Periodic Task Scheduling in Soft Real Time System using Multiple Linear Regression

Rohit Topiya, Prof. Jay Teraiya, Dr. Apurva Shah

Abstract— Real-time systems play vital role in our modern society. They are used in critical systems that working on timely response and prefect result. In this paper, the multiple linear regression – machine learning method for the real time periodic task scheduling has been proposed. During the research work, results are gathered with different periodic process sets. Results are analyzed with respect to Success Ratio (SR) and Effective CPU Utilization (ECU). Regression – Machine Learning based scheduling has compared with Least Slack Time (LST) and Shortest Job First (SJF) with performance parameter SR and ECU. LST is dynamic scheduling algorithm and it performs better in case of under lead situation. It is observed that regression scheduling method is almost equal in under load situation. We have taken different process sets with lead from 0.5 to 5 and contains processes from 1 to 10 for results. We have implemented one tool to generate such kind of dataset with different values of process sets, processes in set & load that can be useful for research work. We have made it also available online for others.

Index Terms— Linear Regression, LST(Least Slack Time), Machine Learning, Periodic Task, Scheduling, SJF(Shortest Job First), Soft Real Time System

1 Introduction

Real time systems are those which must produce the error-less response within the finite time limit. A real time system is a system where the preciseness of system function depends not only on accurate results, but also on the time when these results are generated. For Example, Air-Plane Control System, Nuclear Plant Control System, Anti-lock brakes etc....

A scheduling algorithm is a set of rules that determine the task to be performed at a particular moment.

Real Time Scheduling algorithms are divided into two categories static and dynamic. In case of static algorithm its applied priorities at initial (design) level and remain same throughout the task. In case of dynamic algorithm based on task priority is assigned dynamically runtime.

We can make dynamic scheduling with static priority where Rate Monotonic (RM) and Deadline Monotonic (DM) are two examples of it and Earliest Deadline First (EDF) and Least laxity (Slack) time First (LLF) are dynamic scheduling with dynamic priority.

1.1 Static Scheduling Algorithm

Rate Monotonic Algorithm (RM)

The Rate Monotonic (RM) scheduling algorithm assigns priorities to tasks according to their periods / request rates. Specifically, tasks with shorter periods will have higher priorities. Priorities assigned to processes are inversely proportional to the length of the period. RM is a fixed priority assignment because periods are constant: a priority Pi is assigned to the task before execution and does not change over time. RM is pre-emptive: the currently running task is pre-empted by a newly arrived task with shorter period.

In the case of the rate-monotonic scheduling algorithm, following are the constraints upon the process system. These include [6]:

1. Fixed set of processes;
2. All processes are periodic;
3. All processes have deadline equal to period;
4. One instance of a process must be complete before subsequent instances are run;
5. All processes have known worst-case execution times;
6. No synchronization is permitted between processes;
7. All processes have an initial release at time 0.

In 1973, Liu and Layland [14] showed that RM is optimal among all fixed-priority assignments in the terms of that no other fixed-priority algorithms can schedule a task set that cannot be scheduled by RM. Liu and Layland also derived the least upper bound of the processor utilization factor for a generic set of n periodic tasks [14].

Deadline Monotonic Algorithm (DM)

The Deadline Monotonic algorithm is an optimal static priority algorithm with preemption. It schedules periodic tasks with deadline less than or equal to period. Each task is assigned a fixed priority inversely proportional to relative deadline. This implies that a task with a shorter relative deadline has a higher priority. The rate monotonic algorithm is a special case of the deadline monotonic approach when deadline is equal to the period.

1.2 Dynamic Scheduling Algorithm

Earliest Deadline First Algorithm (EDF)

The Earliest Deadline First (EDF) algorithm is a dynamic scheduling algorithm that selects tasks according to their absolute deadlines. Specifically, tasks which have earlier deadlines will be executed at higher priorities.

EDF is a dynamic priority assignment. Moreover, it is typically executed in preemptive mode, thus the currently executing task is preempted whenever another periodic instance with earlier deadline becomes active.

Note that EDF doesn't build any specific assumption on the periodicity of the tasks; therefore, it can be used for scheduling periodic as well as aperiodic tasks.

