

A New Intuitionistic Fuzzy Mamdani Inference System for Priority and Starvation in Scheduling Policy

Rajeev Sharma

Department of Computer Science, IIMT Engineering College, Meerut, India

Email: rajeev1418mtechcse@gmail.com

Atul Kumar Goel

Department of Mathematics, A.S. (P.G.) College, Mawana, Meerut, India

Email: atulgoel69@gmail.com

Dharamvir Singh

GBSSS MOLARBAND, New Delhi, India

Email: dharmvir.sgh@gmail.com

Mukesh Kumar Sharma

Department of Mathematics, Chaudhary Charan Singh University, Meerut, India

Email: drmukeshsharma@gmail.com

(Received June 22, 2024, Accepted August 12, 2024)

Abstract: Process scheduling algorithms make multiprogramming more productive by optimizing the utilization of the CPU. It is a very crucial task of scheduling algorithms that the CPU should not ideal if processes are in ready queue. We developed a unique algorithm for improving the process execution in real-time systems by utilizing the modified round robin (RR) algorithm variations that utilizes the use of the degree of multiprogramming. The developed algorithm removes the issue of starvation as compared with traditional priority scheduling algorithms. We also proposed an intuitionistic fuzzy inference system based MMFNP model to calculate the new priority that plays a key role for the execution of urgent tasks. Maximum value of New Priority represents the uppermost priority. The proposed algorithm increases the throughput of the system and reduces the average response time, average waiting time and average turnaround time. For the performance evaluation of the suggested algorithm, we also compared the results with RR algorithm and priority-based RR algorithm.

Keywords: Intuitionistic fuzzy inference system (IFIS), new priority (NP), mathematical model for new priority (MMFNP), starvation, real time system.

1. Introduction

Process scheduling is an important aspect in operating system (OS)¹, as it dictates that how the process scheduler fetches the processes from storage or from the job pool and assigned to the ready queue. A process is a program in execution. A process is carried out by one or more threads, each of them has its own set of program code and activity. The OS generated a new process from the execution of a program. The execution of a process involves numerous states, including "New," "Running," "Waiting," "Ready," and "Terminated," and all are finished by the process scheduling. Process scheduling make multiprogramming² more productive by maximizing the CPU utilization and controlling the degree of multiprogramming. According to the process scheduling, there are several processes in queue that are available. The management of process scheduling is done by process scheduler³. There are different types of process schedulers for the process scheduling and out of these each performed its task in the form of a decision maker. Long terms scheduler- It picked the processes from the pool (secondary memory) and retain them in the ready queue. Medium term scheduler- Medium term scheduler take care of the swapped-out processes. Short term scheduler- It fetched the processes from ready queue and allocate to the CPU for the execution. Scheduler used the scheduling algorithms for the decision making in selection of processes for execution. Scheduling algorithms are played a key role to performs this task⁴. Preemptive and Non-Preemptive are only two types of scheduling algorithms. Preemptive scheduling; When a process shifts from the running or waiting to the ready states, it is said to be preemptive scheduling⁵. Priority plays an important role in switching processes. Preemptive scheduling permits a scheduler to preempt a running low-priority process at any time when a high-priority process reached the ready state. Non-Preemptive scheduling; Non-preemptive Scheduling⁶ is a technique for CPU scheduling in which a process takes a resource and holds it, until the ends or switches to the waiting condition. It doesn't change the process's state from running to ready. Numerous preemptive and non-preemptive algorithms are existed for the execution of process on CPU. Some of them are: First Come First Serve (FCFS) scheduling; In this scheduling, execution of processes are based on the first in first out (FIFO) queue⁷. Processes are assigned to the CPU for execution as per their arrival in ready state. Interruption of a process cannot be done by another process during the execution. It is non-preemptive in nature. Shortest Job First (SJF) scheduling; in this scheduling, scheduler decide the selection of processes for execution as per their burst time. Minimum burst time process

is executed first other than the highest burst time process⁸. It has two versions; one is preemptive in nature and other is non-preemptive. Priority scheduling; it is most popular scheduling technic in batch systems⁹. Each process is executed as per their arrival time, less arrival time process is executed first but if the two or more process have same arrival time, then execution of process is done by their assigned priorities. It is also preemptive and non-preemptive in nature. Multilevel Feedback Queue Scheduling¹⁰; this scheduling is based on the strategy to use the multiple queues to assign the process as per their burst time. Higher burst time processes are placed in lower priority queue and lower burst time processes are placed in higher priority queue but if a process waiting for long time in lower priority queue, then it will transfer into higher priority queue. It is starvation free scheduling. Round robin scheduling¹¹; It is based on the principal of “Round robin” in which a time quantum is equally distributed to all processes to run on CPU. CPU is allocated to each process as per their distributed time quantum (TQ) for execution, if a process has higher burst time compare to assigned time quantum, then it will be pre-empted and move in waiting queue for the next cycle. Selection of TQ length is main challenge in this scheduling. It is similar as FCFS with TQ. Fuzzy round robin scheduling¹²; Fuzzy technic is used to find the appropriate length of TQ. Fuzzy rules are used with knowledge base system to calculate the TQ. B. Alam *et al*¹³ used the fuzzy inference system (FIS) to find the dynamic priority. FIS take three inputs as, static priority, remaining time, waiting time and produce the dynamic priority (dp) as output. The proposed approach improved traditional Priority Scheduling in terms of throughput and waiting time. P. Ajmani *et al*¹⁴ proposed a fuzzy logic-based algorithm. To calculate the dynamic priority (dp), burst time (bt), static priority (sp) and arrival time (at) as input to FIS. For the evaluation of the membership function of priority $\mu_p = sp / \max(sp) + 1$ and for burst time $\mu_b = 1 - (bt / \max(bt) + 1)$ are used. calculation of dp is based on: process with smallest priority used $dp = (\mu_p + \mu_b)$ otherwise used $dp = \max(\mu_p, \mu_b)$. “Design and Implementation of Modified Fuzzy based CPU Scheduling Algorithm” has been designed to improve the average waiting time (AWT) and average turnaround time (ATAT). A. Kumari *et al*¹⁵ designed new fuzzy rules with two membership function have used to calculate the value of new priority. Execution of process as per new priority increase system performance. X.Z. Gao *et al*¹⁶ described advanced Fuzzy Based CPU Scheduling (AFCS) algorithm uses an ideal time slice to execute processes and avoids process starvation using the deadline notion. B. Granam *et al*¹⁷ presented a new approach to classified the priority dynamically using the FIS. Proposed approach decided the execution of

