IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 01 January 2024

Real Time Uni-Processor and Multi-Processor Scheduling Algorithms Survey

[1] Premkumar Balani, [2] Apurva Shah

[1] Department of Information Technology, GCET CVMU, Vallabh Vidyanagar, Anand, Gujarat, India
[2] Department of Computer Science and Engineering MSU, Vadodara, Gujarat, India
Corresponding Author Email: [1] pdsindhi@gmail.com, [2] Apurva.shah-cse@msubaroda.ac.in

Abstract*— Scheduling algorithms play a pivotal role in the effective functioning of real-time systems, and their diversity arises from the evolving needs and specifications of various ap- plications. The selection of a suitable algorithm is crucial in real- time environments and is significantly influenced by the system's characteristics, such as being preemptive or non-preemptive. Additionally, the choice is intricately linked to the system architecture, specifically the number of processors, whether it's a Uni-processor or a Multi-processor system. This paper aims to categorize scheduling algorithms in real-time systems into two primary classifications: Uni-processor scheduling algorithms and Multi-processor scheduling algorithms.

Index Terms*— Real-time Operating Systems (RTOS), Task Scheduling, Preemptive Scheduling, Non-preemptive Scheduling, Task Prioritization, Deadline-based Scheduling, Earliest Dead- line First (EDF), Rate Monotonic (RM) Scheduling, Processor Utilization.

I. INTRODUCTION

Real-time systems are characterized by their dependence not only on the logical outcomes of computations but also on the timely production of results. These systems find application in critical domains such as safety systems, power plant control units, satellite controllers, command systems, and flight control systems. Real-time systems are broadly classified into hard real-time systems and soft real-time systems. In hard real- time systems, responses must meet strict deadlines to prevent severe losses and potential dangers, as seen in applications like missile control systems. On the other hand, soft real-time systems, while valuing deadlines, can still operate effectively if deadlines are occasionally missed, as tasks can be rescheduled or completed after the specified time. Examples of soft real-time systems include multimedia and gaming systems [1]. This paper delves into the discussion of two widely used algorithms in real-time systems: Rate Monotonic and Earliest Deadline First algorithms. Additionally, it provides an exploration of various scheduling algorithms, presenting their respective advantages and disadvantages. The paper also investigates the suitability of these algorithms concerning the nature of the real-time system, such as whether it falls under the category of soft real time or hard real time. The subsequent sections of this paper are structured as follows. Section II presents a comprehensive review of relevant literature. In Section III, the focus is on scheduling algorithms tailored for Uni-processor systems, with a specific examination of

RM and EDF. Section IV provides an overarching view of scheduling algorithms designed for Multi-processor systems, accompanied by a comparative analysis between Uni-processor and Multi-processor scheduling algorithms. Finally, Section V encapsulates the concluding remarks and recommendations derived from the findings of this study.

II. A COMPREHENSIVE REVIEW OF RELEVANT LITERATURE

Numerous research endeavors have been dedicated to ex- ploring scheduling algorithms in real-time systems with the goal of identifying optimal solutions. This section encapsulates some of these scholarly contributions.

In reference to [2], the paper engages in a detailed exploration of real-time system scheduling algorithms, focusing on their impact on throughput capacity and response time. It delves into the features and performance metrics of real- time systems within various environments, categorizing them into Uni processor scheduling, Multi-processor scheduling, distributed scheduling, and specific algorithms like RMS, EDF, and LLF in Uni processors.

The work discussed in [3] provides a comprehensive overview of the real-time domain, emphasizing scheduling and operating systems. The paper categorizes scheduling models into static scheduling, preemptive scheduling with fixed priority, dynamic scheduling, and dynamic scheduling for best effort within real-time operating systems.

Within the context of [4], the authors concentrate on enhancing Earliest First Deadline (EDF) Algorithms to minimize relay tasks and predict their behavior. The paper introduces the Earliest First Deadline First (EFDF) algorithm, demonstrating reduced complexity through performance analysis, particularly in Multi-processor real-time systems.

