International Journal of Engineering Research & Management Technology ISSN: 2348-4039
May-2015 Volume 2, Issue-3
Email: editor@ijermt.org www.ijermt.org

FUZZY ANALYSIS OF SHORTEST JOB FIRST

Mukesh K. Sharma
Dept of Mathematics,
R.S.S. (PG) College,
Pilkhuwa, Hapur (INDIA)

Dr.Kapil Kumar Bansal
Head, Research & Publication
SRM University, NCR Campus
Modinagar, Ghaziabad

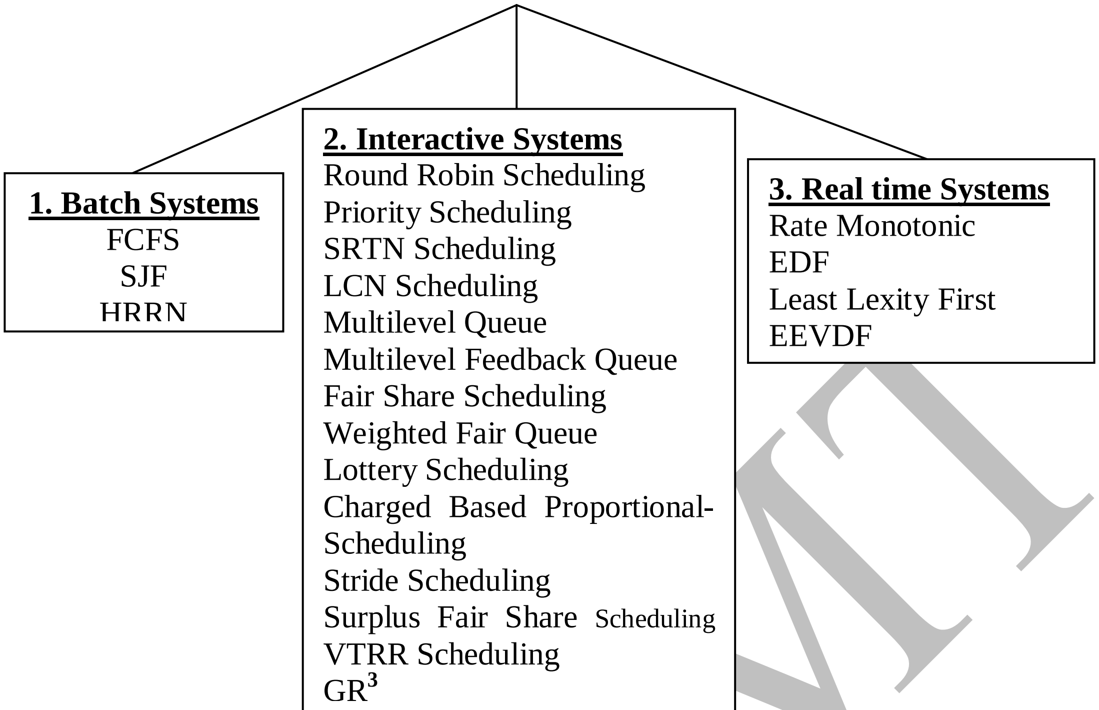
ABSTRACT:
Present paper discusses a new approach to SJF scheduling that takes into account the expected CPU time requirements of the processes along with the order of their arrivals. The basic approach of the proposed method is to select the first process and then use the fuzzy inference engine to obtain the first CPU burst. A fuzzy inference engine is also used to obtain the fuzzy estimate of a fixed CPU burst length.

KEYWORDS: Process, Dispatcher, Short- term scheduler, Starvation, PCB, Fuzzy-Inference Engine, Fuzzy SETS.

