Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.230 – 241

RESEARCH ARTICLE

Performance Enhancing in Real Time Operating System by Using HYBRID Algorithm

Harshal Garodi[#1], Kiran More[#2], Nikhil Jagtap[#3], Suraj Chavhan[#4], Prof.Chandu Vaidya[#5]
[1,2,3,4] Department of CSE, Rajiv Gandhi College of Engineering & Research, R.T.M.N.U, Nagpur, India
[5] Lecturer, Department of CSE, Rajiv Gandhi College of Engineering & Research, R.T.M.N.U, Nagpur, India
[1] harshalhrg93@gmail.com; [2] kirankmore7777@gmail.com; [3] nikhiljagtap1190@gmail.com;
[4] chavhansuraj06@gmail.com; [5] chandu.nyss@gmail.com

Abstract— The Real Time Operating System (RTOS) supports applications that meet deadlines, in addition to provides logically correct it's outcome. In multitasking operating system for the applications request to meeting of time deadlines and functioning in proper real time constraints, To meet the real-time constraints ; in Real time system for scheduling the task, different scheduling algorithms used. Most of the real-time systems are designed using priority based preemptive scheduling algorithm, worst case execution time estimates to guarantee the execution of high priority tasks.
The hybrid Algorithm schedules the process on single processor when it is preemptive. The advantage of the proposed algorithm is that it automatically switches between the EDF and ACO scheduling algorithm and overcome the limitation of both the previously discussed algorithms in paper.

Keywords—Earliest Deadline First (EDF), Ant Colony Optimization (ACO), Real-Time Operating System (RTOS), Fixed Priority (FP), Rate Monotonic (RM).

I. INTRODUCTION

Real-time systems [1][2] are playing a crucial role in our society. The applications of real-time systems being such as chemical industries, space missions, flight control systems, military systems, and so on. All make use of real-time technologies. In other words, real-time systems have timing requirements that must be guaranteed. The most common dynamic priority scheduling algorithm for real-time systems is the Earliest Deadline First(EDF) which was introduced by Liu and Leyland [3] in 1973. According to the EDF algorithm, an arrived job with the earliest deadline is executed first. The EDF algorithm has been proven by Dertouzos [4] "Procedural Control of Physical Processes" To be optimal among all scheduling algorithms on a uniprocessor system. In real time system sense that if a real-time task cannot be scheduled by EDF, then this task set cannot be scheduled by any algorithm. C.L. Liu and J.W. Layland, Presented a necessary and sufficient schedulability condition for EDF systems under the assumption that all task's relative deadlines are equal to their periods.
The schedulability condition is that the total utilization of the task set must be less than or equal to 1 [5].

A) AIM AND OBJECTIVES:

The aim of implementing a Scheduling algorithm which is the fusion of Earliest Deadline First (EDF) and Ant Colony Optimization (ACO) are stated below .The EDF algorithm places the process in a priority queue and run the task by using its deadline. The priority of the processes depends upon its deadline and handles the under-loaded condition [6], [7]. The limitation of EDF algorithm is that it cannot handle the overloaded condition.The execution of ACO algorithm is based on the execution time.which process has less “Execution Time” is executed first. The limitation for executing process in ACO scheduling algorithm it takes more execution time than the EDF scheduling algorithm.For that, to remove the drawbacks of EDF and ACO we propose new algorithm that will remove the drawback of the above discussed algorithm.
Hybrid algorithm will enhance the performance of the system and reduces the system failure ratio. Also the percentage of missing deadline is decreased. The main advantage of the Hybrid scheduling algorithm is, it will handles over-loaded and under-loaded condition simultaneously.

B) RELATED WORK

1) Brief Information about Real-time Systems (RTOS)

Real-time systems deliver services while meets timing constraints not necessarily fast, but must meet its deadline. A real-time system requires that results be produced within a specified deadline period .
Features of RTOS.

- Clock and Timer Support
- Real-Time Priority Levels
- Fast Task Preemption
- Requirements on Memory Management.

