International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-1 Issue-4, April 2012

Clustering-Based Deadline Monotonic Scheduling with Dynamic and Intelligent Time Slice for Real-time Systems

H. S. Behera, Sushree Sangita Panda, Jana Chakraborty

Abstract*: In this paper, clustering based deadline monotonic scheduling with dynamic and Intelligent Time Slice is proposed for real-time systems. The proposed algorithm determines the schedulability of a process, and groups the schedulable and unschedulable process into separate clusters. Then an improved Round Robin algorithm is used for all schedulable processes to calculate average turn-around time, average waiting time and context switch. Then the experimental analysis shows that the performance of the proposed algorithm is better in terms of average turnaround time, average waiting time and context switch.

***Keywords:* Real-time systems, Deadline monotonic scheduling(DM), Response Time Analysis Schedulability test, Clustering, Round Robin(RR) scheduling algorithm Turnaround time and Waiting time, Upper Quartile.**

I. INTRODUCTION

Real-time systems are divided into two types hard real time systems and soft real-time systems. The goal of a hard real-time system is to ensure that all deadlines are met, but for soft real-time systems the goal becomes meeting a certain subset of deadline in order to optimise some application specific criteria. A real-time system is a finite collection of recurrent processes each of which is independent of the other. Each process is characterised by an arrival time, an execution time and deadline and each process should complete its execution between its arrival time and deadline. The objective of cluster analysis is the classification of objects according to similarities among them and organising of data into groups. Clustering techniques are among the unsupervised methods. Different classification can be related to the algorithmic approach of the clustering techniques. Partitioning, hierarchial, graph, theoretical methods and methods based on objective function can be distinguished. Clustering techniques can be applied to data that is quantitative, qualitative or a mixture of both. Clustering is mainly used in data mining and statistical data analysis which includes machine learning, pattern recognition, image analysis, information retrieval and bioinformatics. The deadline monotonic scheduling algorithm is a priority driven schedulability algorithm that assigns priority to processes based upon their deadlines that is the task with shortest deadline is assigned highest priority. A process $\square_i$ is characterized by three parameters i.e. $\square_i = (C_i, D_i, T_i)$, where

T_i= time period for sequence of successive processes.

C_i= execution requirement and

D_i= deadline

A schedulability test for DM scheduling accepts as input the specifications of process and determines the schedulability of process. This paper outlines the determination of schedulability clusters of processes for real-time systems. In conjunction to this, schedulable processes are scheduled using improved RR algorithm with dynamic and intelligent time slice and performance metrics are obtained.

A. RELATED WORK

The priority assignment scheme that caters for processes with the relationship:

Computation time <= deadline <= timeperiod was defined by Leung et.al [10]. It proposed an algorithm for deadline monotonic scheduling in which priority assigned to processes are inversely proportional to the length of their deadline. To generate a schedulability constraint for deadline monotonic scheduling the behaviour of processes released at a critical instant is fundamental if all processes are proved to meet their deadlines during executions beginning at a critical instant these processes will always meet their deadlines [9]. The deadline monotonic approach for hard real-time systems was proposed by N. C. Audsley et.al [2] and [4]. According to [7] fixed priority scheduling with deadline prior to completion for real-time systems is considered. With reference to [8] misconception about real time computing which is a serious problem for next generation computers was considered. Paper [1] outlines how to determine which process in the process set is schedulable according

to the schedulability tests in deadline monotonic scheduling. In paper [2] the schedulability analysis of global deadline-monotonic scheduling is done.\

Manuscript published on 30 April 2012.
* Correspondence Author (s)
H.S. Behera*, Computer Sc. And Engineering, Veer Surendra Sai University of Technology, Burla, Sambalpur, India, (e-mail: hsbehera_india@yahoo.com).
Sushree Sangita Panda, Computer Science And Engineering, Veer Surendra Sai University of Technology, Burla, Sambalpur, India, 9437252608, (e-mail: sushreesangita@rocketmail.com).
Jana Chakraborty, Computer Science And Engineering, Veer Surendra Sai University of Technology, Burla, Sambalpur, India, 9439823779, (e-mail: jana.chakraborty@gmail.com).

