIFS), vol. 26 (4), pp. 2035-2046, 2014.

[30] A. Rahman, "Applications of Hybrid Intelligent Systems in Adaptive Communications", Modeling, Analysis and Applications of Nature-Inspired Metaheuristic Algorithms, Edition: 1st, Chapter: 10, pp. 183-217, Publisher: IGI Global, 2017.