Designing of Modified Fuzzy based CPU Scheduling Algorithm

Samiksha Deshwal
CSE Department, BPS Mahila Vishwavidhalaya, KhanpurKalan, Sonipat
Haryana, INDIA
deshwal.samiksha@gmail.com

Mr. Vinod Saroha
Assistant Professor, CSE Department, BPS Mahila Vishwavidhalaya, KhanpurKalan, Sonipat
Haryana, INDIA
vnd.saroha@gmail.com

Abstracts:
CPU Scheduling is the base of multiprogramming. Scheduling is a process that decides order of task from a group of multiple tasks that are able to execute. There are ranges of CPU scheduling algorithms accessible, however it's terribly troublesome task to make a decision that one is best. This paper discusses the planning and implementation of modified fuzzy primarily based CPU scheduling algorithm. This paper presents a replacement set of fuzzy rules. It demonstrates that scheduling done with new priority improves average waiting time and average turnaround time.

Keywords -- Fuzzy logic, Operating System, Priority, Scheduling Algorithms, Process Scheduling, average turnaround time, dynamic time quantum, average waiting time.

I. INTRODUCTION

Scheduling is concern with the optimal allocation of resources to activities in a specified time. It is a key feature in multitasking, multiprocessing and real-time operating system design. It is a decision making process that deals with the allocation of common resources to various tasks at different time periods to achieve multiple objectives. Scheduling problem involves jobs that must schedule on machines subject to certain constraints to optimize some objective functions. It is done by means of scheduler and dispatcher. Scheduler selects the processes that are ready to execute. A dispatcher is a module that performs the work of passing the CPU to the next selected process. The time to do this is called dispatch latency. In real time environments, such as embedded systems for automatic control in

industry, the scheduler needs to ensure that processes meet the deadlines that are crucial for keeping the system safe. The resources and task can be of different forms in homogeneous/heterogeneous organization.
Five basic features to decide best scheduling algorithm are as follow [1, 2]:

- Utilization of CPU time
- Throughput
- Turnaround Time
- Waiting Time
- Response time

A. Scheduling Criteria :

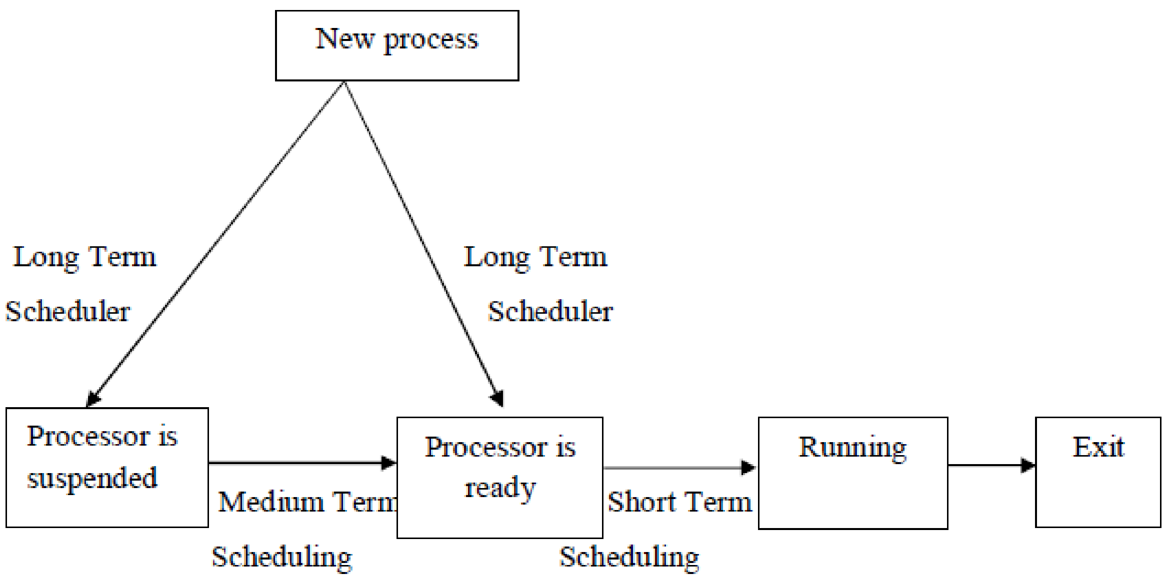
Scheduling helps processes to perform input/output operation in the normal course of computation. Since input/output operation require more time to complete than CPU instructions, multiprogramming system allocate the CPU to process which invokes an input/output operation. 1.1 Types of Processor Scheduling The aim of processor scheduling is to assign processes to be executed by the processor in a way that meets system objectives, such as response time, throughput, and processor efficiency. In many systems, this scheduling activity is broken into three separate functions (i) Long term scheduling (ii) Medium term scheduling (iii) Short term scheduling.