[5] concludes that the EDF scheduling algorithm is optimal for Uni processors, albeit receiving limited attention from the industry. In contrast, Fixed Priority remains popular in commercial real-time operating systems, despite offering lower theoretical schedulable processor utilization.

In [6], the authors introduced LLREF, an optimal real-time

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 11 Issue 01 January 2024

scheduling algorithm for Multi-processors that does not rely on time quanta. This algorithm is designed based on the T- L plane abstraction technique, demonstrating that scheduling for Multi-processors can be conceptualized as recurring T-L planes. Effective scheduling on a single T-L plane leads to an optimal solution for all times.

Regarding [7], the authors emphasized the significance of the EDF algorithm in scheduling real-time tasks based on their deadlines. EDF is extensively studied as a dynamic priority-driven scheduling scheme due to its optimality for various types of tasks, including periodic, aperiodic, sporadic preemptive, sporadic non-preemptive, periodic non-preemptive tasks, and its ability to achieve high processor utilization for preemptive tasks. While finding an optimal schedule for periodic and aperiodic non-preemptive tasks is proven to be NP-hard, experiments in [9] demonstrate EDF's excellent performance under light system loads. However, in overloaded scenarios, EDF tends to perform poorly, prompting researchers to propose adaptive techniques for handling such situations.

The role of scheduling algorithms is highlighted in environments with a high number of users compared to available resources, as discussed in [10]. Various constraints, such as user or process priority and power consumption, must be addressed. These challenges are magnified in cloud environments [21] [22], where multiple resources need to be optimally utilized by numerous users. The mobile cloud computing paradigm, including layers like the Internet of Things (IoT), mist computing, edge computing, and fog computing [23]

[24] [25], further accentuates the need for optimal resource allocation to enhance quality of service and maximize benefits while ensuring user satisfaction [26] [27] [28].

III. SCHEDULING ALGORITHMS TAILORED FOR UNI-PROCESSOR SYSTEMS

Real-time systems utilizing a Uni processor employ various scheduling algorithms, as illustrated in Fig. 1. These algorithms fall into two main categories: static and priority-driven. In the static category, algorithms like Round Robin (RR) evenly allocate processor time among tasks. The focus of this section is on priority-driven algorithms, particularly the widely used ones, EDF and RM.

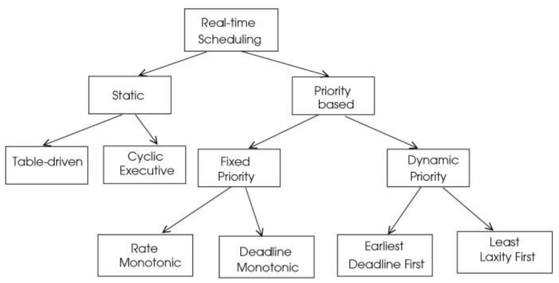

Fig. 1. Classification of Scheduling Algorithms for Uni-processor [11]

A. Priority-driven Scheduling Algorithms

As depicted in Fig. 1, priority-driven scheduling algorithms are further classified into fixed and dynamic based on priority assignment stability. This section provides an overview of two prominent priority-driven algorithms in real-time systems: EDF and RM.

The Rate Monotonic Algorithm, also known as the RM Scheduling Algorithm, operates with a fixed or static priority system based on task periods. Despite its drawback of not yielding optimal results in low-load situations, RM outperforms dynamic scheduling when the system is overloaded. In RM, tasks with shorter periods receive higher execution priority [12].

The Earliest Deadline First (EDF) Scheduling Algorithm, a dynamic scheduling approach, prioritizes tasks based on their deadlines. The task with the earliest deadline is assigned the highest priority. EDF Scheduling achieves 100 percent task utilization under loaded conditions or when utilization is less than or equal to 1. However, under slightly overloaded conditions, processor utilization decreases exponentially when task utilization exceeds the cross-load factor [12]. Table I outlines the advantages and disadvantages of RM and EDF.

