The International Journal of Interpretation, Observation and Analysis 2025;4,1 (October-December) *IJIOA*

TASK SCHEDULING FOR DISTRIBUTED REAL-TIME SYSTEMS IN A FUZZY ENVIRONMENT

[1] Soniya Saini, [2]Dr. Sandeep Kumar Tiwari and [3]Dr. Ankur Nehra
[1]Research Scholar, Department of Mathematics, Faculty of Science, Motherhood University, Haridwar, Uttarakhand, 24766, India
[2]Supervisor, Department of Mathematics, Faculty of Science, Motherhood University, Roorkee, Uttarakhand, 247667, India
[3]Co-supervisor, Department of Mathematics, Dhanauri P.G. College, Dhanauri, Haridwar, Uttarakhand, 247667, India
Corresponding Author: [1] Soniya Saini, Research Scholar [1](soniyasaini599@gmail.com), [2]Email ID: fos.sandeep@motherhooduniversity.edu.in, [3]Email ID: drankurnehra648@gmail.com

ABSTRACT: Distributed computing systems enable the execution of multiple tasks by assigning them to different processors, thereby improving overall system performance and resource utilization. Efficient task scheduling in such systems is challenging due to uncertainty in execution costs, communication overheads, and dynamic processor characteristics. This paper proposes a novel mathematical framework for dynamic task scheduling in distributed computing systems operating under a fuzzy environment. The proposed model aims to maximize system dependability while minimizing overall execution cost. Phase-wise execution costs, job residency costs on heterogeneous processors, inter-task communication costs, and task relocation costs are modeled using fuzzy numbers to capture real-world uncertainties more effectively. A defuzzification technique is employed to transform the fuzzy optimization problem into an equivalent crisp formulation, enabling practical implementation. The effectiveness and applicability of the proposed approach are demonstrated through detailed mathematical formulation and illustrative numerical examples. The proposed framework is flexible and can handle processors with a random number of execution phases and varying program structures, making it suitable for complex and dynamic distributed computing environments.

[Saini, S., Tiwari, S.K. and Nehra, A **TASK SCHEDULING FOR DISTRIBUTED REAL-TIME SYSTEMS IN A FUZZY ENVIRONMENT.** *The International Journal of Interpretation, Observation and Analysis*, 2025; Volume 4, Issue 1:198-206 (October-December). ISSN 2349-0713, Peer-reviewed (online/offline), Refereed, Indexed and International Journal (Since 2013), Global Impact Factor: 6.205

Keywords: Fuzzy Logic, Fuzzy Inference System, Membership Functions, Fuzzy Rule Base

1. Introduction

A distributed real-time system (DRTS) enables the simultaneous execution of multiple transactions across several processors. Due to its inherent complexity and strict timing constraints, DRTS has attracted significant research interest. Among the various challenges associated with DRTS, system reliability remains one of the most critical issues. Reliability in DRTS can be enhanced by distributing program execution across multiple processors, efficiently utilizing available system resources, and exploiting heterogeneous computational capabilities. A highly reliable system is expected to operate continuously until the successful completion of program execution. Compared to centralized systems, DRTS typically offers higher reliability owing to the availability of multiple resources. However, merely increasing the number of resources does not guarantee improved reliability; effective task allocation and scheduling strategies are essential. Several researchers have proposed methodologies for evaluating and enhancing system reliability. Zahedi and Ashrafi (1991), Srinivasan and Jha (1999), and Lin (2002) presented different approaches for reliability assessment in distributed systems. According to Ramamritham and Stankovic (1994), DRTS plays a vital role in industrial and safety-critical infrastructures such as flight control systems, process control systems, and air traffic control. In such applications, time constraints are fundamental, and failure to respond within specified deadlines may lead to catastrophic consequences. This study proposes a fuzzy inference system (FIS) to optimize system reliability in distributed real-time environments. The literature presents a wide range of techniques aimed at maximizing system reliability; however, many of these approaches require precise and accurate system models to achieve practical applicability. In real-world environments, such precision is often difficult to obtain due to uncertainty and dynamic system behavior. To address this limitation, fuzzy logic provides an effective framework for handling vagueness and partial truth. As introduced by Zadeh (1980, 1994), fuzzy logic extends classical Boolean logic by incorporating degrees of truth, while fuzzy sets generalize traditional crisp sets. Fuzzy logic has been widely applied to scheduling and optimization problems in distributed and cloud computing environments. Fahmy (2010) proposed a fuzzy logic-based scheduling approach for non-periodic jobs in a soft real-time single-processor system to maximize throughput. Kong et al. (2011) developed an efficient dynamic task scheduling scheme using fuzzy prediction in virtualized data centers. Kaur and Verma (2012) introduced an evolutionary algorithm-based job scheduling method for cloud computing, while Krishnasamy (2013) applied hybrid particle swarm optimization for job scheduling in cloud environments. Tang et al. (2014) proposed a dynamic forecast-based scheduling strategy for virtual machine deployment, and Chen et al. (2015) developed an energy-efficient scheduling method for real-time workloads under uncertainty. Further advancements include Wang et al. (2016), who proposed a fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flow shop scheduling under machine breakdown conditions. Madni et al. (2017) evaluated heuristic scheduling approaches for Infrastructure-as-a-Service (IaaS) cloud environments, and Ben Alla et al. (2018) introduced a hybrid meta-heuristic task scheduling approach using dynamic queues. Arunarani et al. (2019) provided a comprehensive survey of task scheduling methods in cloud computing, while Alizadeh et al. (2020) presented an extensive review of task scheduling techniques in fog computing. Ebadifard and Babamir (2021) proposed an autonomic task scheduling algorithm incorporating load balancing for dynamic cloud workloads. Bezdan et al. (2022) developed a hybrid bat algorithm for multi-objective job scheduling in cloud environments. More recently, Hajvali et al. (2023) introduced a decentralized and scalable hybrid scheduling-clustering approach for real-time applications in unstable fog-cloud environments, and Ali and Sridevi (2024) proposed a fuzzy logic-based mobility- and security-aware real-time job scheduling framework for IoT-enabled fog-cloud systems.