1.1 *Long term scheduling:* The long term scheduler determines which programs are admitted to the systems for processing, thus, it controls the degree of multiprogramming. Once admitted, a job or user program becomes a process and is added to the queue for the short term scheduler. In a batch system, long term scheduler creates processes and forms the queue wherever it is possible. The more processes are created; the smaller is the percentage of time for which each process can be executed. Long term schedulers may limit the degree of multiprogramming to provide satisfactory service to the current set of processes.

1.2 Midterm scheduling: The scheduling of processes is mainly based on the requirement of the resources. It is essentially concern with memory management and often design as a memory management subsystem of an operating system. It temporarily removes a process from the main memory which is of low priority or has been inactive for a long time. This is known as "swamping out" of a process. The scheduler may decide to swamp out the process which is frequently page-faulting or a process which is taking large amount of memory. Its efficient interaction with the short term scheduler is very essential for the performance of the systems with virtual memory.

Samiksha Deshwal *et al*, International Journal of Computer Science and Mobile Applications,
Vol.3 Issue. 6, June- 2015, pg. 66-79 **ISSN: 2321-8363**

1.3 Short Term Scheduling: In terms of frequency of execution, the long term scheduler executes relatively infrequently and makes the coarse grained decision of whether or not to take on a new process and which one to take. Short term scheduler is invoked whenever an event occurs that may lead to the blocking of the current process that may provide an opportunity to preempt a currently running process in favor of another. CPU scheduling decisions can occur on the given conditions: (a) either the running process changes from running to waiting state or when the running process terminates (b) The waiting process becomes ready and (c) the current process switches from running to ready state.



B. Goals of Scheduling Algorithm for Different Systems

There are some goals that must be achieved in order to perfectly schedule the task on the processor. Some of these goals are mentioned below: (i) Fairness: Fairness is important under all circumstances. A scheduler makes sure that each process gets its fair share of the CPU and no process suffer indefinite postponement. Note that giving equivalent or equal time is not fair [25]. (ii) Policy Enforcement: The scheduler has to make sure that system's policy is enforced. For example, if the local policy is safety then the safety control processes must be able to run whenever they want to, even if it means delay in payroll processes. (iii) Efficiency: Scheduler should keep the system busy cent percent of the time when possible. If the CPU and the entire input/output device run for entire time, more work gets done per second. (iv) Meeting deadlines : Scheduler should finish all its processes before the deadline of process otherwise catastrophic results can occur.

C. Scheduling Techniques:

Scheduling algorithms[3,4] can be classified in different categories –

- **Preemptive/ Non-preemptive Techniques** – In case of preemptive scheduling algorithm, scheduler can stop/postpone execution of any task. But in case of non-preemptive algorithms it is not possible till job completion or running job voluntarily give up resources.[2,3]
- **Static /Dynamic priority scheduling** – Static priority do not change, but dynamic priority can be modified. This paper considers three scheduling algorithms Shortest Job First, Priority Scheduling and Fuzzy based CPU scheduling.

❖ **Shortest Job First (SJF) Scheduling Algorithm:** This algorithm schedules jobs in keeping with their execution time, job with lowest execution time scheduled first and job with largest execution time scheduled last. This is often a non-preemptive rule. Its variant called shortest remaining time is pre-emptive. This algorithm reduces average waiting time. Shortest job first is effectively used in case of interactive processes that typically follow a pattern of alternating between looking ahead to a command and execution of it. If the execution time of a technique is taken into account a separate "job", past behavior can indicate that methodology to run next, supported estimate of its fundamental quantity[5]

❖ **Priority Scheduling Algorithm:** The operating system decides priority to every technique, therefore the hardware arranges the processes among the ready queue so as of their priority. Processes with low priority is also interrupted once incoming process has high priority. [4]There is a variety of priority scheduling; SJF is additionally a special case of priority scheduling. This algorithmic rule typically face problem of infinite block and starvation, this problem is removed by aging technique.