A set of periodic tasks is schedulable with EDF algorithm if and only if [2].

$$U_n \leq 1$$

Least slack time (LST) scheduling

This scheduling algorithm assigns highest priority to processes that have the least "slack time". Slack time is defined as the difference between the deadline, the elapsed time and the remaining execution time.

To modify the running headings, select View | Header and Footer. Click inside the text box to type the name of the journal the article is being submitted to and the manuscript identification number. Click the forward arrow in the pop-up tool bar to modify the header or footer on subsequent pages.

IJSER staff will edit and complete the final formatting of your paper.

2 Background Work

A set of rules that determine the task to be performed at a particular moment.

Static Algorithms / off line Algorithms
Applied priorities at initial level & remain same till complete life cycle

- Rate Monotonic (RM)
- Deadline Monotonic (DM)

Dynamic Algorithms / Online Algorithms
Priority assign on runtime which depended on parameters of task & it may change in future depending upon situation of given system.

- Earliest Deadline First (EDF)
- Least Slack Time (LST)

Dynamic Scheduling algorithms performs better in under load situations while Static Scheduling Algorithms performs better in over load situations [2].

There are two major challenges:

- Standard Dataset to check algorithm
- Algorithm that can perform better in both situation under load & over load

2.1 Machine Learning

An application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Unsupervised Learning

Unsupervised learning is training of machine using non-categorized information & allowing machine to generate output for this information without any supervision. Unsupervised learning is to categories information based on their patterns, similarities & differences.

For Example, one image contains multiple shapes like triangles, circles, squares. Machine has no idea about shapes. But now machine can group shapes using their similarities & patterns like all squares.

Supervised Learning

Supervised learning is learning under observation. Supervised learning is a learning in which machine is trained using already well labeled data. Then machine is given new unlabeled data to produce result from labeled data.
Supervised learning is classified in two categories.

Classification

Classification problem is when output result is group or category. For example, given shape is square or not. Another example is given symptoms are of particular disease or not.

Regression

Regression is a parametric machine learning technique used to predict continuous (dependent variables) values from given a set of independent variables.

- Multiple benefits of using regression analysis. They are as follows:
- It indicates the sound relationships between dependent variable and independent variable.
- It indicates the strength of impact of multiple independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities. These benefits help us to eliminate and evaluate the best set of variables to be used for building predictive models.

Linear regression is used to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output).

Function for Linear Regression

$$Y = \beta 0 + \beta 1\, X \qquad (2)$$

X: Input data
Y: Output (labels to data)
$\beta 0$: Intercept
$\beta 1$: Coefficient of X

There are two types of Linear Regression

Simple Linear Regression

In simple linear regression when we have a one input variable, we can use statistics to get the coefficients.

Multiple Linear Regression

When we have more than one input variables, we can use multiple linear regression to get the values of the coefficients.

3 Proposed Algorithm

3.1 Multiple Linear Regression

When we want to predict value from multiple input variables then multiple regression is used. Here we have multiple input variables Period, Deadline & Execution Time.

So, Linear Regression Function will be as follows:

$$Y = \beta 0 + T\,\beta 1 + C\,\beta 2 \quad (3)$$

T: Period
C: Execution Time
Y: Output
β0: Intercept
β1: Coefficient of T (Period)
β2: Coefficient of C (Execution Time)

3.2 Algorithm

Input: Process Set
Output: Process to Execute

1. Enter Process set
2 .Enter Coefficients
a. β0 Regression Coefficient
b. β1 Period Coefficient
c. β2 Execution Time Coefficient
3. Declare Two Variables:
a. MinY
b. SelectedProcessNumber
4. for each process in process set
5. Extract Period(T) & Execution Time(C) from Process
6. Calculate Y = β0 + T β1 + C β2 for selected process
7. If first process
8. Set MinY = Y & SelectedProcessNumber = ProcessNumber
9. else if second process then check if Y < MinY
10. If if Y < MinY
11. SET MinY = Y & SelectedProcessNumber = ProcessNumber
12.end for
13.return SelectedProcessNumber

4 Dataset & Evaluation Environment

4.1 Dataset Generation

We have analyzed during our survey that there is no source available for real time process dataset. So, we have developed one data set generator tool to generate periodic task with different combination of No. of Process, No. of Process Set & Load (CPU Utilization). We have also made it online freely available (www.processdataset.com). It is effort to help researchers to get such dataset easily.