2. Literature Review

Distributed and real-time computing systems have been extensively studied due to their increasing adoption in safety-critical, cloud, fog, and IoT-based applications. One of the most significant challenges in these systems is efficient task scheduling while ensuring high reliability, low latency, and optimal resource utilization. Over the years, researchers have proposed a wide range of scheduling and reliability optimization techniques, evolving from classical deterministic models to intelligent and uncertainty-aware approaches. Early research primarily focused on reliability-driven task allocation and scheduling in distributed systems. Zahedi and Ashrafi (1991) proposed a reliability allocation framework considering system structure, cost, and utility, emphasizing the importance of strategic resource distribution. Srinivasan and Jha (1999) further explored safety- and reliability-driven task allocation in distributed environments, highlighting the trade-offs between performance and system dependability. Ramamritham and Stankovic (1994) provided a foundational discussion on scheduling algorithms and operating system support for real-time systems, establishing time constraints as a critical factor in system design. Lin and Wu (2002) examined system performance from a service quality perspective, indirectly contributing to reliability evaluation in distributed platforms. With the increasing complexity of real-time and distributed systems, researchers began exploring intelligent techniques capable of handling uncertainty and dynamic workloads. Zadeh (1980, 1994) laid the theoretical foundation for fuzzy logic, demonstrating its superiority over classical Boolean logic in handling partial truth and uncertainty. These principles later became instrumental in modeling real-world scheduling problems where precise system parameters are often unavailable. Fahmy (2010) introduced one of the early fuzzy logic-based scheduling algorithms for non-periodic jobs in soft real-time systems, demonstrating improved throughput and adaptability. Kong et al. (2011) extended fuzzy logic to dynamic task scheduling in virtualized data centers by incorporating fuzzy prediction, achieving better load balancing and performance stability. Around the same time, evolutionary and swarm intelligence techniques gained popularity. Kaur and Verma (2012) applied genetic algorithms for efficient task scheduling in cloud computing, while Krishnasamy (2013) proposed a hybrid particle swarm optimization approach to enhance scheduling efficiency. As cloud computing matured, research attention shifted toward virtualization-aware and energy-efficient scheduling. Tang et al. (2014) developed a dynamic forecast-based scheduling algorithm for virtual machine placement, addressing resource prediction challenges. Chen et al. (2015) focused on energy-efficient scheduling of real-time workloads under uncertainty, emphasizing sustainability alongside performance. Wang et al.