❖ **Fuzzy based scheduling**: Fuzzy based scheduling algorithm proposed by Kadhim,Shatha J.,and Kasim M. Al-Aubidy. This algorithm takes input both job priority and execution time and decides new priority using some fuzzy rules. This algorithm uses some linguistic variables for priority given by system, new computed priority and execution time. Performance of this algorithm compared with SJF and classical priority scheduling methods

II. FUZZY LOGIC

Fuzzy logic is a style of multi-valued logic. It deals with approximation rather than exactness. In contrast to classical sets (Classical set takes true or false values) fuzzy logic variables (also known as linguistic variable) can have a truth value that ranges in interval between 0 and 1. Fuzzy logic has been prolonged to grasp the concept of fractional truth, where the truth value may range between completely true and completely false. Moreover, when linguistic variables are practiced, these degrees may be determined by specific methods.[6]

In recent years the application of fuzzy systems in real world issues is increasing due to the actual fact that fuzzy systems will affect linguistic knowledge beside the numerical knowledge. Now a day an outsized range of researchers are shifting towards fuzzy logic.[7] Fuzzy logic has shown itself to be a strong design and analysis methodology in control theory, enabling the implementation of advanced knowledge-based control methods for complicated dynamic systems like those rising applications for systems and artificial biology.[8] A priority scheduler has been developed for mobile ad hoc networks by C. Gomathy et al. in.[9] Rajani Kumari et al. discussed air conditioning system with fuzzy logic.[10] A. Saleh proposed a grid-scheduling algorithm on fuzzy matchmaking approach[11]

Fuzzy logic based algorithms are getting more popularity now a day as Reza salami et al. developed an efficient task scheduling algorithm for computational grids using NSGA II with fuzzy variance based crossover . In this algorithm two functions are defined to generate two inputs for fuzzy based system. Variance of costs and presence of resource in scheduling are used to specify probability of crossover.[12] It provides better solution in less number of iterations. H. Chuan et al. discussed fuzzy job shop scheduling problem based in interval number theory.[13] S. Mandloi et al. gives new idea based on ant colony optimization with genetic parameter selection for job scheduling in computational grid.[14] Fuzzy based scheduling algorithms implemented in cloud also. C. Yi Chun et al. proposed a fuzzy based dynamic load decision making scheme in cloud computing. [15]

III. PROPOSED WORK

Fuzzy Round Robin CPU Scheduling Algorithm:

In Round Robin Scheduling the time quantum is fixed and then processes are scheduled such that no process get CPU time more than one time quantum in one turn. Too Large time quantum increases the response time of the processes too much which may not be tolerated in interactive environment. Too small time quantum causes unnecessarily frequent context switches leading to more overheads resulting in less throughput. In this work a method using Fuzzy Logic has been proposed that decides a value that is neither too large nor too small such that every process has got reasonable response time and the throughput of the system is not decreased due to unnecessarily

 ISSN: 2321-8363

context switches. In Round Robin scheduling CPU scheduler goes around the ready queue allocating the CPU to each process for a time interval of up to one time quantum. If the process do not complete its execution within the time quantum, the process go to the end of ready queue and process switch occurs where state of the running process is put onto stack and the state of the next process is taken from the stack and its execution is restarted. If the time required for the running process is slightly more than time quantum even by a fraction, even then Scheduler prempt the process resulting into context switch. This causes more waiting time for that process and more overheads due to unnecessary context switches. In this work an algorithm named fuzzy round robin scheduling algorithm has been proposed that tries to remove these two problems.

FIS for Finding Time Quantum :

The Fuzzy Inference System for finding the time quantum has got two inputs and one output. First input is N that specifies the number of user/processes in the system and second input is the average burst time of the processes in the ready queue. Time Quantum is the output of the FIS. Block diagram, rules base, surface view and rule view of the FIS designed are shown below in Fig. 1, Table 1, Fig. 2 and 3 respectively. This FIS solves the first problem