Table 1: RM and EDF Advantages and Disadvantages

Algorithms	**Advantages**	**Disadvantages**
Rate Monotonic (RM)	*Good performance in overload condition*	*Poor CPU Utilization*
Earliest Deadline First (EDF)	*Optimum Performance in load* <= 1	*Poor performance in load* > 1

Case Study: RM and EDF Behavior on the Same Task Set Consider a task set with three tasks, each represented by its computation time (Ci) and period (Pi): T1(2,6), T2(3,8), T3(2,12). In Fig. 2(a), RM assigns priority based on the period, while in Fig. 2(b), EDF prioritizes based on task deadlines.

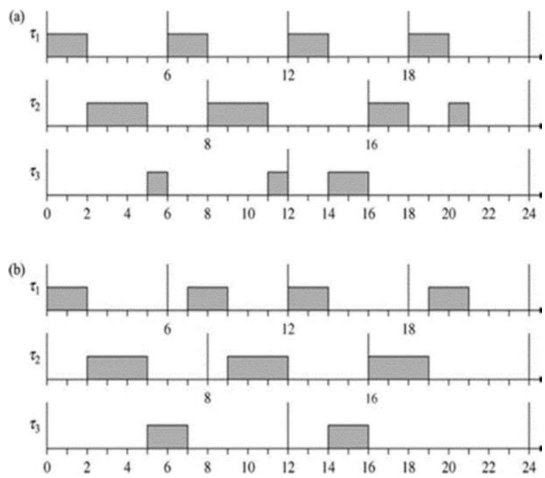

Fig. 2. RM and EDF scheduling for the Same Task Set

Case Study: RM and EDF Behavior on the Same Task Set

For a task set with tasks T1(2,5) and T2(4,7), RM may cause deadline misses due to preemptions, as illustrated in Fig. 3(a). In contrast, EDF, shown in Fig. 3(b), successfully schedules the task set without causing deadline misses. According to [13], ”For larger task sets, the number of preemptions caused

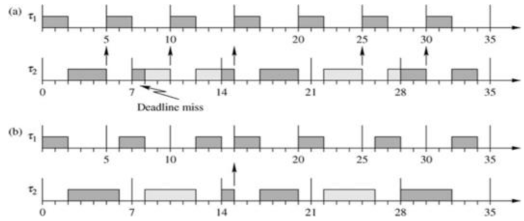

Fig. 3. RM under performance in Certain Conditions

by RM increases, thus the overhead due to the context switch time is higher under RM than EDF.”

IV. SCHEDULING ALGORITHMS DESIGNED FOR MULTI-PROCESSOR SYSTEMS

Over time, the demand for multiple processors has increased to handle more intricate and resource-intensive computations. Multi-processor systems necessitate distinct scheduling approaches compared to Uni-processor systems, leading to extensive research in the pursuit of optimal scheduling algorithms. As depicted in Fig. 4, Multi-processor system algorithms are categorized into classic, heuristic, and evolutionary types.

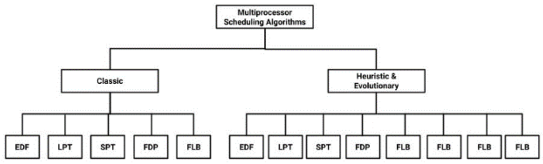

Fig. 4. Multiporcessor Scheduling Algorithms

Classic algorithms, despite not being exclusively designed for Multi-processor environments, exhibit lower time complexity in such systems compared to other categories. However, they often fall short of guaranteeing optimal solutions. On the other hand, heuristic and evolutionary algorithms strive for near-optimal solutions but typically incur higher running times.

In Table I and Table II, [15] provides a comprehensive com- parison of various Uni-processor and Multi-processor scheduling algorithms across different metrics, including priority, CPU utilization, context switching frequency, optimality, deadline miss probabilities, response time, predictability, effectiveness, and limitations.