(2016) combined fuzzy logic with estimation of distribution algorithms to solve distributed permutation flow-shop scheduling problems under machine breakdown conditions, demonstrating the robustness of hybrid intelligent approaches. Further advancements were observed in heuristic and meta-heuristic scheduling strategies. Madni et al. (2017) conducted a comparative performance evaluation of heuristic algorithms in Infrastructure-as-a-Service (IaaS) cloud environments. Ben Alla et al. (2018) proposed a hybrid meta-heuristic scheduling approach using dynamic queues, achieving significant improvements in response time and resource utilization. Arunarani et al. (2019) presented a comprehensive literature survey on task scheduling techniques in cloud computing, categorizing methods based on objectives, constraints, and solution strategies. With the emergence of fog and edge computing, scheduling research expanded beyond centralized cloud architectures. Alizadeh et al. (2020) provided a systematic review of task scheduling approaches in fog computing, highlighting latency reduction and decentralized decision-making as key challenges. Ebadifard and Babamir (2021) proposed an autonomic task scheduling algorithm using load balancing for dynamic cloud workloads, enabling self-adaptive system behavior. Bezdan et al. (2022) introduced a hybridized bat algorithm for multi-objective task scheduling, optimizing execution time, cost, and resource utilization simultaneously. Recent studies have increasingly focused on real-time, decentralized, and security-aware scheduling in heterogeneous environments. Hajvali et al. (2023) proposed a decentralized and scalable hybrid scheduling-clustering approach suitable for volatile fog-cloud environments, demonstrating improved resilience and scalability. Ali and Sridevi (2024) introduced a fuzzy logic-based mobility- and security-aware real-time task scheduling framework for IoT-enabled fog-cloud systems, addressing emerging challenges related to device mobility and data security. By 2025, it is evident that task scheduling research has evolved toward intelligent, hybrid, and uncertainty-aware models that integrate fuzzy logic, meta-heuristics, and decentralized architectures. While significant progress has been made, challenges remain in achieving optimal reliability, scalability, and real-time responsiveness under highly dynamic and uncertain environments. These gaps motivate the development of advanced fuzzy inference-based frameworks capable of optimizing system reliability while accommodating heterogeneous resources and unpredictable workloads.

3. The Proposed Model

The Mamdani FIS is constructed using execution times (ET), inter-task communication times (ITCT), and processor performance (POP) as input variables and the reliability of the system (ROS) as an output variable. Fig. 1 displays the block diagram for FIS.

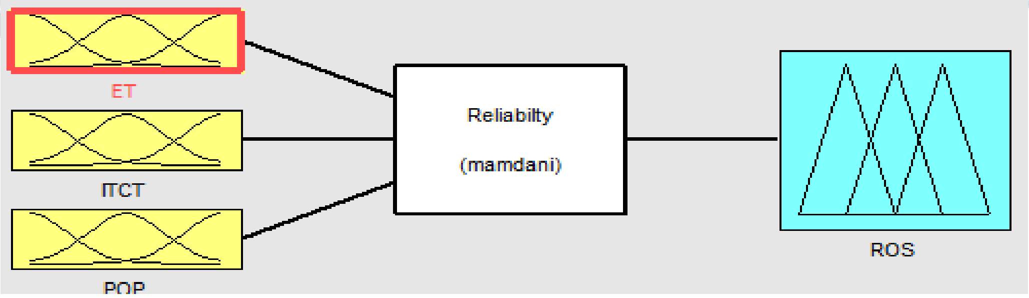


Fig. 1: Block Diagram of Fuzzy Inference System.

4. Membership Functions

For the first input variable, execution time (ET), the following membership functions are obtained.

$$\mu_{very\_low}(x)=\begin{cases}0, & x \geq 0112.5\\ 1, & 0.10 \leq x < 012.59\\ \dfrac{0112.50-x}{0112.50-012.590}, & 012.590 \leq x < 0112.50\end{cases}$$

$$\mu_{low}(x)=\begin{cases}0, & x<012.590 \text{ and } x\geq 0237.50\\ \dfrac{x-012.590}{0112.50-012.590}, & 012.590\leq x<0112.50\\ 1, & 112.5\leq x<0137.50\\ \dfrac{0237.50-x}{0237.50-0137.50}, & 0137.50\leq x<0237.50\end{cases}$$

$$\mu_{medium}(x)=\begin{cases}0, & x<0137.50 \text{ and } x\geq 0362.50\\ \dfrac{x-0137.50}{0237.50-0137.50}, & 0137.50\leq x<0237.50\\ 1, & 0237.50\leq x<0262.50\\ \dfrac{0362.50-x}{0362.50-0262.50}, & 0262.50\leq x<0362.50\end{cases}$$