The Following parameters need to be entered to generate dataset.

1. Load (CPU Execution Time)
2. No. of Process in One Process Set
3. No. of Process Set

4.2 Processor Utilization Factor

We have used processor utilization factor to be the fraction of processor time spent in the execution of the task set. In other words, the utilization factor is equal to one minus the fraction of idle processor time. Since C/T is the fraction of processor time spent in executing task t, for n tasks, the utilization factor is mention in Equation 1:

Example of calculation of processor utilization factor. A real time system consists of three tasks T1, T2, and T3. Their characteristics have been shown in the following table.

Process (P)	Arrival Time	(C)	(D)	(T)
P1	00	10	20	20
P2	10	10	40	40
P3	30	10	50	50

$U = C1/T1 + C2/T2 + C3/T3$
$U = 10/20 + 10/40 + 10/50$
$U = 0.5 + 0.25 + 0.2$
$U = 0.95$

This means that 95 percent of the processor time is used to execute the periodic tasks, whereas the CPU is idle in the remaining 5 percent.

4.3 Under Load Training Data & Coefficients Derivation

We have generated thousands of processes with load < 1(Under Load) using our process generation tool with different combination of No. of Process, No. of Process Sets & load.

Then we have implemented Earliest Deadline First (EDF) algorithm using GCC Complier & C Language to Schedule generated processes over 500ms time. Rate Monotonic (RM) gives highest priority to process with nearest deadline.

Then we have calculated regression coefficient using our tool. Our tool reads time line data from excel & calculate regression coefficient & execution time coefficient. One Excel file contains 50 sheets, where each sheet contains 50 process sets plotted over 500 ms timeline. Then our tool calculates average of all sheets.

Hence, we have got coefficients for under load processes.

4.4 Under Load Testing Data & Coefficients Derivation

Then again, we have generated new processes with different combination of No. of Process, No. of Process Sets & load < 1 for testing under load coefficients.

Then we have scheduled testing data set using under load coefficients.

We have calculated y parameter of our equation for all processes & takes lowest value as first priority for scheduling. We have got good results.

4.5 Over Load Training Data & Coefficients Derivation

As like under load we have generated processes with load >= 1(Over Load) using our process generation tool with different combination of No. of Process, No. of Process Sets & load.

Then we have implemented Rate Monotonic (RM) algorithm using GCC Complier & C Language to schedule generated processes over 500ms time. Rate Monotonic (RM) gives highest priority to process with shortest period.

Then we have calculated regression coefficient using our tool. Our tool reads time line data from excel & calculate regression

coefficient & execution time coefficient. One Excel file contains 50 sheets, where each sheet contains 50 process sets plotted over 500ms timeline. Then our tool calculates average of all sheets.
Hence, we have got coefficients for over load processes.

4.6 Final Regression Coefficients

Finally, we have calculated average of under load & over load coefficients. We have calculated coefficients for 6,000+ sheets which contains 3,00,000+ processes.

Table – I Regression Coefficients

Coefficient	Value
β_0 (Intercept)	0.7233841
β_1 (Period)	-0.070790128
β_2 (Deadline)	-0.013595236
β_3 (Execution. Time)	0.015169379

Based on above values we can understand that β0 (Intercept) & β3 (Execution. Time) these two coefficients have more impact on dependent value.
Therefore our equation (3) is

$$Y = \beta 0 + T\, \beta 1 + C\, \beta 2$$

5 Result & Analysis

SR (Success Ratio):

Main Key aspect of real time system is to achieve deadline. The most appropriate parameter for performance is the Success Ratio (SR). It is defined as:

ECU (Effective CPU Utilization):

It is important that how efficiently the scheduler utilizes the processes, especially in overload situation. Therefore, the performance parameter is Effective CPU Utilization (ECU). It is defined as:

$ECU = \sum_{(i \in R)} V_i / T$ [2](4)

Where,
V: The value of the task and
V = Execution Time of task, if the task completes within its deadline.
V = 0, it the task fails to meet the deadline.
R: A Set of tasks, which are scheduled successfully.
T: The Total time of scheduling

Table II
(LST, RM & ML Performance in under load)

Load	ECU %			SR %		
	LST	RM	ML	LST	RM	ML
0.50	49.99	49.99	49.99	100	100	100
0.55	54.40	54.40	54.40	100	100	100
0.60	59.39	59.39	59.39	100	100	100
0.65	64.35	64.35	64.35	100	100	100
0.70	69.35	69.35	69.35	100	100	100
0.75	74.31	74.31	74.31	100	100	100
0.80	79.22	79.22	79.22	100	100	100
0.85	84.16	84.16	84.16	100	100	100
0.90	89.16	89.16	89.16	99.99	99.99	99.99
0.95	94.17	94.08	94.08	99.93	99.99	99.93
1.00	99.10	97.78	99.10	98.92	100	98.88

Table III
(LST, RM & ML Performance in over load)

Load	ECU %			SR %		
	LST	RM	ML	LST	RM	ML
1.05	16.09	70.85	72.79	15.84	78.49	78.44
1.10	8.33	75.82	77.49	7.9	80.49	80.41
1.15	5.58	73.2	73.52	5.06	75.88	75.62
1.20	4.21	83.5	84.43	3.67	79.47	78.64
1.25	3.56	79.05	79.2	3.06	77.58	76
1.30	3.09	75.66	75.76	2.53	73.81	72.03
1.35	2.63	74.65	74.14	2.09	70.77	69.06
1.40	2.2	83.55	83.85	1.71	75.47	74.14
1.45	2.01	79.75	80.85	1.52	69.03	68.17
1.50	1.83	85.27	84.25	1.33	70.33	67.8
1.60	1.77	85.61	85.01	1.29	69.52	67.67
1.70	1.58	86.26	84.95	1.07	65.99	63.69
1.80	1.45	86.12	84.67	0.95	65.98	62.99
1.90	1.31	85.83	83.44	0.85	63.51	61.2
2.00	1.19	85.78	84.37	0.76	62.88	61.38
2.25	1.13	84.27	82.9	0.65	56.16	54.25
2.50	0.98	87.06	85.73	0.54	53.82	52.37
2.75	0.91	89.21	87.04	0.47	52.07	50.61
3.00	0.86	94.46	93.82	0.4	48.36	47.76
3.50	0.75	93.48	92.74	0.33	44.5	43.99
4.00	0.73	95.04	95.33	0.27	39.52	39.19
4.50	0.71	96.77	95.82	0.24	36.45	35.96
5.00	0.66	98.13	98.33	0.2	31.72	31.72

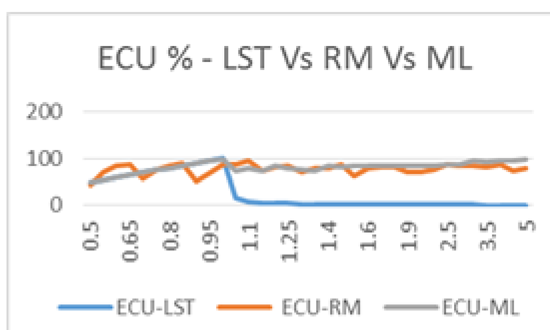

Fig. 1 ECU % - LST Vs RM Vs ML

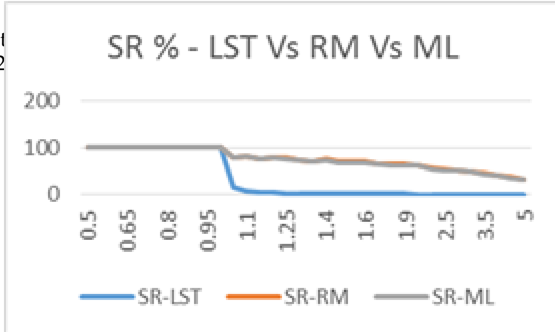

Fig. 2 SR % - LST Vs RM Vs ML

6 Conclusion

From Result Analysis we have concluded that regression machine learning is good approach for scheduling of periodic tasks in soft real time system.