process with an effective manner. R. Sharma *et al* ¹⁸ proposed a modified round robin algorithms using the approach fuzzy logic. FIS take two inputs as number of process and average burst time to calculate the optimal value of time quantum. Implementation of proposed algorithm got increased the throughput of system. M.A. Butt *et al* ¹⁹ proposed a novel intuitionistic fuzzy rule-based decision-making system based on IFS. Nice value (nv) and bt, two inputs intuitionistically fuzzified and triggered the fuzzy rule from rule base to calculate the fuzzy dynamic priority (dp). To defuzzified the fuzzy dp

$$x = \frac{\sum_{j=1}^M x^j ((1-\pi_A)\mu_{Aj} + \mu_{Aj}\pi_{Aj})}{\sum_{j=1}^M ((1-\pi_A)\mu_{Aj} + \mu_{Aj}\pi_{Aj})}$$
 method is used. S. Raheja *et al* ²⁰ introduced a new framework for the scheduling policy based on intuitionistic fuzzy sets. Intuitionistic fuzzy-based RR scheduler (IFRR) integrates the hybrid RR approach and IFIS. Dynamic time slice is produced by IFIS to prove the efficiency of scheduler. There are some limitations of Priority Round Robin scheduling algorithm, defined as follows:

- I. It is not appropriate for real-time systems due to its high context switching.
- II. Allocation of suitable priority for urgent tasks.
- III. Large response time, turnaround time and waiting time.
- IV. Uncertainty is only reflected by the membership function.
- V. Starvation is also issue in this scheduling.

To resolve these issues of Priority Round Robin scheduling, we proposed a new algorithm. The proposed policy removes the drawback of priority based round robin algorithm. This policy used the MMFNP model to calculate the new priorities which increase the performance of system. This strategy strikes a balance between fairness and efficiency, allowing the system to give priority to important operations while making sure every process receives an equal share of CPU time. This enables the system to concentrate on urgent or important activities and guarantee their quick and efficient completion.

The present research work is splitted into six segments. In the first segment, we gave the primary introduction, related work and limitations of proposed policy. Second segment explained the basics of Intuitionistic fuzzy set and Intuitionistic Fuzzy Inference Systems. We designed a new MMFNP model for calculating the new priority in third segment. In fourth segment, we proposed the basic structure of proposed algorithm, IFIS with its component, defuzzification process and final output. In fifth segment, we mentioned numerical computation and performance evaluation of algorithm. Result comparison of proposed algorithm with existing policy is also the part of this segment. Conclusion of entire work and future plan of the proposed work algorithm is also given in the sixth segment of the manuscript.

2. Intuitionistic fuzzy set (IFS)

Sets with elements that having varying degrees of membership and non-membership are referred to as intuitionistic fuzzy sets. Actually, the IFS is an extension of the fuzzy set described by Zadeh. In a conventional fuzzy inference system (FIS), using intuitionistic fuzzy set (IFS) should make it possible to handle more uncertainty. IFS contains the degrees of membership, hesitation, and non-membership. The degree of hesitation is a component of membership, non-membership, or both.

Let X be a non-empty set. An IFS M in X is defined as an object of the following form

$$M = \{(x, \mu_A(x), \nu_A(x)) : x \in X\}$$

where the function $\mu_A(x) : X \rightarrow [0,1]$ is the degree of membership and $\nu_A(x) : X \rightarrow [0,1]$ is the degree of non-membership of the element $x \in X$ to the set M and for every element $x \in X$:

$$0 \leq \{\mu_A(x) + \nu_A(x) \leq 1\}$$

As for the $\pi_A(x) = \{1 - \mu_A(x) - \nu_A(x)\}$ is called the degree of indeterminacy or hesitation margin of the element $x \in X$ and $\pi_A(x) \rightarrow [0,1]$. $\pi_A(x)$ shows ignorance on whether or not x is a member of IFS.

2.1. Intuitionistic Fuzzy Inference Systems (IFIS)

IFIS is also known as Intuitionistic Fuzzy-rule-based system⁷, one of the most widely used computing frameworks that uses fuzzy rules. The IFIS takes the crisp input, fuzzify it, processes it with a knowledge base to produce the fuzzy output, and then defuzzifies the result to produce the final crisp output. The process of mapping deterministic numerical data and qualitative description into IFS is called fuzzification. As an intuitionistic fuzzy set $\tilde{Z}$ with a membership function of $\mu_{\tilde{Z}}(p)$, a non-membership function of $\gamma_{\tilde{Z}}(p)$, and an IF-index of $\pi_{\tilde{Z}}(p)$, Fuzzifier executes interpretations of the rules. Output of an IFIS is possibly can be determined as follows

$$\alpha^\sigma = (1 - \pi_{\tilde{Z}}(p)) * \alpha^n + \pi_{\tilde{Z}}(p) \alpha^\gamma$$

where α^n is the output of FIS^n using the membership function of $\mu_{\tilde{Z}}(p)$ and α^γ is the output of FIS^γ using the non-membership function $\gamma_{\tilde{Z}}(p)$.