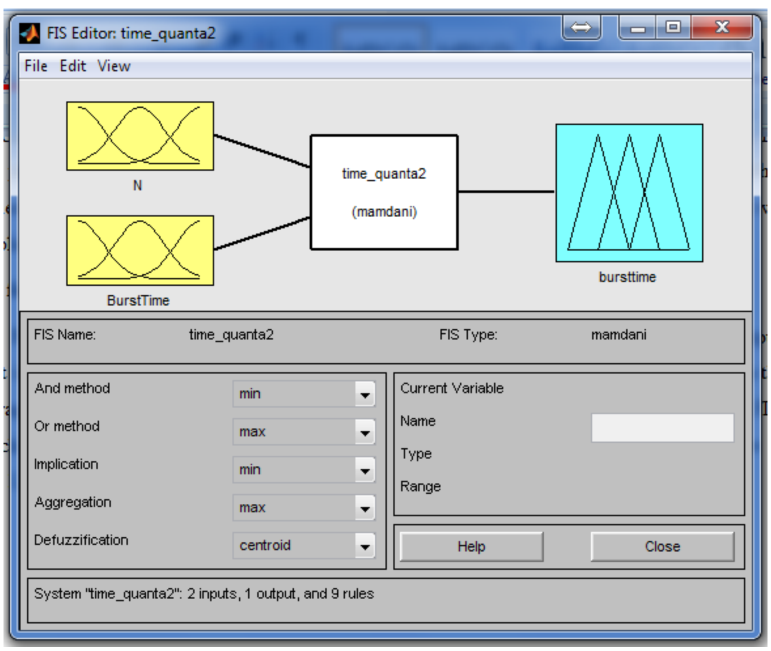


Figure 1: Fuzzy logic interface for round robin fashion

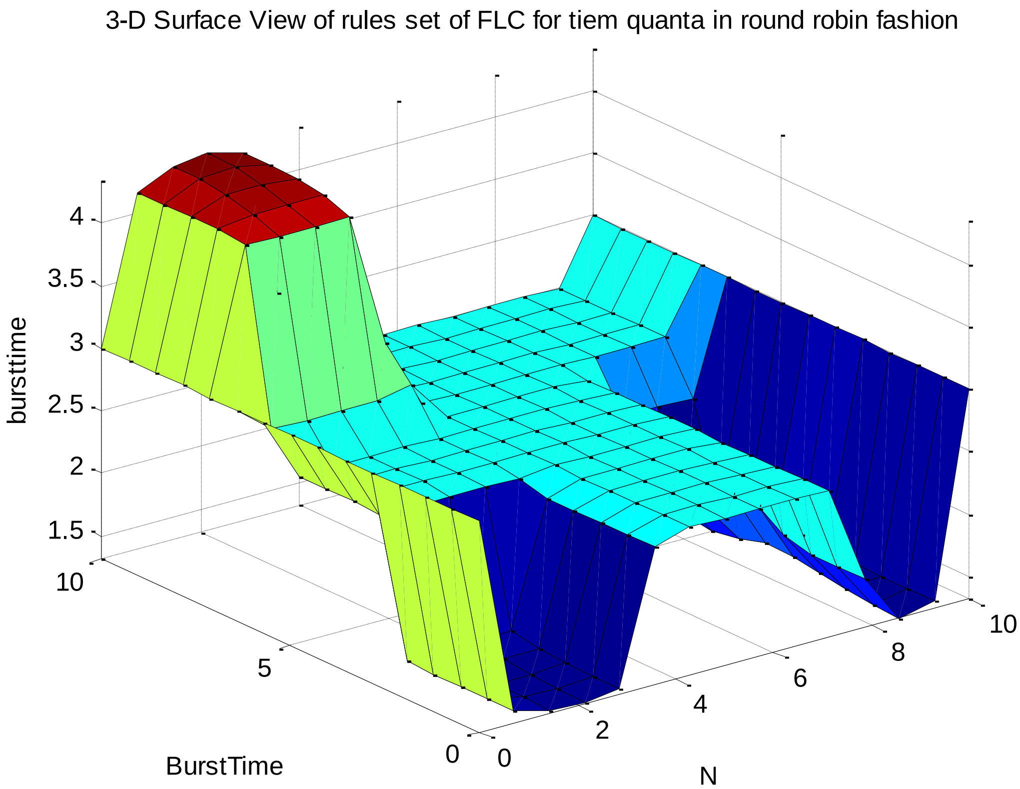


Figure 2: Surface view of rules

Table 1 : fuzzy logic rules for calculating quantum time in round robin fashion

Number of process \ **Burst time**	**Low**	**Medium**	**High**
Low	Low	Medium	Low
Medium	Medium	Medium	Low
High	High	Medium	Medium