Regression machine learning performs better than EDF in under load situations. In over load situations, it performs equivalent to RM.

It can be better by finding more parameters in future, which may perform better in overload situations than traditional algorithms.

Acknowledgment

We would like to acknowledge and give our sincere thanks to leading india ai group which is nationwide initiative on "AI and deep learning Skilling and Research". It is funded by Royal Academy of Engineering, UK under Newton Bhabha.

References

[1] J. Teraiya, A. Shah & K. Kotecha, "ACO Based Scheduling Method for Soft RTOS with Simulation and Mathematical Proofs", in International Journal of Innovative Technology and Exploring Engineering (IJITEE), Scopus Journal, Volume-8, Issue-12, October 2019.

[2] J. Teraiya, A. Shah, "Comparative Study of LST and SJF Scheduling Algorithm in Soft Real-Time System with its Implementation and Analysis", in International Conference on Advances in Computing, Communications and Informatics (ICACCI), Banglore, India, Proceeding in the IEEE Xplore, pp. 706-711, 2018.

[3] K. K. Baseer, Vikram Neerugatti, Sandhya Tatekalva, Akella Amarendra Babu, "Analysing various Regression Models for Data Processing" in International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8 Issue-8 June, 2019

[4] M. Arvindhan, Reetu Singh,Kajol Kathuria, "An Efficient Technique for Scheduling Algorithm In Real Time Environment For Optimizing the Operating System", International Journal of Computer Science & Wireless Security (IJCSWS), Vol. 02(01), pp.276-278, Jan-Feb 2018.

[5] Ki-Young Lee, Kyu-Ho Kim, Jeong-Jin Kang, Sung-Jai Choi,Yong-Soon Im, Young-Dae Lee, Yun-Sik Lim, "Comparison and Analysis of Linear Regression & Artificial Neural Network", in International Journal of Applied Engineering Research, Volume 12, Number 20, pp. 9820-9825, 2017

[6] Siddharth Dias, Sidharth Naik, Sreepraneeth K, Sumedha Raman, NamrathaM, "A Machine Learning Approach for Improving Process Scheduling : A Survey", in International Journal of Computer Trends and Technology (IJCTT), Volume 43, Number 1, January 2017.

[7] Wamika Basu & Subho Chaudhuri, "A Survey of Scheduling Algorithms in Real Time Systems", in International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Volume 6, Issue 2, February 2016.

[8] Abroyan, Narek & Hakobyan, Robert. (2016). A Review of the Usage of Machine Learning in Real-time Systems. Proceedings of NPUA.

[9] Reshu Manwani, Nisha Phogat, "Real-Time Scheduling of Soft Periodic Tasks on Multiprocessor Systems: A Fuzzy Model", in International Journal of Computer Science and Mobile Computing, Volume-3, Issue-6, pg. 348-355, June 2014.

[10] S. Eswaraiah Naidu, Abdul Rahim.B and Fahimuddin Shaik, "Non-Preemptive Real-Time Scheduling of Earliest Deadline Approach through Periodic and Sporadic Tasks" in Journal of Basic and Applied Engineering Research, Volume 1, Number 12, pp. 137-141; October-December 2014.

[11] Mr. Vishal Vora & Dr. Ajay Somkuwar, "Implementation & Performance Analysis of Real Time Scheduling Algorithms for Three Industrial Embedded Applications (IEA)" in International Journal of Information Technology Convergence and Services (IJITCS) Vol.2, No.6, December 2012.

[12] G.Saini, "Application of Fuzzy logic to Real-time scheduling", Real-Time Conference, 14th IEEE-NPSS. 2005.

[13] G. C. Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, 3rd ed., Springer, 2005.

[14] C. Liu and J. Layland, "Scheduling algorithm for Multiprogramming in a Hard Real-Time Environment", in Journal of the ACM, vol. 20, no. 1, pp.46-61, 1973.