II. DEADLINE MONOTONIC SCHEDULING

The deadline monotonic scheduling algorithm is a priority driven scheduling algorithm that assigns priority to tasks according to their deadlines: the smaller the deadline greater the priority.

The schedulable processes should have the following relationship:

Computation time <= deadline <= time period

For the ith process (if we have n processes then process 1 has the highest priority and process n has the lowest priority in the system).

Ci <= Di <= Ti

Deadline monotonic priority assignment is an optimal static priority scheme which implies that by using an algorithm deadline monotonic priority ordering for processes that will schedule the process set where process deadlines are unequal to their time period.

III. SCHEDULABILITY TEST

In Response Time Analysis schedulability test for a given process Ji, Here task i's worst-case response time, R, is calculated first and then checked (trivially) with its deadline $R_i \leq D_i$

Ri will be equal to the process's worst case execution time Ci plus all of the interference given by the execution of the higher priority processes.

$R_i = C_i + I_i$ ⟶ 1

Each release of process k will impose an interference of C_k. Hence maximum interference is imposed by process k on process i is given by:

$$I_i = \left\lceil \frac{R_i}{T_k} \right\rceil C_k \longrightarrow 2$$

Let $h_p(i)$ be the set of processes with higher priority.

$$I_i = \sum_{k \in hp(i)} \left\lceil \frac{R_i}{T_k} \right\rceil C_k \longrightarrow 3$$

From equation 1 and 3

$$R_i = C_i + \sum_{k \in hp(i)} \left\lceil \frac{R_i}{T_k} \right\rceil C_k$$

The response time equation can be solved by forming a recurrence relation

$$W_i^{n+1} = C_i + \sum_{k \in hp(i)} \left\lceil \frac{W_i^n}{T_k} \right\rceil C_k$$

- $W_i^0 = C_i$ is initialised
- The sequence of values $W_i^0, W_i^1, \ldots\ldots$ is monotonically non-decreasing
- When $W_i^n = W_i^{n+1}$, then the solution to the equation has been found

  Then the solution to the equation has been found.
  - If W_i^n exceeds T_i process will not meet its deadline.

IV. PROPOSED APPROACH

In the proposed algorithm Response Time Analysis schedulability test is implemented in clustering algorithm to determine the schedulability of any random process. Further they are clustered into schedulable and unschedulable clusters.

Finally the schedulable processes are scheduled using the improved Round Robin scheduling algorithm with dynamic and intelligent time slice. For the purpose of scheduling, in a single scheduling cycle of the processes sorted in ascending order of their Intelligent Time Slice, the time quantum is the intelligent time slice of the median of all the processes. The scheduling continues with this time quantum up to the medianth process. For the succeeding processes, the time quantum is the Intelligent Time Slice of the Upper Quartile of all the processes. Let the original time slice(OTS) is the time slice to be given to any process. Priority Component (PC) is assigned depending on the priority which is inversely proportional to the priority number, process having highest priority is assigned 1 and rest 0. Shortness Component (SC) is the difference between the burst time of current process and the previous process, if the difference is less than 0, then SC=1 else 0. For calculating Context Switch Component, first PC, SC and OTS are added and then subtracted from the burst time, if difference is less than OTS, then it will be considered as CSC.

ITS=OTS + PC + SC + CSC.

The ITS of all the processes are sorted in ascending order. The median of all the ITS is calculated and taken as the time quantum till the median position of the processes. In the next step the ITS of the rest processes are calculated by applying the Upper Quartile method which is not more than the maximum burst time. The process is continued till all the processes are scheduled and removed from the ready queue.