- This scheduling deal with some fuzzy rules and these rules are based on assigned priority and execution time. This work is proposed to compute the New Priority for all tasks using pre priority and execution time with the help of Mamdani type inference.[16]

This paper use suitable linguistic variables as input and output for compute a crisp value for new priority. Pre Priority measured as Very Low, Low, Medium, High and Very High. Execution Time measured as Very Small, Small, Medium, Long and Very Long.[17] New Priority measured as Very Low, Low, Medium, High and Very High. The proposed scheduling is a collection of linguistic fuzzy rules which describe the relationship between defined input variables and output.[18]

Fuzzy rules set used for designing is given in table2.

Old Priority / Execution time	**Very Low**	**Low**	**Medium**	**High**	**Very High**
Very Less	V. High	V. High	V. High	V. High	V. High
Less	Medium	Medium	High	High	V. High
Medium	Very Low	Low	Medium	Medium	Medium
Long	Very Low	Very Low	Low	Low	Low
Very Long	Very Low	Very Low	Very Low	Low	Low

Table 2.

Samiksha Deshwal *et al*, International Journal of Computer Science and Mobile Applications,
Vol.3 Issue. 6, June- 2015, pg. 66-79 ISSN: 2321-8363

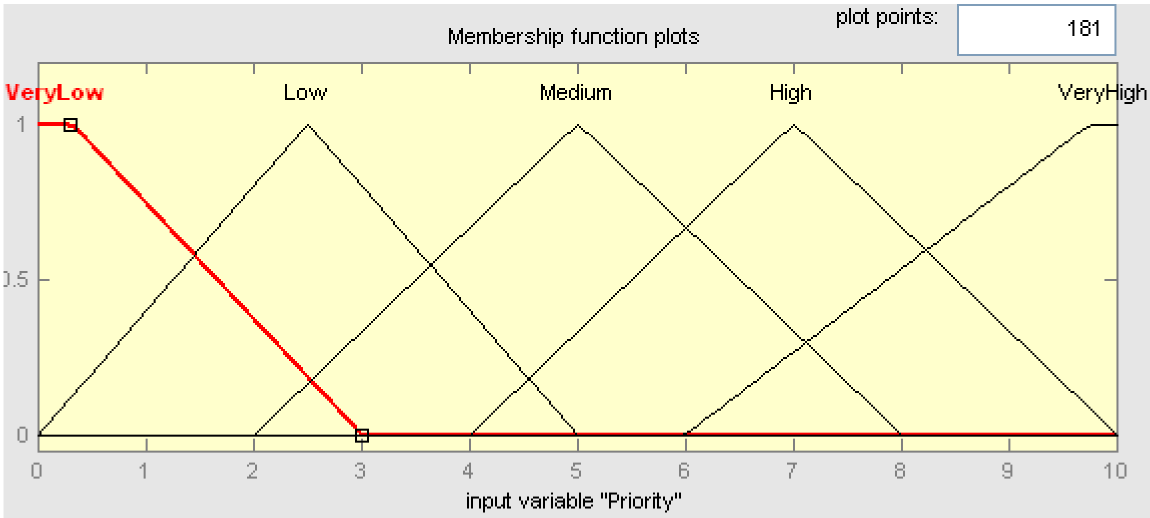


Fig. 3: Membership Function for Priority

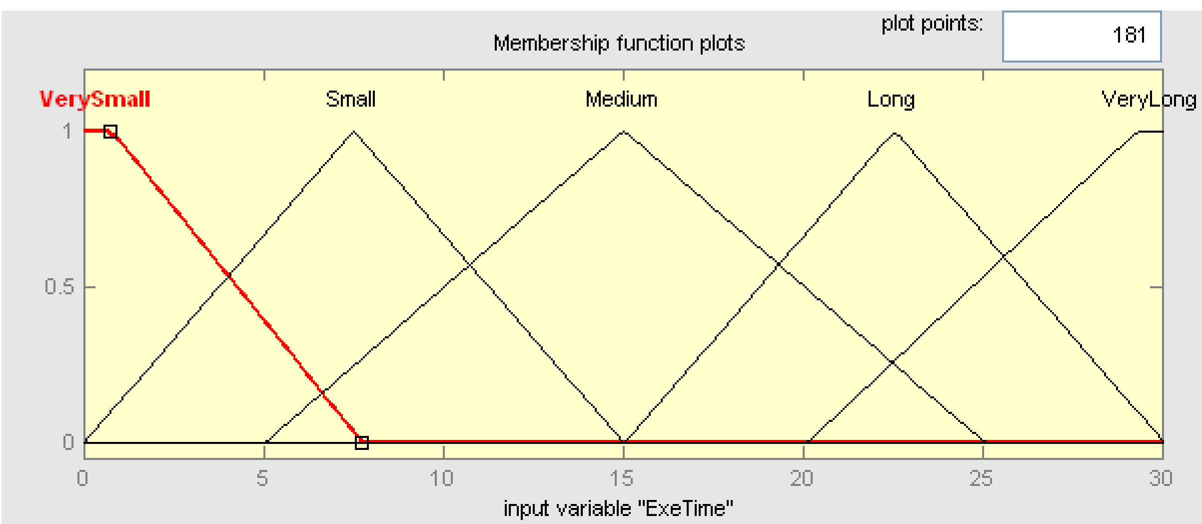


Fig. 4: Membership Function for Execution Time