2) Earliest Deadline First (EDF) :-

Liu and Layland [3] described the Rate-Monotonic (RM) and the Earliest Deadline First (EDF) scheduling policies. EDF algorithm is the type of dynamic scheduling algorithm which depends upon the deadline of the task [8].The task with the nearest deadline gives the highest priority. EDF algorithm schedules the process and to create particular process some of the parameter is required such as Process Start Time, Execution Time, Deadline of Process, Release Time and the Load Time of each process. After creating the process , the process are store in a queue known as priority queue and the priority of that process depend upon the deadline . The below Fig.1 shows the work flow of EDF algorithm.
The limitation of EDF algorithm is that it cannot handle the overloaded condition properly. In overload situations, the performance of the system decreases and the desired output will not be achieved when scheduler schedules the process using EDF scheduling algorithm.
The periodic task $\tau_i=(c_i,p_i)$ is characterized by two parameters i.e. execution time “c_i “and period “p_i”. The utilization of periodic task “τ_i“ is defined as [9].

$$ui = \frac{ci}{pi} \qquad (1)$$

A task can be practicability scheduled using EDF algorithm if the total utilization of a task is

$$U(\tau) = \sum_{i=1}^{n} ui \qquad (2)$$

When the process is underloaded, the EDF algorithm executes all the process and CPU usage for that process is also minimum but when the process is overloaded some process fails to execute. This is the drawback of EDF algorithm. To overcome this drawback the Hybrid scheduling algorithm combines Earliest Deadline First with the Ant Colony Optimization scheduling algorithm.

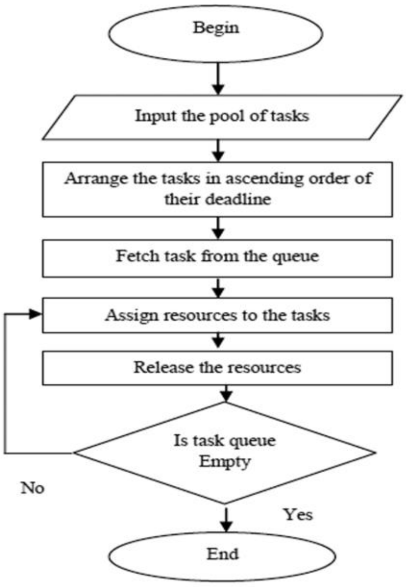

Fig. 1 EDF Scheduling Algorithm Flow Chart

3) Ant Colony Optimization (ACO) :-

ACO algorithm was proposed by Dorigo and Gambardella in the early 1990s and by now has been successfully applied to various combinatorial optimization problems [10]. This scheduling algorithm is based on the nature of real ant, where each ant constructs a path and one or more ants simultaneously active at the same time. ACO scheduling algorithm mainly working with the time slice manner. The limitation of ACO algorithm is, that it takes more time for execution than the EDF algorithm.
In ACO algorithm, each ant is called as a "node" and each of them will start their journey from different node. To apply in ACO scheduling algorithm each node is considered as a task and possibility of each node depend upon the "pheromone" value τ and heuristic value η and it is calculated as [11].

Where,

$pi(t)$ is the probability of i^{th} node at time t.
τi is pheromone on i^{th} node at time t.
ηi is a heuristic value of i^{th} node at time t and is determined by,

$$\eta i = \frac{k}{di - t} \quad (3)$$

Here,

t is current time k is constant and D_i is absolute deadline of i^{th} node.
"α" and "β" are constants which decide importance of τ and η .

The ACO algorithm is applied to the process with the same parameter as the EDF algorithm. The process is switched from EDF to ACO when the process is loaded and the process count value is more. After switching to ACO, process executed depends upon the execution time, the process with less execution time is executed first. The advantages of ant colony optimization is it indirectly communication between ants using pheromone variables. It also handles the overloaded condition. The behavior of the ant is successfully applied to several optimization problems. Below Fig. 2 shows the work flow of ACO scheduling algorithm.