3. Mathematical model for new priority (MMFNP)

Let $p_1, p_2, p_3, \dots, p_n$ are the static priorities of the processes which are arranged as per arrival time in queue and $b_1, b_2, b_3, \dots, b_n$ are expected

average burst time (ABT) of processes. Z is the new priority for individual process to run on CPU. All are given in triangular intuitionistic fuzzy numbers. Membership function of priority $p = \langle e, f, g \rangle$; $e < f < g \in \mathbb{R}$ and non-membership function of priority $p^* = \langle e', f', g' \rangle$; $e' < f' < g' \in \mathbb{R}$. Then membership function of priority $P_1 = \langle e_1, f_1, g_1 \rangle$; $e_1 < f_1 < g_1 \in \mathbb{R}$

$$P_1(\gamma) = \begin{cases} \frac{\gamma - e_1}{f_1 - e_1} & \text{if } e_1 \leq \gamma < f_1 \\ \frac{g_1 - \gamma}{g_1 - f_1} & \text{if } f_1 \leq \gamma \leq g_1 \\ 0 & \text{else} \end{cases}$$

and non-membership function of priority $P_1^* = \langle e'_1, f'_1, g'_1 \rangle$; $e'_1 < f'_1 < g'_1 \in \mathbb{R}$

$$P_1^*(\gamma') = \begin{cases} \frac{\gamma' - e'_1}{f'_1 - e'_1} & \text{if } e'_1 \leq \gamma' < f'_1 \\ \frac{g'_1 - \gamma'}{g'_1 - f'_1} & \text{if } f'_1 \leq \gamma' \leq g'_1 \\ 0 & \text{else} \end{cases}$$

Membership function of priority $P_2 = \langle e_2, f_2, g_2 \rangle$; $e_2 < f_2 < g_2 \in \mathbb{R}$

$$P_2(\gamma) = \begin{cases} \frac{\gamma - e_2}{f_2 - e_2} & \text{if } e_2 \leq \gamma < f_2 \\ \frac{g_2 - \gamma}{g_2 - f_2} & \text{if } f_2 \leq \gamma \leq g_2 \\ 0 & \text{else} \end{cases}$$

and non-membership function of priority $P_2^* = \langle e'_2, f'_2, g'_2 \rangle$; $e'_2 < f'_2 < g'_2 \in \mathbb{R}$

$$P_2^*(\gamma') = \begin{cases} \frac{\gamma' - e'_2}{f'_2 - e'_2} & \text{if } e'_2 \leq \gamma' < f'_2 \\ \frac{g'_2 - \gamma'}{g'_2 - f'_2} & \text{if } f'_2 \leq \gamma' \leq g'_2 \\ 0 & \text{else} \end{cases}$$

Similarly for the n^{th} process we have

Membership function of priority $P_n = \langle e_n, f_n, g_n \rangle$; $e_n < f_n < g_n \in \mathbb{R}$

$$P_n(\gamma) = \begin{cases} \frac{\gamma - e_n}{f_n - e_n} & \text{if } e_n \leq \gamma < f_n \\ \frac{g_n - \gamma}{g_n - f_n} & \text{if } f_n \leq \gamma \leq g_n \\ 0 & \text{else} \end{cases}$$

and non-membership function of priority $P_n^* = \langle e'_n, f'_n, g'_n \rangle$; $e'_n < f'_n < g'_n \in \mathbb{R}$

$$P_n^*(\gamma') = \begin{cases} \frac{\gamma' - e'_n}{f'_n - e'_n} & \text{if } e'_n \leq \gamma' < f'_n \\ \frac{g'_n - \gamma'}{g'_n - f'_n} & \text{if } f'_n \leq \gamma' \leq g'_n \\ 0 & \text{else} \end{cases}$$

Membership function of ABT $b = \langle m, n, o \rangle$; $m < n < o \in \mathbb{R}$, and non-membership function of priority $b^* = \langle m', n', o' \rangle$; $m' < n' < o' \in \mathbb{R}$. Then Membership function of ABT $b_1 = \langle m_1, n_1, o_1 \rangle$; $m_1 < n_1 < o_1 \in \mathbb{R}$

$$b_1(\gamma) = \begin{cases} \frac{\gamma - m_1}{n_1 - m_1} & \text{if } m_1 \leq \gamma < n_1 \\ \frac{o_1 - \gamma}{o_1 - n_1} & \text{if } n_1 \leq \gamma \leq o_1 \\ 0 & \text{else} \end{cases}$$

And non-membership function of priority $b_1^* = \langle m'_1, n'_1, o'_1 \rangle$; $m'_1 < n'_1 < o'_1 \in \mathbb{R}$

$$b_1^*(\gamma') = \begin{cases} \frac{\gamma' - m'_1}{n'_1 - m'_1} & \text{if } m'_1 \leq \gamma' < n'_1 \\ \frac{o'_1 - \gamma'}{o'_1 - n'_1} & \text{if } n'_1 \leq \gamma' \leq o'_1 \\ 0 & \text{else} \end{cases}$$

Membership function of ABT $b_2 = \langle m_2, n_2, o_2 \rangle$; $m_2 < n_2 < o_2 \in \mathbb{R}$

$$b_2(\gamma) = \begin{cases} \frac{\gamma - m_2}{n_2 - m_2} & \text{if } m_2 \leq \gamma < n_2 \\ \frac{o_2 - \gamma}{o_2 - n_2} & \text{if } n_2 \leq \gamma \leq o_2 \\ 0 & \text{else} \end{cases}$$

And non-membership function of priority $b_2^* = \langle m'_2, n'_2, o'_2 \rangle$; $m'_2 < n'_2 < o'_2 \in \mathbb{R}$

$$b_2^*(\gamma') = \begin{cases} \frac{\gamma' - m'_2}{n'_2 - m'_2} & \text{if } m'_2 \leq \gamma' < n'_2 \\ \frac{o'_2 - \gamma'}{o'_2 - n'_2} & \text{if } n'_2 \leq \gamma' \leq o'_2 \\ 0 & \text{else} \end{cases}$$

Similarly for the n^{th} average burst time we have

Membership function of ABT $b_n = \langle m_n, n_n, o_n \rangle$; $m_n < n_n < o_n \in \mathbb{R}$

$$b_n(\gamma) = \begin{cases} \frac{\gamma - m_n}{n_n - m_n} & \text{if } m_n \leq \gamma < n_n \\ \frac{o_n - \gamma}{o_n - n_n} & \text{if } n_n \leq \gamma \leq o_n \\ 0 & \text{else} \end{cases}$$