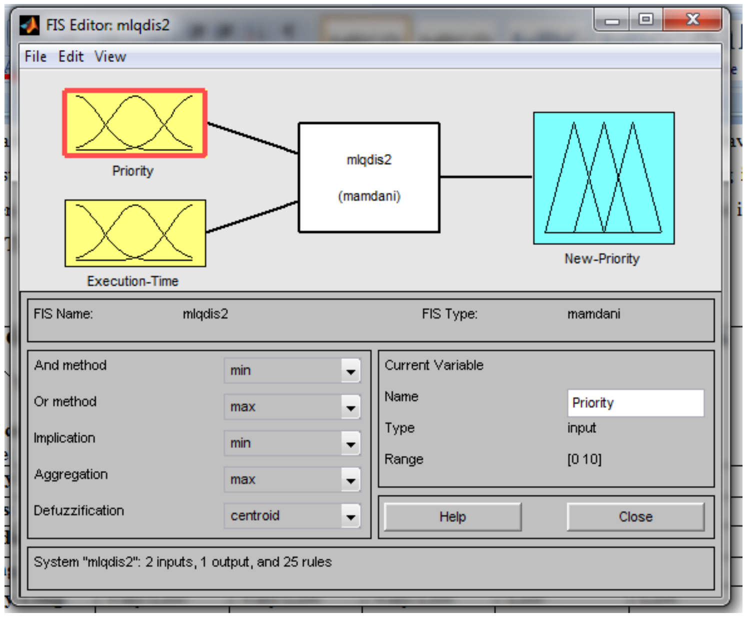


Figure 5: User Interface of Fuzzy Logic

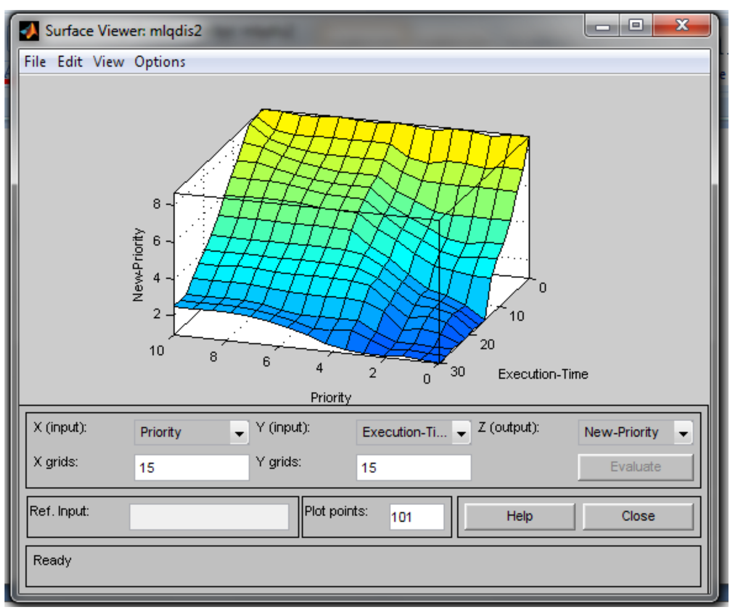
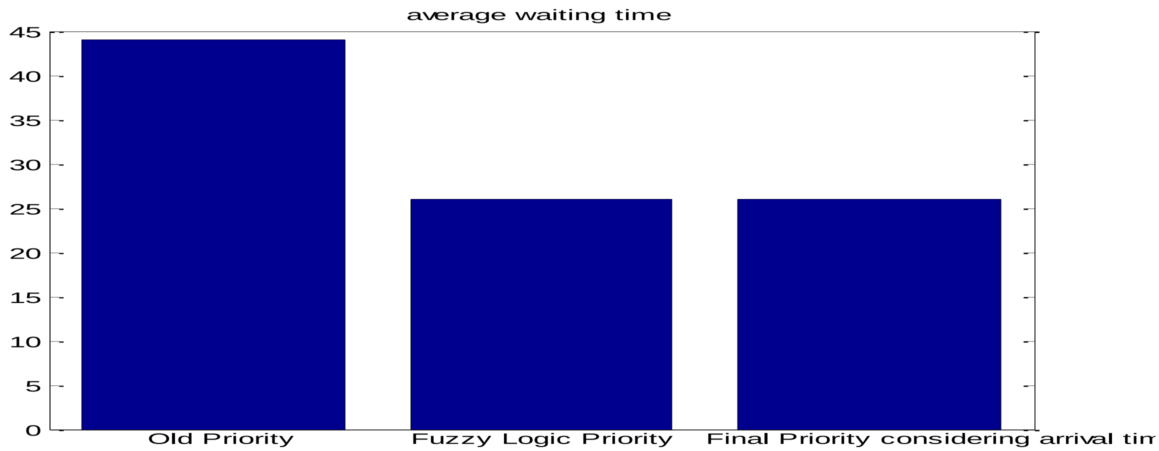


Figure 6 : Surface View of fuzzy rules set

Samiksha Deshwal *et al*, International Journal of Computer Science and Mobile Applications,
Vol.3 Issue. 6, June- 2015, pg. 66-79 ISSN: 2321-8363

ALGORITHM 1: Fuzzy based CPU scheduling

```
clear all
close all
clc
        % execution Time(s)   Priority  Arrival time
input=[       3                    6         2  ;        % process1
             24                    5         1  ;        % process2
              6                    1         2  ;        % process3
              9                    4         1  ;        % process4
              8                    2         0  ] ;      % process
% fuzzy('mlqdis2.fis')
a = readfis('mlqdis2.fis'); % read fuzzy logic rules set
gensurf(a)
title('3-D Surface View of rules set of FLC for Execution time & priority')
ruleview('mlqdis2.fis')
% priority calculated by fuzzy logic
for ii=1:size(input,1)
   temp=[input(ii,2) input(ii,1)];
   new_priority(ii)=evalfis(temp,a);
end
% calculate the priority based on arrival time. the lesser the arrival
% time, highest is the execution priority.
[arrival_time,ind]=sort(input(:,3));
for ii=1:size(input,1)
   new_arrival_priority(ii)=new_priority(ind(ii));
end
```



	Old Priority	Fuzzy Logic Priority	Final Priority considering arrival time too
1	6	7.6038	4.2361
2	5	2.6690	2.6690
3	1	5.4027	5.2474
4	4	5.2474	7.6038
5	2	4.2361	5.4027

Figure7: Numerical Table for priorities

IV. CONCLUSIONS

The proposed fuzzy based scheduling algorithm is an efficient scheduling algorithm that is obtained batter result rather than other algorithm. In multiple queues scheduling task is divided into multiple queues (MLQ) and each queue can use different scheduling algorithm like FCFS, round robin etc. It is the extended form of priority scheduling. The task with lower priority with small CPU time has always been assigned to lower level queues using these techniques; but handling their preciseness dynamically rather than fixing the queues on the basis of priority, this task may be assigned to higher queues and can improve the performance of scheduler. Although MLQ performs well than other techniques in case of stalling problem of tasks yet there is always room for modifications. The average waiting time and average turnaround time of proposed fuzzy based algorithm is much better than the Priority algorithm, Fuzzy based CPU scheduling algorithm and closer to obtain by SJF algorithm, but SJF algorithm doesn't deal with task priority. It shows the comparison between SJF, Fuzzy based CPU scheduling algorithm and proposed new fuzzy based scheduling algorithm. Results prove that algorithm proposed in this paper is much better than existing algorithms.