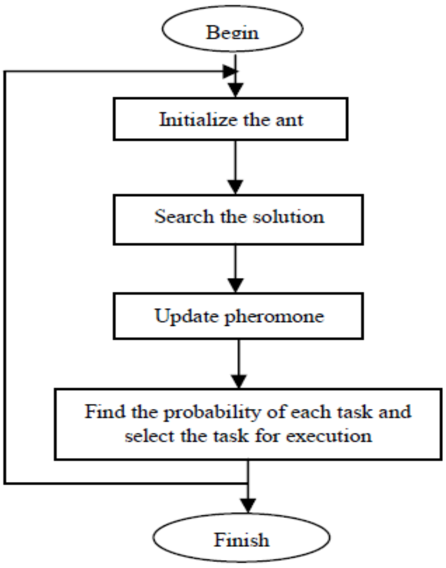

Fig.2 ACO Scheduling Algorithm Flow Chart

II. LITERATURE REVIEW

Krithi Ramamritham and John Atankovic,(1994) Summarizes the state of the real-time field in the areas of scheduling and operating system kernels and discuss paradigms underlying the scheduling approaches. In this particular paper, scheduling paradigms were identified: like static table-driven scheduling, static priority preemptive scheduling, dynamic planning-based scheduling and dynamic best effort scheduling. Allocation and scheduling the communication as well as processing resources in an integrated fashion still remains a problem awaiting efficient and flexible solutions.

Jinkyu Lee and Kang G. Shin,(2014) it defines cp-EDF in which the CP policy is applied to EDF, and analyze its schedulability. This schedulability analysis is then utilized to develop an algorithm that assigns the optimal control parameters of cp-EDF. The earliest-deadline-first (EDF) policy has been widely studied for the scheduling of real-time jobs for its effectiveness and simplicity. The schedulability analysis is then utilized to develop an algorithm that assigns the optimal control parameters of cp-EDF. Our in-depth evaluation has demonstrated that cp-EDF with the optimal parameter assignment improves EDF schedulability over existing preemption policies by up to 7.4%.

M.Kaladevi and Dr.S.Sathiyabama,(2010) The main objective of this paper is to compare two important task schedulers such as Earliest Deadline First (EDF) scheduler and Ant Colony Optimization Based (ACO) scheduler. It also presents a system that can be schedule multiple independent tasks. Tasks can obtain minimum guaranteed execution time with its performances. The real-time constraints in Real time system for scheduling the task, different scheduling algorithms were used. Most of the real-time systems are designed using priority based preemptive scheduling and worst case execution time estimates to guarantee the execution of high priority tasks. The main limitations of both algorithms are; During under loaded condition, the execution time taken by the EDF algorithm (i.e. less time) and during overloaded condition, it gives performance of ACO based scheduling algorithm (i.e. more efficiency).

III. PROBLEM STATEMENT

The purpose of an Real Time system[1] is to fulfill it's limitation in particular given time or its deadline.The selection of an appropriate algorithm is necessary to fulfill RTOS task constraints. As per the literature survey made so far there are some problems in scheduling algorithm to handle the process operations (like process crash, abnormally termination of process etc.). In current system RM scheduling algorithm is use to handle the process operation [12].RM works on a fixed priority scheduling algorithm and it does not support "a-periodic" task operation. Fixed priority algorithms are a sub-class of the more general class of dynamic priority algorithms; the priority of a task does not change in these type of algorithm. In RM it does not handle dynamic task . The most important (and analyzed) dynamic priority algorithm is EDF. The priority of a job (instance\task) is inversely proportional to its absolute deadline. In other words, the highest priority job is the one with the earliest deadline; If two tasks have the same absolute deadlines, then choose one of the two at random (ties can be broken arbitrary). The priority is dynamic since it changes for different jobs of the same task.

IV. PROPOSED APPROCHES

A) PROPOSED WORK

- The main aim to design “Hybrid Algorithm” is to substantially improve the performance of system and generate result in a more accurate and timely manner (in term of RTOS constraints).
- In order to overcome/reduce the overload and underload condition problem we design new scheduling algorithm i.e. Hybrid Scheduling Algorithm; i.e. the fusion of EDF and ACO algorithm.
- The limitation of the EDF algorithm is that it cannot handle the overloaded condition properly. When the overload condition occurs in RTOS system, EDF algorithm fails to execute and that may causes the system failure.
- On the other the ACO algorithm handles the overload condition properly but the execution time required for handling process is more than the EDF scheduling algorithm.
- This is the limitation of ACO; So to overcome the limitation of both the algorithm the new algorithm is developed which is called as the “Hybrid Scheduling Algorithm”.