INTRODUCTION:
CPU scheduling is the activity of selecting the next request to be handled by the CPU. A request/process is considered as arrived, when the user submits it to Operating System (OS). The kernel of OS gathers information in respect of nature and size of the process and keeps record of its arrival time before sending it to the queue of pending ready requests. The role of a short term scheduler is to pick one process from the list of ready processes and hand it over to dispatching mechanism for execution on the CPU. The basic objective of the scheduler is to make effective use of the CPU to facilitate the processes in terms of the waiting time, turnaround time, priorities, starvation and throughput of the system. Several scheduling policies exist in the literature that benefits the waiting processes in one criterion or the other. The scheduling policy used in an OS influences user service, efficient use of resources and system performance. Scheduling policies use the fundamental techniques of preemption, reordering of requests and variation of time slice to achieve their goals. In the present research paper we have reformulated the formula $\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n$ of [Operating Systems Concepts by Galvin et. al Wiley India] next CPU burst by using fuzzy approach. A different approach to CPU Scheduling is the shortest job first scheduling (SJF) algorithm. This algorithm associates with each process the length of the latter's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the two processes have the same length next CPU burst then FCFS scheduling is used to break the tie. Note that a more appropriate term would be the shortest next CPU burst because the scheduling is done examining the length of the next CPU burst of a process rather than its total length.

AVAILABLE TECHNIQUES:
The whole class of scheduling algorithms can be basically categorized into three major Heads: scheduling Batch systems, Interactive Systems and Real time systems.

SCHEDULING ALGORITHMS:



1. ASSUMPTIONS:

1. Let N_0 be the initial approximation. First burst would be taken.
2. λ_i fuzzy set 'about t_1'
3. α is a fuzzy number and α =fuzzy set about ½ and $\alpha \epsilon [0 \;\; 1]$
4. Here, we will consider two aspect
   (i) Fuzzy numbers 'about ½'
   (ii) Fuzzy numbers 'about t_i'

2. FUZZY ANALYSIS:

Let N_0 be the estimated CPU burst time of the first arrived process and α ($o < \alpha \leq 1$) be any constant chosen in proportion of N_0, then

λ_i = defuzz ($[N_0 - 2\alpha, N_0 - \alpha, N_0] \cup [N_0 - \alpha, N_0, N_0 + \alpha] \cup [N_0, N_0 + \alpha, N_0 + 2\alpha]$)

where the expression within the parenthesis is the union of three triangular fuzzy numbers about $N_0 - \alpha$, N_0, and $N_0 + \alpha$ respectively. A triangular fuzzy number [a, b, c] is given by following membership function:

$$\mu_A(x) = \begin{cases} 0 & , \quad if \quad x \leq a \\ \dfrac{x-a}{b-a} & , \quad if \;\; a < x \leq b \\ \dfrac{c-x}{c-b} & , \quad if \; b \leq x < c \\ 0 & , \quad if \; x \geq c \end{cases}$$

3. PROPOSED ALGORITHM:

$$\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n$$
$$\tau_{n+1} = \alpha t_n + \tau_n - \alpha\tau_n$$
$$= \tau_n + \alpha(t_n - \tau_n)$$

Let $\epsilon_n = t_n - \tau_n$

Then

$$\tau_{n+1} = \alpha\epsilon_n + \tau_n$$

If we put $n = n-1$

$$\tau_n = \alpha t_{n-1} + (1-\alpha)\tau_{n-1}$$
$$= \alpha\, t_{n-1} - \alpha\,\tau_{n-1} + \tau_{n-1}$$
$$= \alpha[\, t_{n-1} - \tau_{n-1}] + \tau_{n-1}$$
$$= \alpha\ \epsilon_{n-1} + \tau_{n-1}$$

Where, $\epsilon_{n-1} = t_{n-1} - \tau_{n-1}$

Therefore,

$$\tau_{n+1} = \alpha\,\epsilon_n + \alpha\,\epsilon_{n-1} + \tau_{n-1}$$

Similarly, we can see that

$$\tau_{n+1} = \alpha\,\epsilon_n + \alpha\,\epsilon_{n-1} + \alpha\,\epsilon_{n-2} + \alpha\,\epsilon_{n-3} + \cdots \alpha\,\epsilon_1 + \tau_0$$

$$\tau_{n+1} = \sum_{k=0}^{n} \epsilon_k + \tau_0$$

Now using the fuzzy technique:

First we will consider the fuzzy set 'aboutt_1' and α is a fuzzy set 'about ½'.