A. *Pseudo code of clustering based deadline monotonic scheduling with improved RR scheduling.*

Let n: no. of processes.
C_i: execution time for i^{th} process.
D_i: deadline for process i.
R_i: response time
ITS: Intelligent Time Slice.
OTS: Original Time Slice.
PN: priority number.
PC: priority component.
CSC: context switch component.
SC: shortness component.
MTQ: median(burst time of all the processes in ready queue)
UTQ: upper quartile(burst time of all the remaining processes in the ready queue after calculating the median)

1) Generate random numbers = computation time,C_i .
2) Divide those processes into 2 clusters randomly.
3) for(j = 1; j <= 100; j++)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

{Obtain the mean of each cluster

$$\text{mean} = \frac{\sum \text{computation time of all processes}}{\text{no. of processes}}$$

Deadline, D_i = mean + C_i}

4) for each process
5) Calculate response time:

$$R_i = C_i + \sum_{k \in hp(i)} \left\lceil \frac{R_i}{T_k} \right\rceil C_k$$

6) if ((R_i == R_{i-1}) && (R_i <= D_i))
Display ("process schedulable") and cluster them.
7) else
Display("process unschedulable") and from another cluster
8) exit
9) if(burst time of current process-burst time of previous process < 0)
SC=1;
10) else SC=0;
endif
11) if(burst time - (PC + SC + OTS) < 0)
CSC=1;
12) else CSC=0;
endif
13) ITS = OTS + PC + SC + CSC
14) if(ITS<ITS of medianth process)
TQ = ITS of MTQ
15) else
//assign cpu to process and give it time of slice = UTQ
Assign UTQ to process
TQ = ITS of UTQ
endif

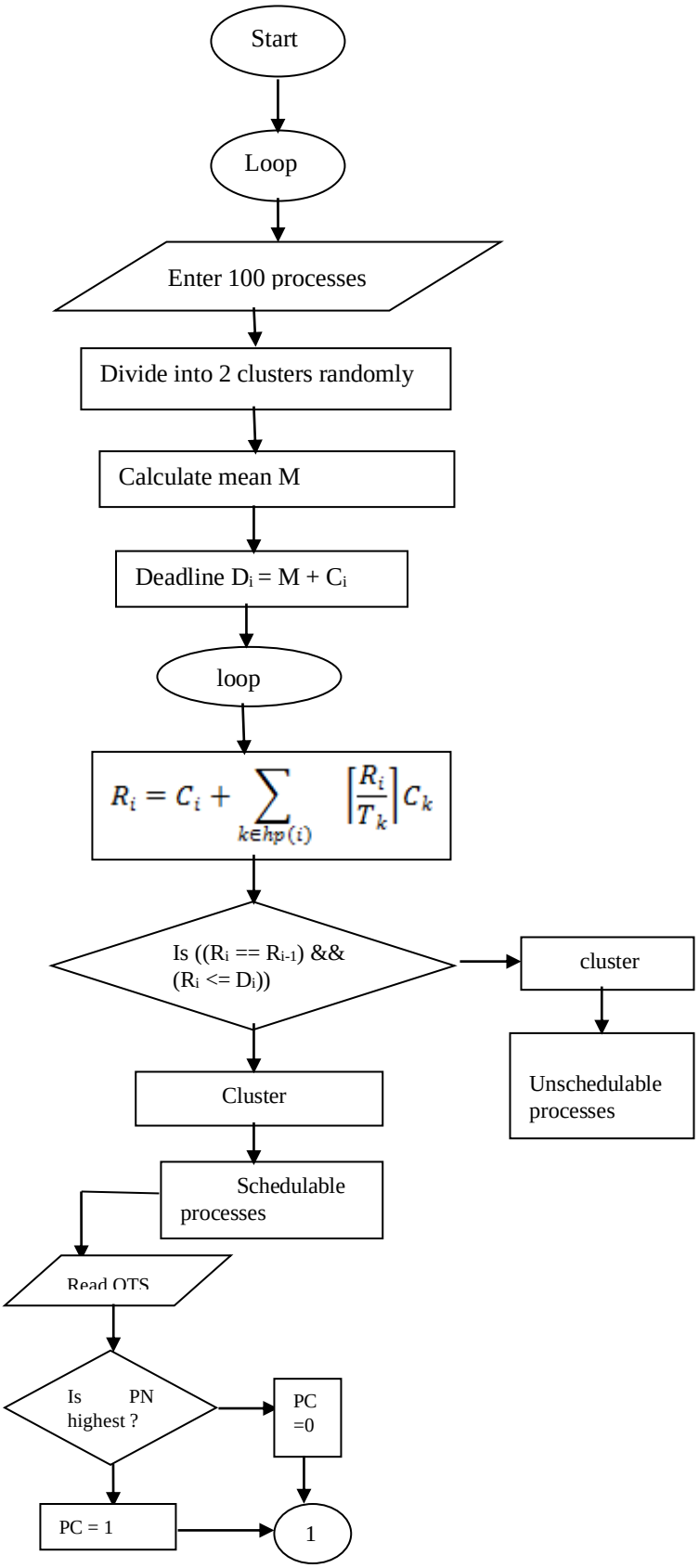
16) Average TAT:

$$A_{TAT} = \sum_{i=1}^{N} \frac{TAT\ of\ all\ the\ processes}{no.of\ processes}$$

17) Average WT:

$$A_{WT} = \sum_{i=1}^{N} \frac{WT\ of\ all\ the\ processes}{no.of\ processes}$$

Fig-1: pseudo code for schedulability and clustering of schedulable and unschedulable processes in conjunction with improved Round Robin algorithm with dynamic and Intelligent Time Slice



Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

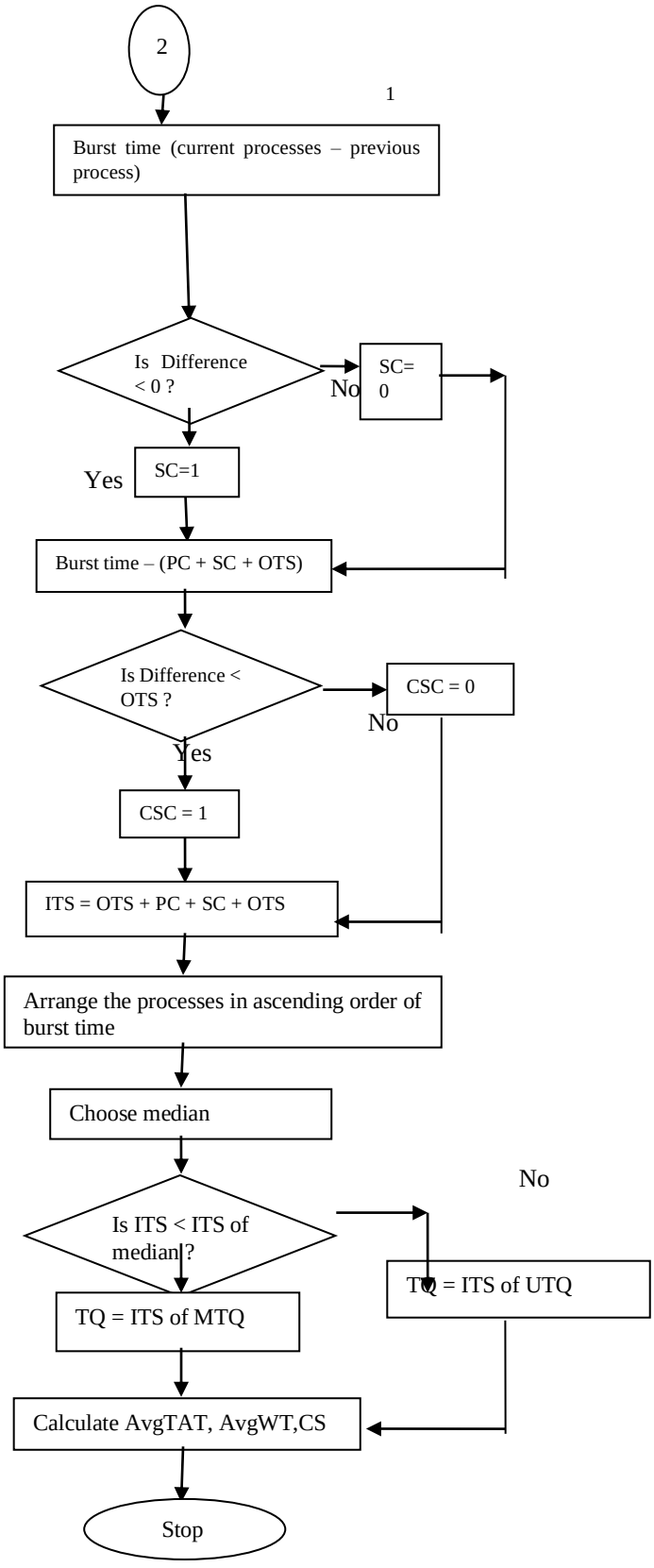


Fig-2: flowchart of the proposed algorithm.

V. ILLUSTRATION

A set of 100 jobs in a process set along with Ci , Ti and Di that is, computation time , time period and deadline respectively in a real-time system is taken. Then each job is clustered taking into account the worst-case response time and checked with its deadline. Ri = Ci + Ii

where Ii is the interference to a job due to execution of higher priority jobs. For a process to be clustered into schedulable cluster, Ri must be less than or equal to its deadline, Ri <= Di.

Schedulable jobs are put in a ready queue and taking a time quantum and the arrival time 0 the Round Robin scheduling is performed with Intelligent Time Slice and Upper Quartile method. Original time slice is taken as 10. The PC,SC,.CSC is calculated according to the algorithm. The ITS of all the processes are arranged in ascending order initially. Then the ITS of the process in the median position is taken as the time quantum for the processes till the medianth process. For the remaining processes, the ITS of the last process is taken as time quantum by applying Upper Quartile method.

VI. EXPERIMENTAL ANALYSIS

A.Assumptions

All the processes were processed in the real-time systems with single processor environment and all the processes are independent of each other. The time period is more than the deadline and deadline is greater than the computation time. All the attributes like burst time, numbers of processes, priority, Intelligent Time Slice are known before submitting the processes to the processor. The arrival time is assumed to be 0.