B) Process Handle Mechanism In EDF

- The working of the EDF algorithm is as follows ; the processes are added one by one in a process queue or EDF queue.In detail let $P=(p_1,p_2,p_3........p_n)$denote a set of process $P_i=(r_i,e_i,d_i)$ is characterized by its release time “r_i”, execution time “e_i” and deadline of the process “d_i”.
- The process is added in the process queue and the process with highest priority is switched to EDF algorithm and the switching depends upon deadline of the process .
- When the process is underloaded, the EDF algorithm executes all the process and CPU usage for that process is also minimum but when the process is overloaded some process fail to execute.
- This is the main drawback of an EDF algorithm [13]. To overcome this drawback, the fusion of both the scheduling algorithm is design;i.e. Hybrid algorithm.(refer Fig.4)
- The following below Fig.3 shows the process handle mechanism of an EDF scheduling algorithm.

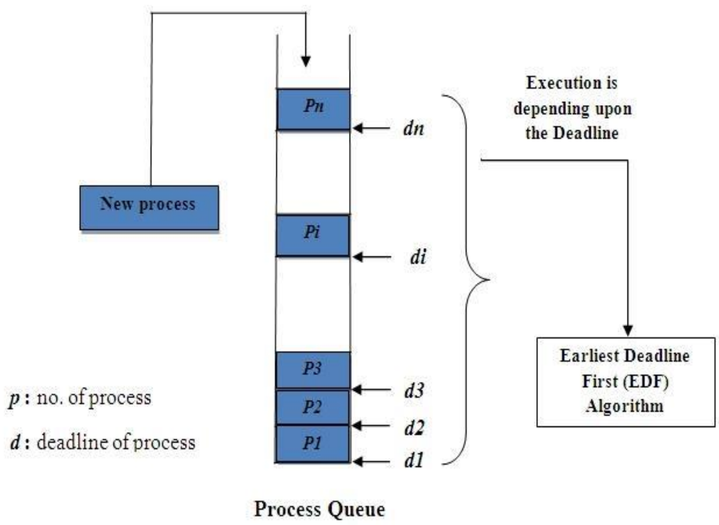

Fig.3 Process Working Mechanism in EDF

C) Process Handle Mechanism In ACO

- In ACO algorithm, the algorithm checks the release time as well as the execution time of each process.
- After that when user wants to apply the ACO algorithm for the same process which is created.

D) Proposed architecture

- The below Fig.4 shows the fusion of two scheduling algorithm. The Hybrid Algorithm is the fusion of EDF and ACO algorithm. The Hybrid Algorithm schedules the process on single processor when it is preemptive.
- The advantage of the "Hybrid algorithm" is that it will automatically switches between the EDF and ACO algorithm when the overload and underload condition occurs in the system and overcome the limitation of both the algorithms (i.e.EDF and ACO) algorithms.

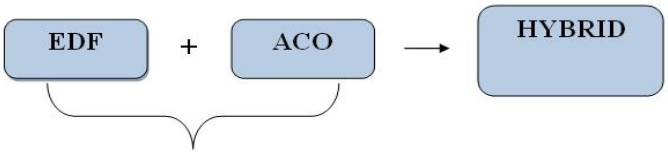

Fig.4 Proposed Architecture

E) Success Ratio(SR) and CPU Utilization

In real time system, meeting process deadline is the most important factor. For that most convenient performance metric is the Success Ratio and it is defined as in [14].

$$Sucess\ Ratio = (Number\ of\ process\ schedule)/(Total\ number\ of\ process\ ) \quad (5)$$

It is same as the number of process scheduled. So the Table I shows that the Success Ratio and CPU utilization by using the Hybrid scheduling algorithm when system is underloaded. Table II shows that the Success Ratio and CPU utilization by using the Hybrid scheduling algorithm when system is overloaded .CPU utilization is the total amount of work handle by the CPU. CPU utilization depends upon the task or processes. Refer equation (2). When adding a new task in the system then the CPU Utilization is high.