V. FUTURE SCOPE

The work presented in this paper can be expanded in many directions like:

1) Studying performance in real time applications where tasks have priorities and deadline constraints.

2) Applying scheduling technique on distributed systems.

3) Employing different performance criteria for comparison such as the turnaround time and response time.

4) Implemented on two parallel queues which run in backhand and forehand. To make this simulation environment MATLAB's parallel computing toolbox will be used and designed fuzzy logic rules will help to determine the new priority level in both queues.

REFERENCES

[1] Kadhim, Shatha J., and Kasim M. Al-Aubidy. "Design and Evaluation of a Fuzzy-Based CPU Scheduling Algorithm" Information Processing and Management (2010): 45-52.

[2] Tanenbaum, A. S. (2008). Modern Operating Systems (3rd ed.). Pearson Education, Inc. p. 156. ISBN 0-13-600663-9.

[3] Stallings, W.: Operating Systems Internals and Design Principles, 5th edn. Prentice-Hall, Englewood Cliffs (2004)

[4] Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J.: Scheduling Computer and Manufacturing Processes. Springer, Berlin (2001)

[5] Shahzad, B., Afzal, M.T.: Optimized Solution to Shortest Job First by Eliminating the Starvation. In: The 6th Jordanian Inr. Electri cal and Electronics Eng. Conference (JIEEEC 2006), Jordan (2006).

[6] Hiwarkar, Tryambak A., and R. Sridhar Iyer. "New Applications of Soft Computing, Artificial Intelligence, Fuzzy Logic & GeneticAlgorithm in Bioinformatics." (2013).

[7] Varma, K. Ajay, D. K. Mohanta, and M. J. B. Reddy. Applications of type-2 fuzzy logic in power systems A literature survey. Environment and Electrical Engineering (EEEIC), 2013 12th International Conference onIEEE2013

[8] Xia, Feng, et al. "Fuzzy logic based feedback scheduler for embedded control systems." Advances in Intelligent Computing. Springer Berlin Heidelberg, 2005. 453-462.

[9] Gomathy, C., and S. Shanmugavel. "An efficient fuzzy based priority scheduler for mobile ad hoc networks and performance analysis for various mobility models." Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE. Vol. 2. IEEE, 2004.

[10] Rajani Kumari et al.,"Air Conditioning System with Fuzzy Logic and Neuro-fuzzy Algorithm", Advances in Intelligent Systems and Computing, Springer, Vol. 236, ISBN 978-81-322-1601-8.(2013)

[11] Saleh, Ahmed I. "An efficient grid-scheduling strategy based on a fuzzy matchmaking approach." Soft Computing 17.3 (2013): 467-487.

[12] Salimi, Reza, Navid Bazrkar, and Mostafa Nemati. "Task Scheduling for Computational Grids Using NSGA II with Fuzzy Variance Based Crossover." Advances in Computing 3, no. 2 (2013): 22-29.

[13] He, Chuan, Dishan Qiu, and Hao Guo. "Solving Fuzzy Job Shop Scheduling Problem Based on Interval Number Theory." In Proceedings of the 2012 International Conference on Information Technology and Software Engineering, pp. 393-401. Springer Berlin Heidelberg, 2013.

[14] Mandloi, Saurabh, and Hitesh Gupta. "Adaptive job Scheduling for Computational Grid based on Ant Colony Optimization with Genetic Parameter Selection." International Journal (2013).

[15] Chang, Yi Chun, and Yao Tien Wang. "A Fuzzy-Based Dynamic Load Decision Making Scheme in Cloud Computing." Advanced Materials Research 718 (2013): 2191-2196.

[16] Yen, John, and Reza Langari. Fuzzy logic: intelligence, control, and information. Prentice-Hall, Inc.,

[17] S. Behera, Rakesh Mohanty, Jajnaseni Panda, Dipanwita Thakur, Subasini Sahoo “Experimental analysis of a new fair-share scheduling algorithm with waited time slice for real time systems”. Journal of Global Research in Computer Science (ISSN-2229-371X), Volume 2, No. 2, 54-60, February 2011.

[18] Al-Husainy, M.A.F.,”Best-job-first CPU scheduling algorithm” 2007, Inform. Technol. J.,Volume 6: Number 2, Pp: 288-293(pp) .