Let $\widetilde{A}_1 = (\text{fuzzy set about } t_1)$

$\tau_1 = \widetilde{A}_1$

$\epsilon_1 = t_1 - \tau_1$

$\epsilon_1 = [\text{defuzz}((\widetilde{A}_1) - t_1]$

$\tau_2 = \alpha\epsilon_1 + \tau_1$

$\tau_2 = \alpha[\text{defuzz}(A_1) - t_1] + \tau_1$

$\epsilon_2 = [\text{defuzz}((\widetilde{A}_2) - t_2]$

$\tau_3 = \alpha\epsilon_2 + \tau_2$

$= \alpha[\text{defuzz}((\widetilde{A}_2) - t_2] + \alpha[\text{defuzz}(A_1) - t_1] + \tau_1$

Similarly, we can

$$\tau_{n+1} = \alpha[\text{defuzz}((\widetilde{A}_n) - t_n] + \alpha[\text{defuzz}(A_{n-1}) - t_{n-1}] \ldots\ldots + \ldots\ldots\ldots\ldots \tau_1$$

$$\tau_{n+1} = \alpha[\sum_{k=1}^{n}[\text{defuzz}(A_k) - t_k]] + \tau_1$$

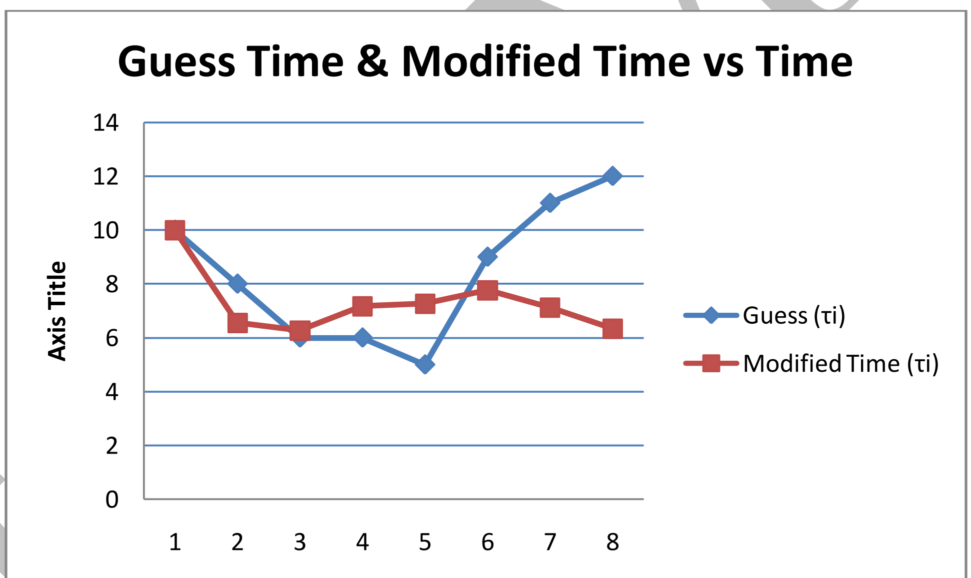
Or we can say that

$$\tau_{n+1} = \alpha[\text{defuzz}(\widetilde{A}_n) - t_n] + \tau_n$$

This will be equivalent to

$$\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n$$

CPU burst (ti)	Guess (τi)	Modified Time (τi)
0	10	0
6	8	6.5637
4	6	6.2772
6	6	7.1842
4	5	7.2744
13	9	7.7723
13	11	7.1365
13	12	6.3456



REFERENCES :

1. Umar Saleem and Muhammad Y. Javed, 'Simulation of CPU scheduling algorithms', Proceedings Tencom 2000, Vol2, pp 562-567 (2002).
2. Stuart E. Madnick and John J. Donovan, ' Operating Systems', Tata McGaw-Hill Edition 1997.
3. William Stallings, 'Operating Systems- Internals and Design Principles', Pearson Prentice Hall, 2006.
4. Al-Husainy Mohammed A.F., "Best-Job first CPU Scheduling Algorithm", Information Technology Journal 6(2), pp 288-293, **2007**.
5. Pandey D., Vandana and Sharma M.K., "CPU Scheduling: FCFS with Shorter Processes First", MR International Journal of Engineering and Technology, 1 (2), pp 11-17, **2008**.