Depending upon the parameters process is created.Fig.3 shows the process creation mechanism. By using the above parameter process is created one by one. Let the process $P1$ is created $P1$=($d1$,$r1$,$e1$) Where, $d1$ is the deadline of process $P1$. $e1$ is the total time required for process $P1$. $r1$ is the release time of $P1$. Process $P1$ is created and the load of process $P1$ is calculated as

$$Load(l) = \sum_{i=0}^{m} \frac{ei}{qi} \quad (6)$$

Where,

ei is the execution time required for each process.

qi depends upon the value of period pi and the deadline di.

- Below table shows the success ratio (SR) and CPU utilization when the load is less than or equals to one (1).
- Below graph shows the result in graphical way.

Table I

Success Ratio (SR) and CPU Utilization when Load ≤ 1

Load	%SR			% CPU Utilization		
	EDF	ACO	Hybrid	EDF	ACO	Hybrid
0.1	100	100	100	11.15	12.14	12.37
0.2	100	100	100	16.37	16.97	16.81
0.3	100	100	100	24.66	24.66	24.66
0.4	100	100	100	34.7	33.81	34.7
0.5	100	100	100	48.9	49.71	49.9
0.6	100	100	100	54.88	54.88	54.88
0.7	100	100	100	68.11	69.24	69.24
0.8	100	100	100	76.54	77.82	77.8
0.9	100	100	100	89.61	89.54	89.11
1.0	100	100	100	99.31	98.54	99.37

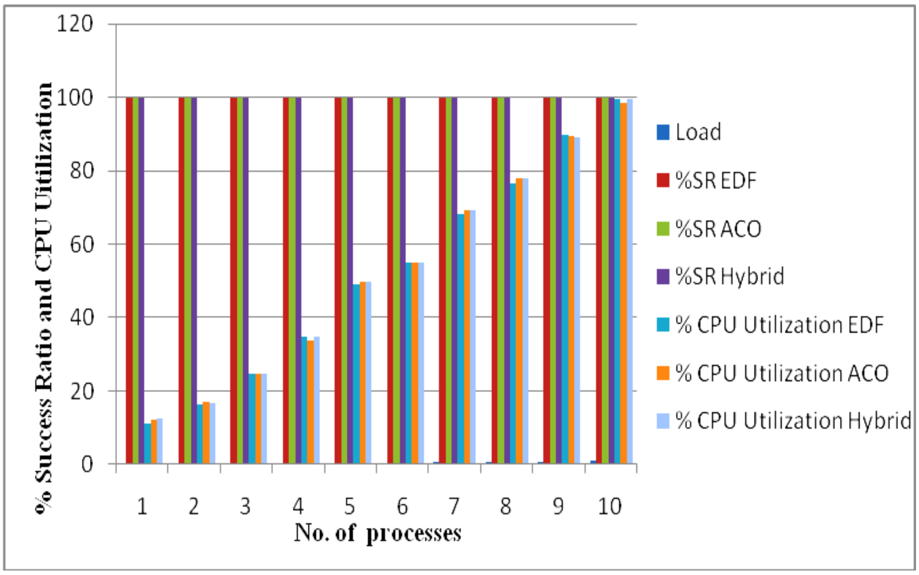

Fig.5 Success Ratio (SR) and CPU Utilization when Load ≤ 1

- Below table shows the success ratio (SR) and CPU utilization when the load is greater than one (1).
- Below graph shows the result in graphical way.

Table II

Success Ratio and CPU Utilization when Load > 1

Load	%SR			% CPU Utilization		
	EDF	ACO	Hybrid	EDF	ACO	Hybrid
1.0	48	64	55	45	60	57
1.2	43	59	50	32	55	37
1.4	33	50	41	24	47	29
1.6	23	53	30	21	34	28
1.8	20	39	35	19	29	26
2.0	18	31	27	16	26	24
2.2	15	26	23	14	25	21
2.4	13	21	18	12	23	17
2.6	9	18	12	10	20	15
2.8	8	15	13	8	15	13
3	7	13	11	7	13	10
3.2	6	10	9	6	10	8
3.4	5	8	7	5	8	6