And non-membership function of priority $b_n^* = \langle m'_n, n'_n, o'_n \rangle$; $m'_n < n'_n < o'_n \in \mathbb{R}$

$$b_n^*(\gamma') = \begin{cases} \frac{\gamma' - m'_n}{n'_n - m'_n} & \text{if } m'_n \leq \gamma' < n'_n \\ \frac{o'_n - \gamma'}{o'_n - n'_n} & \text{if } n'_n \leq \gamma' \leq o'_n \\ 0 & \text{else} \end{cases}$$

If I_1 is P_1 and I_2 is b_1 then y_1 is Z_1

If I_1 is P_1^* and I_2 is b_1^* then y_1 is Z_1^*

If I_1 is P_2 and I_2 is b_2 then y_2 is Z_2

If I_1 is P_2^* and I_2 is b_2^* then y_1 is Z_2^*

If I_1 is P_3 and I_2 is b_3 then y_3 is Z_3

If I_1 is P_3^* and I_2 is b_3^* then y_1 is Z_3^*

For n^{th} process, we have the fuzzy rule as

If I_1 is P_n and I_2 is b_n then y_n is Z_n

If I_1 is P_n^* and I_2 is b_n^* then y_1 is Z_n^*

Here I_1 = static priority, given to P_i

I_2 = ABT, given to b_i

New priority (Z) = {low, medium, high}

Membership function of $Z_{low} = \langle \alpha_1, \alpha_2, \alpha_3 \rangle$; $\alpha_1 < \alpha_2 < \alpha_3 \in \mathbb{R}$

$$Z_{low}(\gamma) = \begin{cases} \frac{\gamma - \alpha_1}{\alpha_2 - \alpha_1} & \text{if } \alpha_1 \leq \gamma < \alpha_2 \\ \frac{\alpha_3 - \gamma}{\alpha_3 - \alpha_2} & \text{if } \alpha_2 \leq \gamma \leq \alpha_3 \\ 0 & \text{else} \end{cases}$$

And non-membership function of $Z_{low}^* = \langle \alpha'_1, \alpha'_2, \alpha'_3 \rangle$; $\alpha'_1 < \alpha'_2 < \alpha'_3 \in \mathbb{R}$

$$Z_{low}^*(\gamma') = \begin{cases} \frac{\gamma' - \alpha'_1}{\alpha'_1 - \alpha'_2} & \text{if } \alpha'_1 \leq \gamma' < \alpha'_2 \\ \frac{\alpha'_3 - \gamma'}{\alpha'_3 - \alpha'_2} & \text{if } \alpha'_2 \leq \gamma' \leq \alpha'_3 \\ 0 & \text{else} \end{cases}$$

Membership function of $Z_{mid} = \langle \alpha_4, \alpha_5, \alpha_6 \rangle$; $\alpha_4 < \alpha_5 < \alpha_6 \in \mathbb{R}$

$$Z_{mid}(\gamma) = \begin{cases} \frac{\gamma - \alpha_4}{\alpha_5 - \alpha_4} & \text{if } \alpha_4 \leq \gamma < \alpha_5 \\ \frac{\alpha_6 - \gamma}{\alpha_6 - \alpha_5} & \text{if } \alpha_5 \leq \gamma \leq \alpha_6 \\ 0 & \text{else} \end{cases}$$

And non-membership function of $Z_{mid}^* = \langle \alpha'_4, \alpha'_5, \alpha'_6 \rangle$; $\alpha'_4 < \alpha'_5 < \alpha'_6 \in \mathbb{R}$

$$Z_{mid}^*(\gamma') = \begin{cases} \frac{\gamma' - \alpha'_4}{\alpha'_5 - \alpha'_4} & \text{if } \alpha'_4 \leq \gamma' < \alpha'_5 \\ \frac{\alpha'_6 - \gamma'}{\alpha'_6 - \alpha'_5} & \text{if } \alpha'_5 \leq \gamma' \leq \alpha'_6 \\ 0 & \text{else} \end{cases}$$

Membership function of $Z_{high} = \langle \alpha_7, \alpha_8, \alpha_9 \rangle$; $\alpha_7 < \alpha_8 < \alpha_9 \in \mathbb{R}$

$$Z_{high}(\gamma) = \begin{cases} \frac{\gamma - \alpha_7}{\alpha_8 - \alpha_7} & \text{if } \alpha_7 \leq \gamma < \alpha_8 \\ \frac{\alpha_9 - \gamma}{\alpha_9 - \alpha_8} & \text{if } \alpha_8 \leq \gamma \leq \alpha_9 \\ 0 & \text{else} \end{cases}$$

And non-membership function of $Z_{high}^* = \langle \alpha'_7, \alpha'_8, \alpha'_9 \rangle$; $\alpha'_7 < \alpha'_8 < \alpha'_9 \in \mathbb{R}$

$$Z_{high}^*(\gamma') = \begin{cases} \frac{\gamma' - \alpha'_7}{\alpha'_8 - \alpha'_7} & \text{if } \alpha'_7 \leq \gamma' < \alpha'_8 \\ \frac{\alpha'_9 - \gamma'}{\alpha'_9 - \alpha'_8} & \text{if } \alpha'_8 \leq \gamma' \leq \alpha'_9 \\ 0 & \text{else} \end{cases}$$

Let Q_1 = defuzzified value (Z_1) and Q_1^* = defuzzified value (Z_1^*)

Q_2 = defuzzified value (Z_2) and Q_2^* = defuzzified value (Z_2^*)

Q_3 = defuzzified value (Z_3) and Q_3^* = defuzzified value (Z_3^*)

Q_4 = defuzzified value (Z_4) and Q_4^* = defuzzified value (Z_4^*)

The nth defuzzified value (Z_n) and (Z_n^*), we have

Q_n = defuzzified value (Z_n) and Q_n^* = defuzzified value (Z_n^*)

So $y_1 = (1 - \pi) * Q_1 + \pi * Q_1^*$

$y_2 = (1 - \pi) * Q_2 + \pi * Q_2^*$

$y_3 = (1 - \pi) * Q_3 + \pi * Q_3^*$

$y_4 = (1 - \pi) * Q_4 + \pi * Q_4^*$

Similarly, for y_n , we have

$y_n = (1 - \pi) * Q_n + \pi * Q_n^*$

New priority = $\frac{y_1 + y_2 + y_3 + y_4 \dots \dots \dots y_n}{n}$

4. Basic structure and algorithm

The basic structure of our proposed algorithm is described in Figure-1. Two inputs (priority and ABT) are intuitionistically fuzzified. The rules in the rule base are then activated by these two input values for each process, and as a result, we obtain the defuzzified new priority for each process.