B.Experimental Frame Work

The experiment consists of several input and output parameters. The input parameters consist of deadline, computation time, time period, burst time, time quantum, priority and number of processes. The output parameters consist of a cluster of schedulable processes and another of unschedulable processes, interference on a process by other higher priority processes, average waiting time, average turnaround time and number of context switches.

C.Data Set

Several experiments have been performed in order to evaluate the performance of the proposed algorithm. A set of random numbers are generated to get the desired cluster. The data set are the processes with burst time increasing, decreasing and random to perform scheduling.

D.Performance Metrics

For the experimental analysis the significance of performance metrics is as follows:

1. Turnaround time(TAT):for the better performance of the algorithm, average turnaround time should be less.
2. Waiting time(WT): for the better performance of the algorithm, average waiting time should be less.
3. Number of Context Switches (CS): for the better performance of the algorithm, number of context switches should be less.

E.Experiments Performed

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Quartile of processes in the process set. To evaluate the performance of the proposed algorithm, a set of 100 processes was taken. In this the schedulability of processes is determined with Response Time Analysis schedulability test and the processes are assigned clusters accordingly using their density factor. After finding the jobs that are schedulable improved Round Robin scheduling algorithm was applied where the arrival time of processes is 0 and time quantum is the Intelligent Time Slice of the Upper