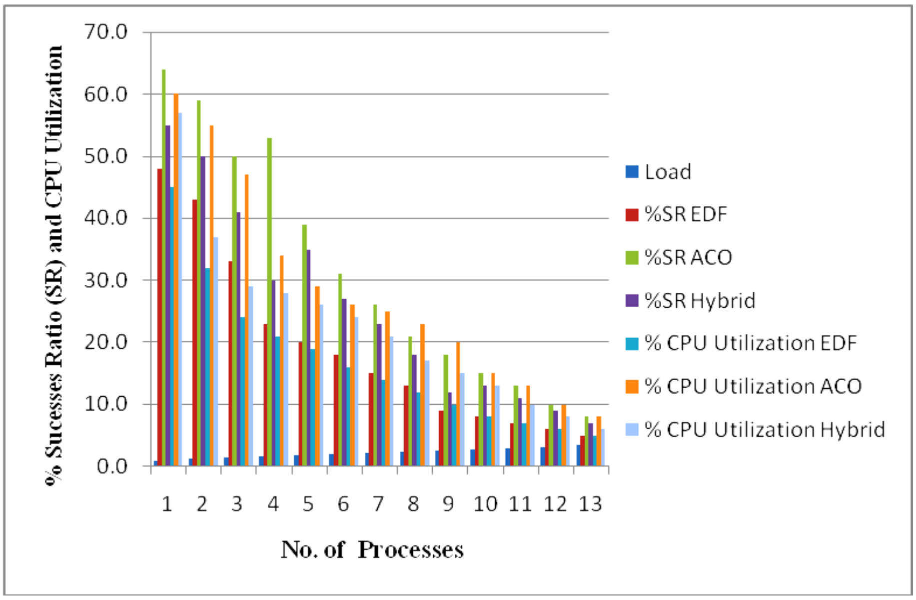

Fig.6 Success Ratio (SR) and CPU Utilization when Load > 1

V. RESEARCH METHODOLOGIES

A) Analysis

We analyzed that the EDF scheduling algorithm does not handle overload condition properly. So, to overcome this situation we propose new scheduling algorithm, this propose algorithm will be handle the job operation even at overload as well as under load condition. The advantage of our propose algorithm(Hybrid algorithm) is that it works on dynamic and static type of scheduling algorithm (i.e. EDF & ACO).

1) Comparison of Scheduling Algorithms

Fig.8 shows the execution time required for EDF, ACO and the Hybrid algorithm and Fig.7 shows number of process schedule by the same algorithm. When the 20 processes are added then the time required for execution by EDF algorithm is less than the ACO and the Hybrid algorithm. In this manner EDF is better for execution but it cannot handle the overloaded condition. Table III show the comparison of number of process schedule by the EDF, ACO and the Hybrid algorithm. Graph shows that if the user gives 20 processes and at that time processor load is 5% then the EDF algorithm schedule the 13 out of 20 processes and rest of three processes failed to execute. At the same time if the user applied ACO algorithm then 17 processes execute and three process is failed to execute because of the execution time.

When the Hybrid algorithm is applied then all the processes are executed and all the process deadline is meet. The hybrid algorithm is very useful when future workload of the system is unpredictable.

Table III

Comparision of scheduled process

No of Process	EDF	ACO	Hybrid Algorithm
20	13	17	20
40	31	34	40
60	46	58	60
80	70	76	80
100	85	95	100
120	111	119	120
140	125	130	136
160	153	154	159
180	169	175	180
200	179	189	200

.