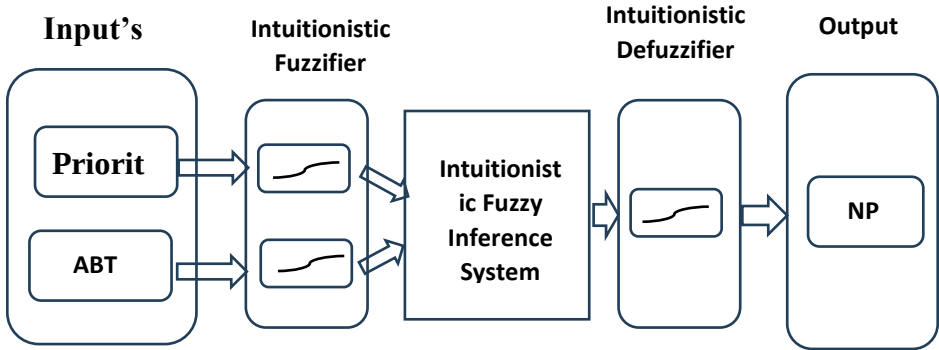


Figure 1: structure of fuzzification and defuzzification

Algorithm

- 1.) Design the Intuitionistic fuzzy inference system (IFIS) and use the MMFNP model to generate the new priority.
- 2.) Initialize n process with new priority, burst time and arrival time (AT).
- 3.) Assigned the processes to ready queue.
- 4.) If $(AT == 0) \forall$ process.
Go to step 5 otherwise go to step 6
- 5.) Choose the process from RQ with the highest NP and execute as:
While $(\frac{P_i}{n} == 1)$, for $i=1,2,3,4,\dots,n$, where p =process and n= total no. of process

```

{
    if  $(BT_i \leq TQ)$ 
    {
         $TQ = BT_i$ 
         $BT_{new}(P_i) = BT_i - TQ$  and terminate the process.
    }
    Else
    {
         $BT_{new}(P) = \begin{cases} 0, & \text{if } BT - TQ < 0 \\ BT - TQ, & \text{otherwise} \end{cases}$ 
         $NP_{new} = \begin{cases} NP - 1, & \text{if } NP > 1 \\ NP, & \text{otherwise} \end{cases}$ 
    }
}
    
```

}
Select the next process from RQ.
}

- 6.) Select the process according to minimum AT form RQ and execute as per step 5.
- 7.) Now check, if $(AT \forall \text{ process} < TQ)$ then repeat the step 5 otherwise repeat the step 6.
- 8.) Repeat the steps from 4 to 7.

4.1. Components of IFIS for fuzzification and defuzzification

The system consists of three primary components: an intuitionistic fuzzifier; an intuitionistic fuzzy rule base and an intuitionistic defuzzifier (ID)²², played a pivot role in fuzzification and defuzzification for the new priority. We used Mamdani method in our proposed IFIS for find the new priority to execute the process on processor.

4.1.(a) intuitionistic fuzzifier

intuitionistic fuzzifier is used to fuzzification of priority and burst time of the processes to calculate the NP so we divided the priority into three linguistic terms as “low”, “medium”, “high” and for burst time; “short”, “mid”, “long”, with triangular membership and non-membership function. The priority and burst time are the two inputs used in this component, and both are converted to intuitionistic fuzzy values by determining the membership and non-membership function²³. Figures-2 show the membership function (MF), non-membership function (NMF), respectively for a process' priority and process burst time.

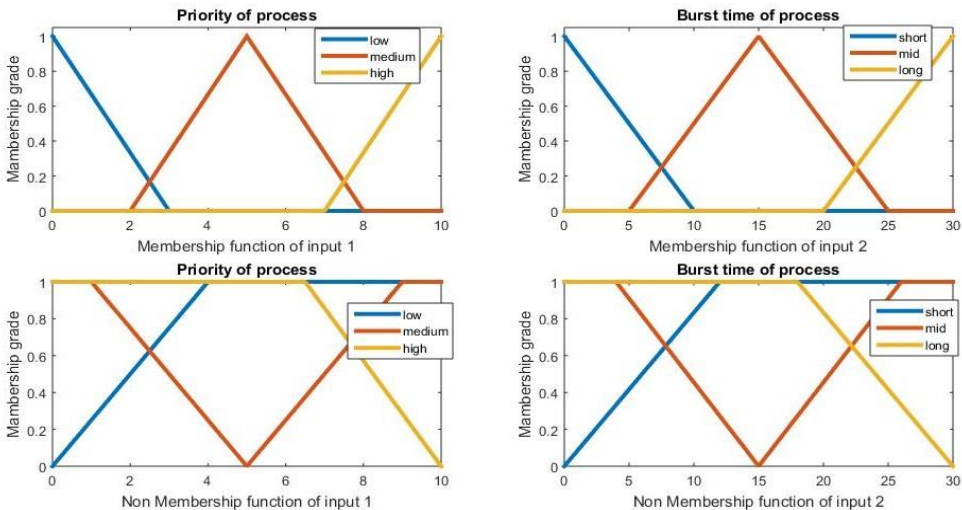


Figure 2: MF and NMF for priority, burst time.