$$\mu_{high}(x)=\begin{cases}0, & x<0262.50 \text{ and } x\geq 0487.50\\ \dfrac{x-0262.50}{0362.50-0262.50}, & 0262.50\leq x<0362.50\\ 1, & 0362.50\leq x<0387.50\\ \dfrac{0487.50-x}{0487.50-0387.50}, & 0387.50\leq x<0487.50\end{cases}$$

$$\mu_{very\_high}(x)=\begin{cases}0, & x<0387.50\\ \dfrac{x-0387.5}{0487.50-0387.50}, & 0387.50\leq x<0487.50\\ 1, & 0487.50\leq x\leq 500.00\end{cases}$$

These membership functions are depicted in Fig. 2.

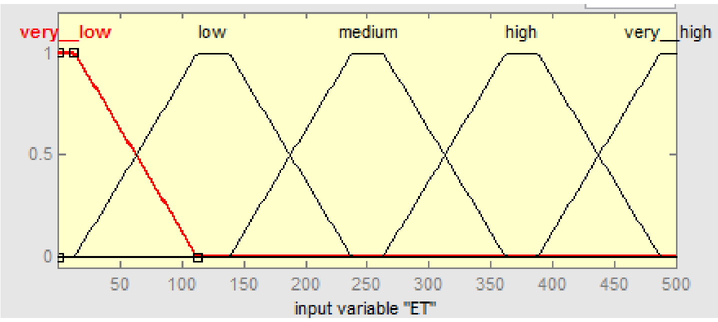


Fig. 2: Membership Function for ET Variables

For inter-task communication time (ITCT), the second input variable, the following membership functions are derived.

$$\mu_{very\_low}(x)=\begin{cases}0, & x \geq 055.80\\ 1, & 0.0 \leq x < 06.20\\ \dfrac{055.80-x}{055.80-06.20}, & 06.20 \leq x < 055.80\end{cases}$$

$$\mu_{low}(x)=\begin{cases}0, & x < 06.20 \text{ and } x \geq 0118.70\\ \dfrac{x-06.20}{055.80-06.20}, & 06.20 \leq x < 055.80\\ 1, & 055.80 \leq x < 068.30\\ \dfrac{0118.70-x}{0118.70-068.30}, & 068.30 \leq x < 0118.70\end{cases}$$

$$\mu_{medium}(x)=\begin{cases}0, & x < 068.30 \text{ and } x \geq 0180.80\\ \dfrac{x-068.30}{0118.70-068.30}, & 068.30 \leq x < 0118.70\\ 1, & 0118.70 \leq x < 0131.20\\ \dfrac{0180.80-x}{0180.80-0131.20}, & 0131.20 \leq x < 0180.80\end{cases}$$

$$\mu_{high}(x)=\begin{cases}0, & x < 0131.20 \text{ and } x \geq 0243.70\\ \dfrac{x-0131.20}{0180.80-0131.20}, & 0131.20 \leq x < 0180.80\\ 1, & 0180.80 \leq x < 0193.30\\ \dfrac{0243.70-x}{0243.70-0193.30}, & 0193.30 \leq x < 0243.70\end{cases}$$

$$\mu_{very\_high}(x)=\begin{cases}0, & x < 0193.30\\ \dfrac{x-0193.30}{0243.70-0193.30}, & 0193.30 \leq x < 0243.70\\ 1, & 0243.70 \leq x \leq 250.00\end{cases}$$

These membership functions are depicted in Fig.

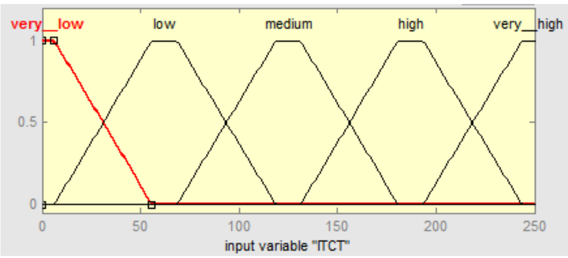


Fig. 3: Membership Function for ITCT Variables

For processor performance (POP), the third input variable, the following membership functions are obtained.