In conclusion, the Instantaneous Utilization Factor (IUF) scheduling algorithm outperforms other Uni-processor algorithms, while the Modified Instantaneous Utilization Factor (MIUF) demonstrates superior response time, CPU utilization, and context switching compared to other Multi-processor algorithms [29] [30].

Table 2: Uni-processor Algorithms Comparison

Parameter	**EDF**	**LLF**	**MUF**	**IUF**
Processor Utilization	High	High	High	High
Context Switches	Less	High	High	High
Optimal	Yes	Yes	In Critical Tasks	Yes
Missing Deadlines	Average	Average	Less	Less
Response Time	High	Average	Low	High
Predictability	No	No	In Transient Load	Dynamic
Effectiveness	Optimal and Easy to Implement	Considers remaining execution time	Performs good in transient overload	Improves utilization bound of schedule
Setbacks	Poor Per- formance in overload	High context switches	Non critical task may miss deadline	Very High Context switches

Table 3: Multi-processor Algorithms Comparison

Parameter	**EDZL**	**ILLF**	**MMUF**	**MIUF**
Processor Utilization	High	High	High	High
Context Switches	Very Less	Less	Less	Less
Optimal	No	Yes	Yes	Yes

Missing Deadlines	Less	Less	Less	Very Less
Response Time	Low	High	Average	Low
Predictability	Better than EDF	Yes	In Transient Load	Dynamic
Miscellaneous	Low context switches	Low context switching	Performs Optimal in noncritical tasks	Better CPU Utilization
Setbacks	Deadline miss of critical tasks	Higher Execution time	Static Utilization of task sets	

V. CONCLUSION AND RECOMMENDATION

This paper presents a comprehensive examination of scheduling algorithms within the realm of real-time systems. A thorough review of prior studies in the field of real-time scheduling algorithms is provided. The focus extends to a detailed discussion of widely utilized Uni-processor algorithms, specifically Rate Monotonic (RM) and Earliest Deadline First (EDF). Furthermore, the paper offers an overview of Multi- processor scheduling algorithms.

In conclusion, the selection of a scheduling algorithm is influenced by numerous factors, and there is no one-size- fits-all optimal solution, given the diverse structures and requirements of different systems. As a potential avenue for future research, the application of evolutionary algorithms in scheduling presents an intriguing area that warrants further exploration and investigation.

REFERENCES

[1] G. Bernat, A. Burns, and A. Llamos´ı, "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 Interna- tional Conference on Computer and Communication Technologies in Agriculture Engineering, 2010, vol. 1, pp. 333–336. doi: 10.1109/CC- TAE. 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 Multi-processors," 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, "Com- puters and Intractability: A Guide to the Theory o[NPCompleteness."

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

[10] L. Georges, P. Mu¨hlethaler, 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]. Avail- able: www.ijsdr.org49

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

[14] M. Rouhifar and R. Ravanmehr, "A Survey on Scheduling Ap- proaches for Hard Real-Time Systems," International Journal of Com- puter 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 com- munication in DVB-S2." J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-02174-w.

[17] A. Rahman, S. Dash, and 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 Watermark- ing", 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

softwaredefined network for cognitive routing optimization using deep extreme learning machine approach," Computers, Materials and Continua, vol. 67, no.1, pp. 1269–1285, 2021.

[24] A. Rahman, Sultan K., Aldhafferi N., Alqahtani A. (2018) Differen- tial 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 Applica- tions. 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 and Fuzzy Systems (JIFS), vol. 30 (1), pp. 801-810, 2016.

[26] A. Rahman, D.N. Zaidi, M.H. Salam, S. Jamil, "User Behavior Classifi- cation 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 Informa- tion 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 Re- source allocation for OFDM Systems using DE and Fuzzy Rule Base System", Journal of Intelligent and 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.