Output (NP) is also categorised into “low”, “medium”, “high” linguistic terms with triangular membership and non-membership function. The output parameter of our intuitionistic decision-making system is depicted in Figures-3 with the MF and NMF. New priority is the output of decision-making system.

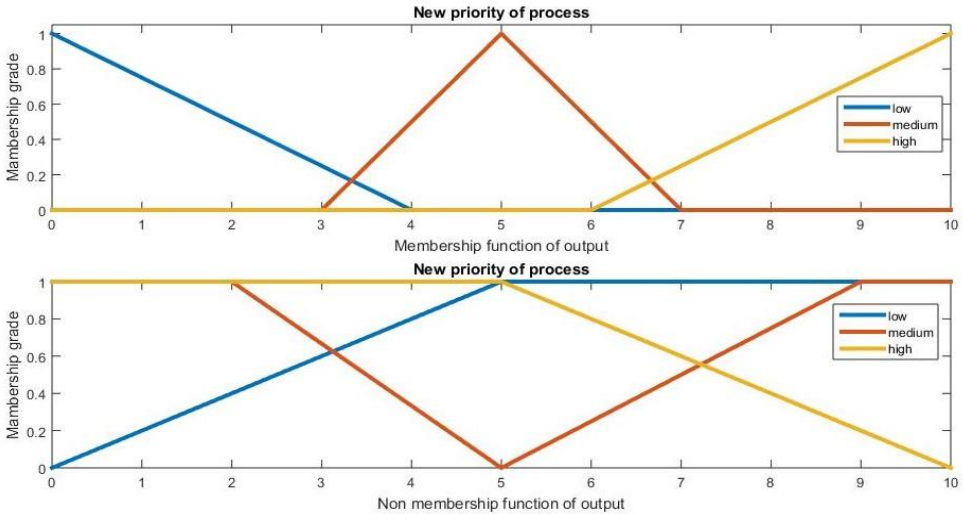


Figure 3: MF and NMF for new priority

4.1.(b) Intuitionistic fuzzy rule base

This component used the IF-THEN rules with knowledge base. In this, we have taken two input and one output based on intuitionistic fuzzy rules. The rules are made according to expert knowledge. To manage the antecedent part of the rules, AND operator is used and consequent part is represented the NP. Nine rules are used for rule base:

- R_1 : if (priority is low) and (burst time is short) then (NP is high)
- R_2 : if (priority is medium) and (burst time is short) then (NP is high)
- R_3 : if (priority is high) and (burst time is short) then (NP is high)
- R_4 : if (priority is low) and (burst time is mid) then (NP is medium)
- R_5 : if (priority is medium) and (burst time is mid) then (NP is medium)
- R_6 : if (priority is high) and (burst time is mid) then (NP is medium)
- R_7 : if (priority is low) and (burst time is long) then (NP is low)
- R_8 : if (priority is medium) and (burst time is long) then (NP is low)
- R_9 : if (priority is high) and (burst time is long) then (NP is medium)

4.1.(c) Defuzzifier

The defuzzifier element take the fuzzy NP of each input and defuzzified it into the crisp new priority for membership and non-membership function. We have used the area of centroid method for defuzzification²⁴ process

$$\mu^* = \frac{\sum_{i=1}^n A_i * x_i}{\sum_{i=1}^n A_i}$$

where A_i represent the distributed area of aggregated consequent part of fuzzy rules for membership function.

Similarly, we can define the method for non-membership function

$$v^* = \frac{\sum_{i=1}^n \bar{A}_i * \bar{x}_i}{\sum_{i=1}^n \bar{A}_i}$$

Where $\bar{A}_i$ represent the distributed area of aggregated consequent part of fuzzy rules for non-membership function.

4.1.(d) Final output

The final output of can be IFIS can be calculated by the following expression

$$y = (1 - \pi) * \mu_{(A)}^* + \pi * v_n^*$$

where $\mu_{(A)}^*$ represent the defuzzified value of membership function and v_n^* represent the defuzzified value of membership function (both are calculated from previous step).

5. Numerical computations and performance of modified priority round robin algorithm

We created the MMFNP model to provide the appropriate value of new priority. Intuitionistic Fuzzy Inference Systems is used priority and burst time as inputs and new priority as the output. To assess the efficiency and effectiveness of an intuitionistic fuzzy-based scheduling method, we taken diverse case studies and comparison some scheduling algorithms like Priority algorithm, priority based round robin and proposed intuitionistic fuzzy based scheduling algorithm.

5.1. Numerical Computation 1

In this study, we have used priority and burst time as inputs in MMFNP model to produce the new priority. Table-1 is displaying the process, arrival time, burst time, priority and new priority (outcome of MMFNP).

Table-1: data set-1

Process	j_1	j_2	j_3	j_4	j_5
Arrival time	0	2	0	4	5
Burst time	16	9	12	26	29
Priority	6	1	5	7	4
New priority	4.6	7.9	5.1	1.9	1.6

We have applied the proposed algorithm on data in the table. The proposed algorithm used NP and fixed TQ = 17 to execute the processes. Gantt chart-1 shows the execution procedure of processes. Table-2 compared the performance of proposed algorithm with traditional round robin and priority-based round robin. The performance of algorithms is evaluated based on average response time (ART), ATAT and AWT.