$$\mu_{very\_low}(x)=\begin{cases}0, & x\geq 0.2080\\ 1, & 0.0\leq x<0.0440\\ \dfrac{0.2080-x}{0.2080-0.0440}, & 0.0440\leq x<0.2080\end{cases}$$

$$\mu_{low}(x)=\begin{cases}0, & x<0.0440 \text{ and } x\geq 0.4460\\ \dfrac{x-0.0440}{0.1960-0.0440}, & 0.0440\leq x<0.1960\\ 1, & 0.1960\leq x<0.2930\\ \dfrac{0.4460-x}{0.4460-0.2930}, & 0.2930\leq x<0.4460\end{cases}$$

$$\mu_{medium}(x)=\begin{cases}0, & x<0.2980 \text{ and } x\geq 0.7010\\ \dfrac{x-0.2980}{0.4510-0.2980}, & 0.2980\leq x<0.4510\\ 1, & 0.4510\leq x<0.5480\\ \dfrac{0.7010-x}{0.7010-0.5480}, & 0.5480\leq x<0.7010\end{cases}$$

$$\mu_{high}(x)=\begin{cases}0, & x<0.5480 \text{ and } x\geq 0.9510\\ \dfrac{x-0.5480}{0.7010-0.5480}, & 0.5480\leq x<0.7010\\ 1, & 0.7010\leq x<0.7980\\ \dfrac{0.9510-x}{0.9510-0.7980}, & 0.7980\leq x<0.9510\end{cases}$$

$$\mu_{very\_high}(x)=\begin{cases}0, & x<0.7980\\ \dfrac{x-0.7980}{0.9510-0.7980}, & 0.7980\leq x<0.9510\\ 1, & 0.9510\leq x\leq 1\end{cases}$$

These membership functions are depicted in Fig. 4.

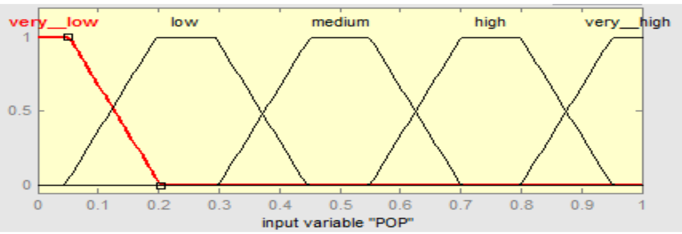


Fig. 4: Membership Function for POP Variables

The dependability of the system (ROS), the output variable, has the following membership functions.

$$\mu_{very\_poor}(x)=\begin{cases}0, & x\geq 0.2010\\ 1, & 0.0\leq x<0.0480\\ \dfrac{0.2010-x}{0.2010-0.0480}, & 0.0480\leq x<0.2010\end{cases}$$

$$\mu_{poor}(x)=\begin{cases}0, & x<0.0480 \text{ and } x\geq 0.4510\\ \dfrac{x-0.0480}{0.2010-0.0480}, & 0.0480\leq x<0.2010\\ 1, & 0.2010\leq x<0.2980\\ \dfrac{0.4510-x}{0.4510-0.2980}, & 0.2980\leq x<0.4510\end{cases}$$

$$\mu_{average}(x)=\begin{cases}0, & x<0.2980 \text{ and } x\geq 0.7010\\ \dfrac{x-0.2980}{0.4510-0.2980}, & 0.2980\leq x<0.4510\\ 1, & 0.4510\leq x<0.5480\\ \dfrac{0.7010-x}{0.7010-0.5480}, & 0.5480\leq x<0.7010\end{cases}$$

$$\mu_{good}(x)=\begin{cases}0, & x<0.5480 \text{ and } x\geq 0.9510\\ \dfrac{x-0.5480}{0.7010-0.5480}, & 0.5480\leq x<0.7010\\ 1, & 0.7010\leq x<0.7980\\ \dfrac{0.9510-x}{0.9510-0.7980}, & 0.7980\leq x<0.9510\end{cases}$$

$$\mu_{execellent}(x)=\begin{cases}0, & x<0.7980\\ \dfrac{x-0.7980}{0.9510-0.7980}, & 0.7980\leq x<0.9510\\ 1, & 0.9510\leq x\leq 1\end{cases}$$

These membership functions are depicted in Fig. 5.