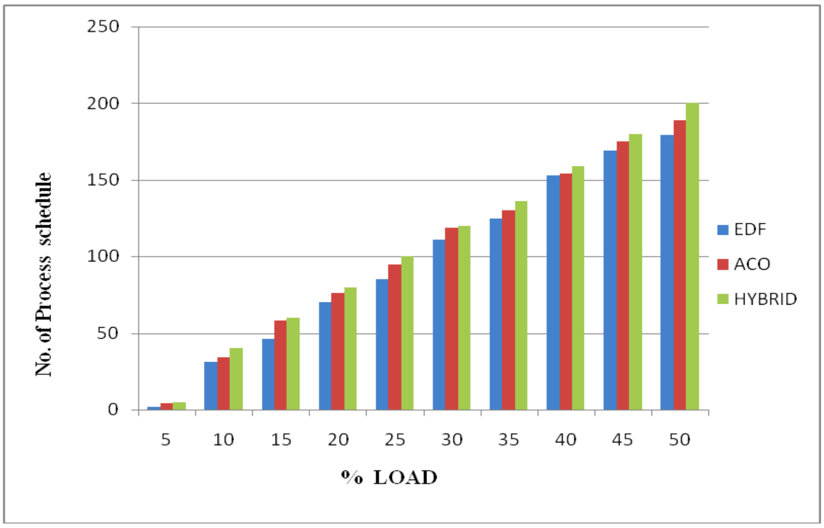

Fig.7 Comparison of Number of Process Schedule

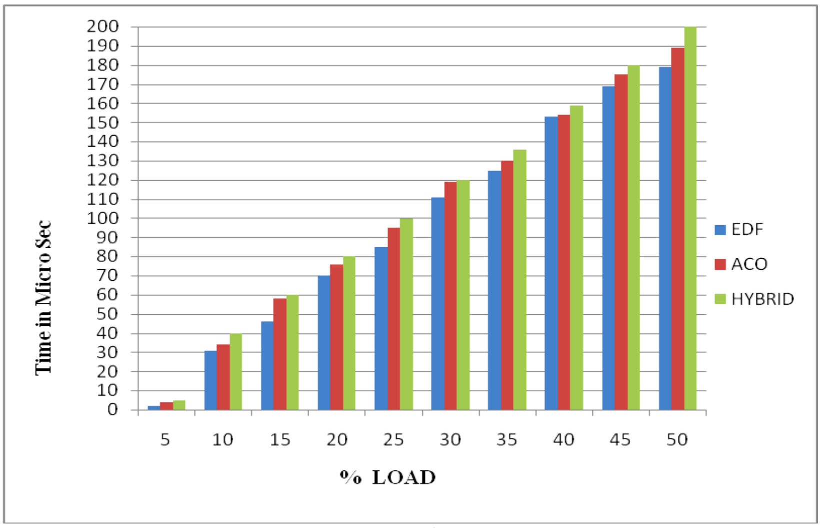

Fig.8 Comparison of Execution Time

2) DESIGN

The whole paper is consisting of Three modules:-

- Module1- Customized Task Manager (for handles process operations).
- Module2- RTOS-Scheduler (for schedules the jobs with their parameters).
- Module3- Hybrid Algorithm (proposed system).

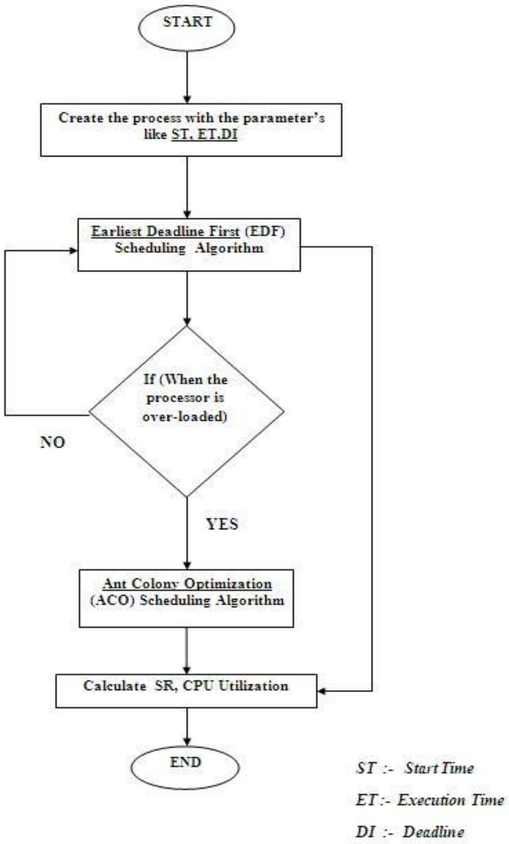

Fig.9 Flowchart of the proposed system

- The above Fig.9 Shows the process creates with the parameter like start time, execution time, deadline, Number of process are scheduled in the process queue.
- During the underloaded condition, that time algorithm uses EDF scheduling algorithm and priority of that process is decided dynamically depending according to its deadline [15].
- During overloaded condition, process switches to the ACO scheduling algorithm and calculates its execution time and minimum time required for each process then process which contain the minimum execution time is executed first.