Gantt chart 1 for execution of processes

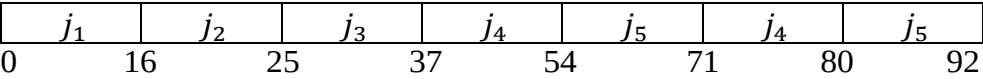


Table-2: Comparison of scheduling policy’s

Scheduling policy’s	ART	ATAT	AWT
Round Robin	24.8	48.4	30
Priority based Round Rot	29	58.6	40.2
Proposed scheme	24.2	47.8	29.2

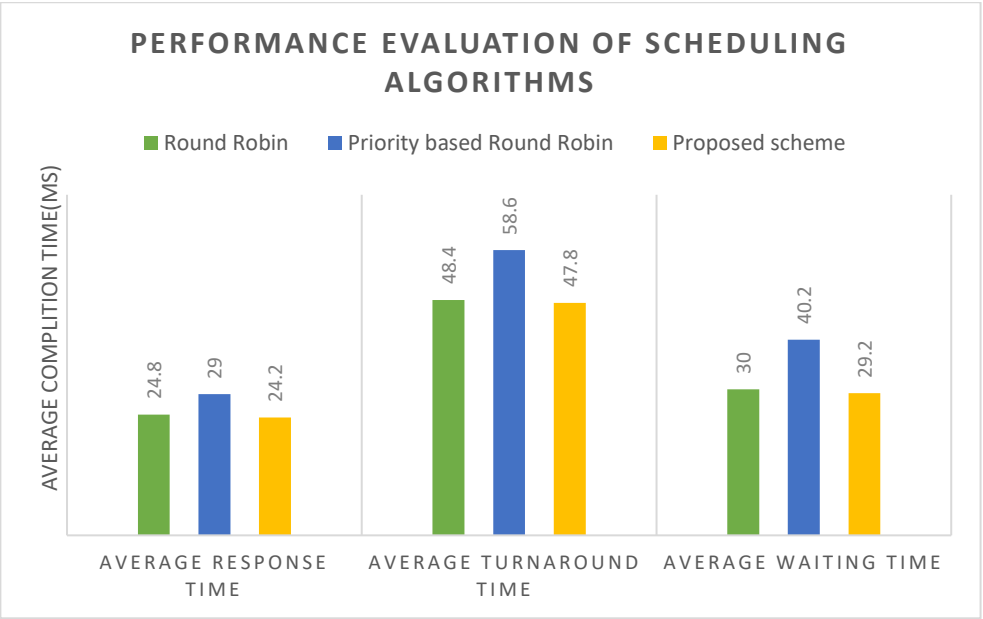


Figure 4: Performance evaluation of various algorithms

The proposed method performed well than the earlier algorithms in terms of response time, turnaround time, and waiting durations, as shown in figure 4.0.

5.2. Numerical Computation 2

In numerical computation 2, the arrival time of all processes = 0 and the TQ = 3. Table-3 contained the five different processes with different burst time, priority and new priority. Here new priority is the outcome of MMFNP

model. After applied the proposed algorithm, we got ART= 6, ATAT= 28.2 and AWT= 18.2 which are less compare to priority based round robin and traditional round robin. The suggested algorithm's performance is compared in the table 4 with the results of traditional round robin and priority-based round robin. Gantt chart 2 show the execution of processes.

Table 3: data set-2

Process	j_1	j_2	j_3	j_4	j_5
Burst time	3	24	6	9	8
Priority	6	5	1	4	2
New priority	8.5	4.8	4.8	7.9	8.1

Gantt chart 2 for execution of processes

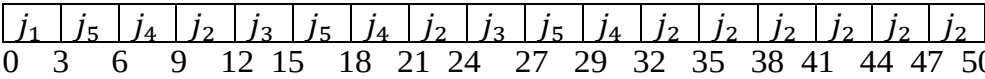


Table 4: Comparison of scheduling policy's

Scheduling policy's	ART	ATAT	AWT
Round Robin	6	28.9	18.8
Priority based Round Robin	6	29.6	19.6
Proposed scheme	6	28.2	18.2

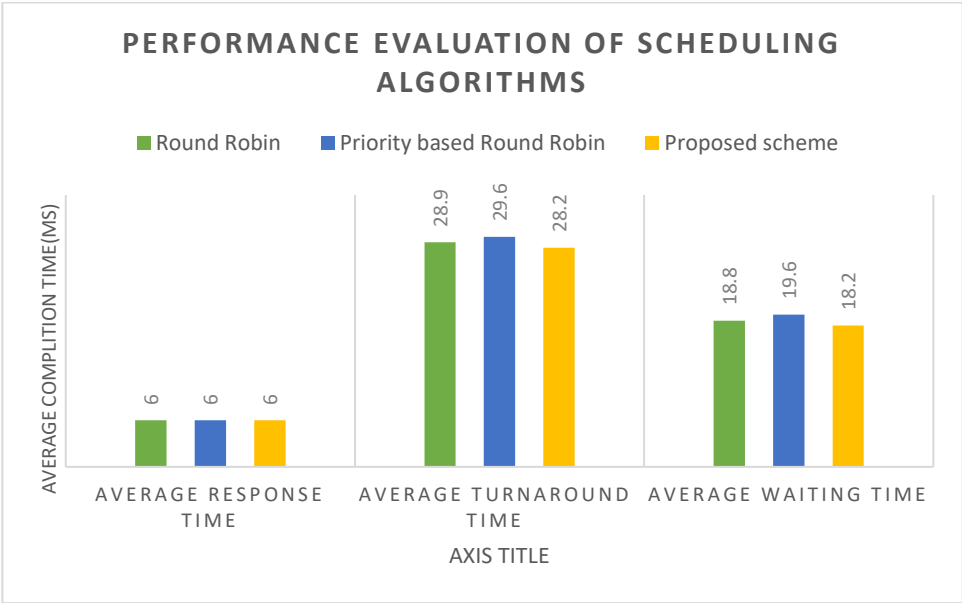


Figure 5: Performance evaluation of various algorithms

So, the proposed approach performs better than the previous algorithms in terms of response time, turnaround time, and waiting durations, as shown in Figure 5.

6. Conclusion and Future Work

The proposed algorithm combined the Round Robin scheduling and Priority scheduling techniques. The proposed algorithm used the MMFNP model to produce the new priority. MMFNP model is based on intuitionistic fuzzy sets. New priority is assigned to each process for the execution which guarantees that the crucial processes are finished quickly and in accordance with deadlines. The proposed algorithm assists the system to increase the throughput of the system. Full study of this research work is illustrated by following points:

- 1) We have used the MMFNP model to calculate the new priority.
- 2) IFIS play a key role to form the MMFNP model.
- 3) Implementation of proposed algorithm, decrease the response time, turnaround time and waiting time.
- 4) Starvation is very crucial issue in priority-based algorithms but proposed algorithm removes the starvation.
- 5) Proposed algorithm can also be used in real time systems.
- 6) Comparison result of this algorithm more batter than the other scheduling algorithms.