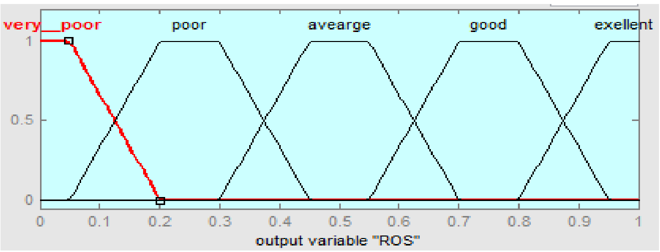
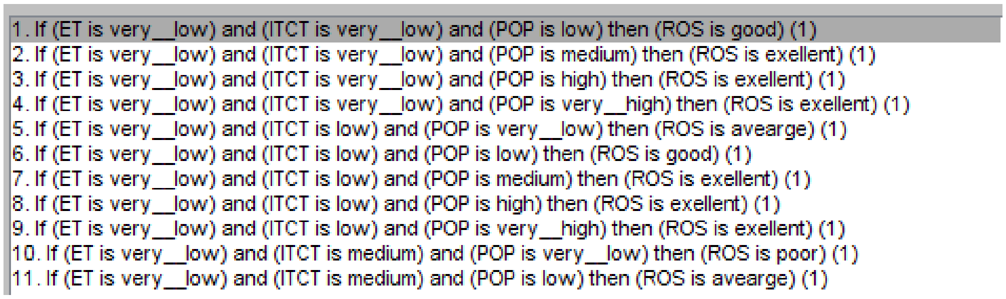


Fig. 5: Membership Function for ROS Variables

Figures 2 to 5 make it clear that the membership values of fuzzy variables correlate with the y-axis. The x-axis is represented by the quantized felt values for the input fuzzy variables execution time, inter-task communication time, and processor performance, in that order. Conversely, the width and center of membership functions of these fuzzy variables can be readily adjusted to accommodate outside circumstances and variables.

5. Fuzzy Rule Base

An expert-provided body of information given in an If-then format is called a fuzzy rule base. It is used to demonstrate the relationship between fuzzy input and output variables as well as to show how one or more input variables affect an output variable. Using the rule editor, the user can add and amend rules that define how the system operates. Based on the separation of input and output variables, 125 rules are created to evaluate the system's dependability. A few sample rules that were extracted from the rule base are shown in Fig. 6.



```
1. If (ET is very__low) and (ITCT is very__low) and (POP is low) then (ROS is good) (1)
2. If (ET is very__low) and (ITCT is very__low) and (POP is medium) then (ROS is exellent) (1)
3. If (ET is very__low) and (ITCT is very__low) and (POP is high) then (ROS is exellent) (1)
4. If (ET is very__low) and (ITCT is very__low) and (POP is very__high) then (ROS is exellent) (1)
5. If (ET is very__low) and (ITCT is low) and (POP is very__low) then (ROS is avearge) (1)
6. If (ET is very__low) and (ITCT is low) and (POP is low) then (ROS is good) (1)
7. If (ET is very__low) and (ITCT is low) and (POP is medium) then (ROS is exellent) (1)
8. If (ET is very__low) and (ITCT is low) and (POP is high) then (ROS is exellent) (1)
9. If (ET is very__low) and (ITCT is low) and (POP is very__high) then (ROS is exellent) (1)
10. If (ET is very__low) and (ITCT is medium) and (POP is very__low) then (ROS is poor) (1)
11. If (ET is very__low) and (ITCT is medium) and (POP is low) then (ROS is avearge) (1)
```

Fig. 6: A Part of the Rules from the Rule Base.

6. Conclusion

We have created a fuzzy inference system in this research to optimize DRTS reliability. For the fuzzy inference system, execution time, intertask communication time, and processor performance are input factors, and system dependability is an output variable. Five categories of linguistic variables have been created to group all input and output factors. In this case, 125 rules total have been produced for the DRTS reliability optimization using a combination of input and output variables. It is extremely evident from this experiment that the ROS changes when the input variables ET, ITCT, and POP change, with the use of MATLAB Surface Viewer to observe these input and output variables. This study presented a fuzzy inference system for optimizing reliability in distributed real-time systems under uncertain and dynamic conditions. The proposed framework overcomes the limitations of conventional scheduling and reliability models by incorporating fuzzy logic to handle imprecision in system parameters. The mathematical formulation and numerical validation demonstrate that the approach enhances system reliability and adaptability in heterogeneous computing environments. The model's flexibility allows it to support processors with varying execution phases and complex task structures, making it suitable for modern distributed architectures. Future research may extend this work by integrating hybrid optimization techniques such as machine learning, evolutionary algorithms, or reinforcement learning to dynamically tune fuzzy rules and membership functions. Additional objectives such as energy efficiency, security, fault tolerance, and scalability can also be incorporated into the model. Furthermore, real-world implementation and simulation using large-scale cloud-fog-IoT environments would provide deeper insights into the practical effectiveness of the proposed approach.