VI. CONCLUSION

The hybrid Algorithm is a dynamic scheduling algorithm and it is beneficial for single processor real-time operating systems. The algorithm is useful when future workload of the system is unpredictable. The hybrid Algorithm combines the EDF and ACO algorithm. The hybrid Algorithm schedules the process on single processor when it is preemptive. The advantage of the algorithm is that it automatically switches between the EDF and ACO algorithm and overcome the limitation of both the algorithms (EDF and ACO) algorithms. The hybrid Algorithm requires less time for execution as compared to EDF and ACO and the algorithm gives the result when the system is overloaded. Memory usage of the system is increased when the load of the process is increased**.**

REFERENCES

Papers:

[1]Krithi ramamritham and john a. atankovic, "Scheduling Algorithms and Operating Systems Support for Real-Time Systems," proceedings of the IEEE, vol.82.no. 1 , january 1994.

[2]M.Kaladevi and Dr.S.Sathiyabama, "A Comparative Study of Scheduling Algorithms for Real Time Task," International Journal of Advances in Science and Technology, Vol. 1, No. 4, 2010.

[3]C. Liu and J. Layland, “Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment,” J. ACM, vol. 20, pp. 46–61, 1973.

[4]M. L. Dertouzos “Control Robotics: The Procedural Control of Physical Processes,” Information Processing, 1974.

[5]Jinkyu Lee and Kang G. Shin, “Preempt a Job or Not in EDF Scheduling of Uniprocessor Systems,” IEEE Transaction on computers , vol. 63, no. 5, May 2014.

[6]Marko Bertogna and Sanjoy Baruah , “Limited Preemption EDF Scheduling of Sporadic Task Systems,” IEEE Transactions on Industrial Informatics, Vol. 6, no. 4, November 2010.

[7]Fengxiang Zhang and Alan Burns, “Schedulability Analysis for Real-Time Systems with EDF Scheduling,” IEEE Transactions on Computers, Vol. 58, no. 9, September 2009.

[8]Jashweeni Nandanwar, Urmila Shrawankar, “An Adaptive Real Time Task Scheduler,” IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 1, November 2012.

[9]Xuefeng Piao, Sangchul Han, Heeheon Kim, Minkyu Park and Yookun Cho , “Predictability of Earliest Deadline Zero Laxity Algorithm for Multiprocessor Real-Time Systems,” Proceedings of the Ninth IEEE International.

[10]Nada M. A. Al Salami, “Ant Colony Optimization Algorithm,” UbiCC Journal, Volume 4,Number 3, August 2009.

[11] W.N. Chen and J. Zhang, “An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem with Various QoS Requirements,” IEEE Trans. System, Man, and Cybernetics- Part C, vol. 39, no. 1, pp. 29-43, Jan. 2009.

[12]Sachin R. Sakhare and Dr. M.S. Ali , “Genetic Algorithm Based Adaptive Scheduling Algorithm for Real Time Operating Systems,” International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.3, September 2012.

[13]A.F.M.Suaib Akhter, Mahmudur Rahman Khan,Shariful Islam , “Overload Avoidance Algorithm for Real-Time Distributed System,” IJCSN International Journal of Computer Science and Network Security, Vol. 12 no.9,September 2012.

[14] Wei-Neng Chen and Jun Zhang, “Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler,”IEEE Transactions on Software Engineering, Vol. 39, no. 1, January2013.

[15] Michael A. Palis, “The Granularity Metric for Fine-Grain Real-Time Scheduling,” IEEE Transactions on Computers, Vol. 54, no. 12, December 2005.

[16]Giorgio C. Buttazzo, Marko Bertogna and Gang Yao , “Limited Preemptive Scheduling for Real-Time Systems,” IEEE Transactions On Industrial Informatics, Vol. 9, no. 1, February 2013.

Websites:

[17]www.embedded.com/electronics-blogs/beginner-s-corner/MichaelBarr/Introduction to rate monotonic scheduling,Feb-2002.

Books:

[18]Silberschatz, Galvin and Gagne , “Operating System Concepts with Java,” 8th Edition.