Our proposed algorithm can be better in future by calculating the dynamic TQ for each execution cycle. For more appropriate value of priority, we can use the neuro fuzzy inference system. In a nutshell we can say the proposed algorithm creates new path merging it with other algorithms.

References

1. J. N. Gray: Notes on data base operating systems, *Operating systems: Lecture Notes in Computer Science*, Springer, Berlin-Heidelberg, 1978.
https://doi.org/10.1007/3-540-08755-9_9.
2. A. S. Tanenbaum: *Modern operating systems*, China-Pub-Com, 2002. ISBN: 7-111-09156-6.
3. W. Stallings: *Operating Systems: Internals and Design Principles*, Pearson Education India, 2009. ISBN:978-81-317-2528-3.
4. N. Goel and R. B. Garg; A comparative study of cpu scheduling algorithms, arXiv (2013). arXiv:1307.4165. <https://doi.org/10.48550/arXiv.1307.4165>.
5. S. B. Bandrupalli, N. P. Nutulapati and P.S. Varma; A Novel CPU Scheduling Algorithm-Preemptive & Non-Preemptive, *International Journal of Modern Engineering Research*, 2-6 (2012), 4484-4490. ISSN: 2249-6645.
6. W. Li, K. Kavi and R. Akl; A non-preemptive scheduling algorithm for soft real-time systems, *Computers & Electrical Engineering*, 33(1) (2007), 12-29.

<https://doi.org/10.1016/j.compeleceng.2006.04.002>.

7. G. D. Bibu and G. C. Nwankwo; Comparative analysis between first-come-first-serve (FCFS) and shortest-job-first (SJF) scheduling algorithms, *International Journal of Computer Science and Mobile Computing*, 8-5 (2019), 176-181. ISSN: 2320-088X.
8. M. Hamayun and H. Khurshid; An optimized shortest job first scheduling algorithm for CPU scheduling, *J. Appl. Environ. Biol. Sci*, 5-12 (2015), 42-46. ISSN: 2090-4274.
9. Z. Lee, Y. Wang and W. Zhou; A dynamic priority scheduling algorithm on service request scheduling in cloud computing. *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, 9 (2011, August), 4665-4669.
10. M. Thombare, R. Sukhwani, P. Shah, S. Chaudhari and P. Raundale; Efficient implementation of multilevel feedback queue scheduling. *IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (2016, March) 1950-1954.
DOI: 10.1109/WiSPNET.2016.7566483.
11. A. Singh, P. Goyal and S. Batra; An optimized round robin scheduling algorithm for CPU scheduling, *International Journal on Computer Science and Engineering*, 2(7) (2010), 2383-2385. ISSN: 0975-3397.
12. B. Alam; Fuzzy Round Robin CPU Scheduling Algorithm, *J. Comput. Sci.*, 9(8) (2013), 1079-1085. doi:10.3844/jcssp.2013.
13. B. Alam, M. N. Doja, R. Biswas and M. Alam; Fuzzy priority CPU scheduling algorithm, *International Journal of Computer Science Issues (IJCSI)*, 8(6) (2011), 386.
14. P. Ajmani and M. Sethi; Proposed fuzzy CPU scheduling algorithm (PFCS) for real time operating systems, BVICA M's *International Journal of Information Technology*, 5(2) (2013), 583.
15. R. Kumari, V. K. Sharma and S. Kumar; Design and implementation of modified fuzzy based CPU scheduling algorithm. (2017), arXiv:1706.02621.
16. X.Z. Gao, S. Gurumurthy and S. Venkatesan; Improved CPU Utilization using Advanced Fuzzy Based CPU Scheduling algorithm (AFCS), *International Journal of Electrical Sciences & Engineering (IJESE)*, 1(1) (2015), 1-5.
17. B. Granam and H. ElAarag; Utilization of Fuzzy Logic in CPU Scheduling in Various Computing Environments, *Proceedings of the 2019 ACM Southeast Conference* (2019, April), 222-225.
18. R. Sharma, A. K. Goel, M. K. Sharma, N. Dhiman and V. N. Mishra; Modified Round Robin CPU Scheduling: A Fuzzy Logic-Based Approach, *Applications of Operational Research in Business and Industries: Proceedings of 54th Annual Conference of ORSI* (2023, May), 367-383.
19. M.A. Butt and M. Akram; A new intuitionistic fuzzy rule-based decision-

- making system for an operating system process scheduler, *SpringerPlus*, 5 (2016), 1-17.
20. S. Raheja, M. Alshehri, A. A. Mohamed, S. Khaitan, M. Kumar and T. Stephan A smart intuitionistic fuzzy-based framework for round-robin short-term scheduler, *The Journal of Supercomputing*, (2022), 1-25.
21. P.A. Ejegwa, S.O. AkoweOtene, P. M., & Ikyule, J. M. (2014). An overview on intuitionistic fuzzy sets. *Int. J. Sci. Technol. Res*, 3(3), 142-145. ISSN 2277-8616.
22. S.K. De, R. Biswas and A.R. Roy: Some operations on intuitionistic fuzzy sets, *Fuzzy sets and Systems*, 114(3) (2000), 477-484.
[https://doi.org/10.1016/S0165-0114\(98\)00191-2](https://doi.org/10.1016/S0165-0114(98)00191-2).
23. M.A. Butt and M. Akram; A new intuitionistic fuzzy rule-based decision-making system for an operating system process scheduler, *SpringerPlus*, 5 (2016), 1-17. <https://doi.org/10.1186/s40064-016-3216-z>.
24. K.T. Atanassov; *On intuitionistic fuzzy sets theory*, Springer, 2012. ISBN: 1434-9922.
25. T. Mitsuishi; Definition of Centroid Method as Defuzzification, *Formalized Mathematics*, 30(2) (2022), 125-134. DOI: <https://doi.org/10.2478/forma-2022-0010>.