References

1. Ali, H. S., & Sridevi, R. (2024). Mobility and security-aware real-time task scheduling in fog-cloud computing for IoT devices: a fuzzy-logic approach. *The Computer Journal*, *67*(2), 782-805.
2. Alizadeh, M. R., Khajehvand, V., Rahmani, A. M., & Akbari, E. (2020). Task scheduling approaches in fog computing: A systematic review. *International Journal of Communication Systems*, *33*(16), e4583.
3. Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, *91*, 407-415.
4. Ben Alla, H., Ben Alla, S., Touhafi, A., & Ezzati, A. (2018). A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for the cloud computing environment. *Cluster Computing*, *21*(4), 1797-1820.

5. Bezdan, T., Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, E., & Tuba, M. (2022). Multi-objective task scheduling in a cloud computing environment by a hybridized bat algorithm. *Journal of Intelligent & Fuzzy Systems*, *42*(1), 411-423.
6. Chen, H., Zhu, X., Guo, H., Zhu, J., Qin, X., & Wu, J. (2015). Towards energy-efficient scheduling for real-time tasks under an uncertain cloud computing environment. *Journal of Systems and Software*, *99*, 20-35.
7. Ebadifard, F., & Babamir, S. M. (2021). Autonomic task scheduling algorithm for dynamic workloads through a load-balancing technique for the cloud-computing environment. *Cluster Computing*, *24*, 1075-1101.
8. Fahmy M. M. M. (2010). A Fuzzy Algorithm for Scheduling Non-Periodic Jobs on Real Real-Time Single-Processor System. *Ain Shams Engineering Journal*. 1, 31-38.
9. Hajvali, M., Adabi, S., Rezaee, A., & Hosseinzadeh, M. (2023). Decentralized and scalable hybrid scheduling-clustering method for real-time applications in volatile and dynamic Fog-Cloud Environments. *Journal of Cloud Computing*, *12*(1), 66.
10. Kaur, S., & Verma, A. (2012). An efficient approach to the genetic algorithm for task scheduling in the cloud computing environment. *International Journal of Information Technology and Computer Science (IJITCS)*, *4*(10), 74-79.
11. Kong, X., Lin, C., Jiang, Y., Yan, W., & Chu, X. (2011). Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction. *Journal of Network and Computer Applications*, *34*(4), 1068-1077.
12. Krishnasamy, K. (2013). Task scheduling algorithm based on Hybrid Particle Swarm Optimization in the cloud computing environment. *Journal of Theoretical & Applied Information Technology*, *55*(1).
13. Lin, C. S., & Wu, S. (2002). Exploring the impact of online service quality on portal site usage. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 2654-2661). IEEE.
14. Madni, S. H. H., Abd Latiff, M. S., Abdullahi, M., Abdulhamid, S. I. M., & Usman, M. J. (2017). Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. *PloS one*, *12*(5), e0176321.
15. Ramamrithm K. and Stankovic J. A. (1994). Scheduling Algorithms and Operating Systems Support for Real-Time Systems. *Proceedings of the IEEE*. 82(1), 55-67.
16. Srinivasan S. and Jha K. N. (1999). Safety and Reliability Driven Task Allocation in Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*. 10, 238-250.
17. Tang, Z., Mo, Y., Li, K., & Li, K. (2014). Dynamic forecast scheduling algorithm for virtual machine placement in the cloud computing environment. *The Journal of Supercomputing*, *70*, 1279-1296.
18. Wang, K., Huang, Y., & Qin, H. (2016). A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flow-shop scheduling problems under machine breakdown. *Journal of the Operational Research Society*, *67*(1), 68-82.
19. Zadeh L. A. (1980). Fuzzy Sets versus Probability. *Proceedings of the IEEE*. 68(3), 421-431.
20. Zadeh L. A. (1994). Fuzzy Logic, Neural Networks, and Soft Computing. *Communication of the ACM*. 37(3), 77-84.
21. Zahedi E. and Ashrafi N. (1991). Reliability Allocation Based on Structure, Utility, Price, and Cost. *IEEE Transactions on Software Engineering*. 17, 345-356.