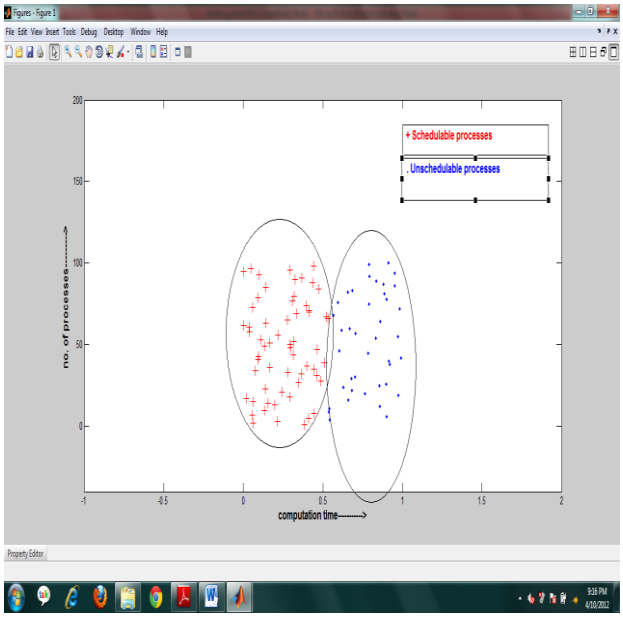


Fig-3: Clustering of jobs in the process set to distinguish between schedulable and unschedulable processes.

After clustering the processes according to the proposed algorithm a set of 7 processes is taken for simplicity and the average Turn Around Time, average Waiting Time and Context Switch component is computed for those 7 processes.

Case 1: 7 processes are taken with arrival time= 0, and increasing burst time and priority.

JOBS	BURST TIME	PRIORITY NO.	PC	SC	CSC	ITS
J_1	18	1	1	0	1	12
J_2	24	2	0	0	0	10
J_3	25	1	1	0	0	11
J_4	30	3	0	0	0	10
J_5	36	4	0	0	0	10
J_6	43	1	1	0	0	11
J_7	45	5	0	0	0	10

Table 1(a): it is the output using the proposed scheduling algorithm for increasing burst time

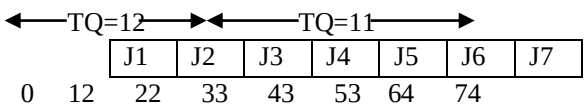


Fig-4: Gantt chart for algorithm as proposed in table 1(a).

ALGORITHM	AVG TAT.	AVG WT	CS
Proposed algorithm.	43	32.4	6

Table 1(b): performance metrics for increasing burst time.

Case 2: 7 processes are taken with arrival time=0, and decreasing burst time and priority.

JOBS	BURST TIME	PRIORITY NO.	PC	SC	CSC	ITS
J1	45	5	0	0	0	10
J2	43	1	1	1	0	12
J3	36	4	0	1	0	11
J4	30	3	0	1	0	11
J5	25	1	1	1	0	12
J6	24	2	0	1	0	11
J7	18	1	1	1	1	13

Table 2(a):output using the proposed scheduling algorithm for decreasing burst time.

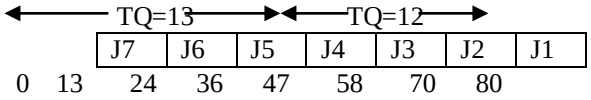


Fig-5: Gantt chart for algorithm as proposed in table 2(a).

ALGORITHM	AVG TAT.	AVG WT	CS
Proposed Algorithm.	46.8	35	6

Table 2(b): performance metrics for decreasing burst time

Case 3: 7 processes are taken with arrival time=0, and random burst time and priority.

JOBS	BURST TIME	PRIORITY NO.	PC	SC	CSC	ITS
J1	30	3	0	0	0	10
J2	25	1	1	1	0	12
J3	36	4	0	0	0	10
J4	43	1	1	0	0	11
J5	45	5	0	0	0	10
J6	18	1	1	1	1	13
J7	24	2	0	0	0	10

Table 3(a): it is the output using the proposed scheduling algorithm for random burst time.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

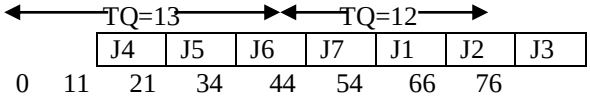


Fig-6: Gantt chart for algorithm as proposed in table 4(a)

ALGORITHM	AVG TAT.	AVG WT	CS
Proposed Algorithm.	43.7	32	6

Table 3(b): performance metrics for random burst time.

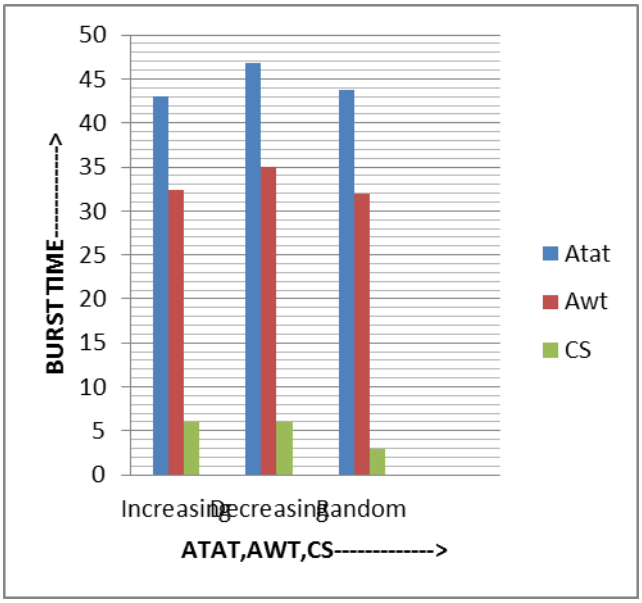


Fig- average turnaround time, average waiting time and context switch for increasing, decreasing and random burst time.

VII. CONCLUSION AND FUTURE WORK

An analysis of clustering in deadline monotonic scheduling algorithm using dynamic and intelligent time slice is made.

In this paper, a certain schedulability test has been implemented for clustering. The schedulable and unschedulable processes are then clustered. Schedulable processes were scheduled using improved Round Robin algorithm. All the processes were assumed to have a critical instant. This is ensured as all the processes have an initial release at time 0. An exact value of Ii must therefore be known, so that exact interleaving of higher priority processes is known. Finally the improved Round Robin scheduling algorithm with dynamic and Intelligent Time Slice was applied to schedulable processes. The experimental analysis shows that the proposed algorithm has better performance in terms of average turnaround time, average waiting time and context switch.

REFERENCES

[1]. H.S. Behera, Jana Chakraborty, Sushree Sangita Panda "Improved Deadline Monotonic Scheduling with dynamic and Intelligent Time Slice for real-time systems" department of computer science and engineering, V.S.S.U.T, Burla.

[2]. N.C. Audsley , A. Burns , M.F. Richardson , A.J. Wellings "Hard Real-Time Scheduling: The Deadline-Monotonic Approach" Dept of Computer Science, University of York, York,YO1 5DD, England.

[3]. Yaashuwanth .C & R. Ramesh "Intelligent Time Slice For Round Robin In Real Time Operating Systems" Dept of electrical and electronics engineering, Anna University Chennai, Chennai 600 025

[4]. Audsley, N.C. (1990), Deadline Monotonic Scheduling, YCS 146, *Dept. of Comp. Sci., Univ. of York.*

[5]. Barbara Korousic –Seljak (1994) "Task scheduling policies for real-time Systems" Journal on MICROPROCESSOR AND MICROSYSTEMS, VOL 18 NO. 9, pg501-512.

[6]. Bur89a. A. Burns and A. J. Wellings, Real time Systems And Their Programming Languages, Addison Wesley(1989).

[7]. Burns. A (1994) "Fixed priority scheduling with deadline prior to completion" Real time systems Research Group Department of computer science university of york, UK.

[8]. Sta88a. J. A. Stankovic, Misconceptions About Real Time Computing: A serious problem for next generation systems, IEEE Computer21(10), pp. 10-19(Oct 1988).

[9]. Liu, C.L. and J. W. Layland (1973) scheduling algorithms for multiprogramming in a hard real time environment J. ACM, 20, pp.40-61

[10]. Leung, J., and Whitehead, J. On the complexity of fixed priority scheduling of periodic, real time tasks.

AUTHOR PROFILE

Dr. H. S. Behera is currently working as a faculty in Dept. of Computer Science and Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Orissa, India. His areas of interest include Distributed Systems, Data Mining and Soft Computing.

Jana Chakraborty is a final year B.Tech student in Dept. of Computer Science and Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Orissa, India.

Sushree Sangita Panda is a final year B.Tech student in Dept. of Computer Science and Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Orissa, India.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)