An Efficient Framework for Peer Selection in Dynamic P2P Network using Q learning with Fuzzy Linear Programming

M. Anandara[1†] **T. Albalawi[2††]**

[1†] PSNA College of Engineering and Technology, Department of IT, Dindigul, Tamilnadu, India
[2††] Imam Mohammad Ibn Saud Islamic University (IMSIU), Department of Computer Science, Riyadh, KSA

Abstract
This paper proposes a new approach where the Q-learning, which is one of the Reinforcement Learning (RL) techniques, is integrated into the Fuzzy Linear Programming (FLP) paradigm for improving peer selection in P2P network. By using Q-learning, the proposed method employs real-time feedback for adjusting and updating the peer selection strategies in real-time. The FLP framework enriches this process by maintaining imprecise information by the use of the fuzzy logic. It is used for achieving multiple objectives such as to enhance the throughput rate, reduce the delay time and guarantee reliable connection. This integration effectively solves the problems of network uncertainty, making the network configuration more stable and flexible. It is also important to note that throughout the use of the Q-learning agent in the network, various state metric indicators including available bandwidth, latency, package drop rates, and connectivity of nodes are observed and recorded. It then selects actions by choosing optimal peers for each node, updating a Q-table that defines states and actions based on these performance indices. This reward system guides the agents learning, refining its peer selection strategy over time. The FLP framework supports the Q-learning agent by providing optimized solutions that balance competing objectives under uncertain conditions. Fuzzy parameters capture variability in network metrics, and the FLP model solves a fuzzy linear programming problem, offering guidelines for the Q-learning agents decisions. Experimental results prove the effectiveness of method. Simulation using Erdos-Renyi model shows throughput increased by 21% and latency decreased by 17%. The computational efficiency was also notably improved, with computation times diminished by up to five orders of magnitude with compared to traditional methods.

Keywords:
Erdos-Renyi model, Fuzzy Linear Programming, Q-learning, P2P network, Q-table, Reinforcement Learning

1. Introduction

P2P networks today are an integral part of the advanced coherent communication systems and allow for efficient, comprehensive and distributed distribution of various resources and services[1]. They contrast with the system of Client-Server where a certain element maintains authority over a number of other elements; with P2P, the communication between the peers is direct. This feature is beneficial in many ways, providing better resilience to failure, more efficient utilization of resources and expandability. However, as in any P2P networks, dynamics and heterogeneity of the environment pose major issues, especially related to peer selection[2,3]. Peer selection is important process that is used in the P2P networks whereby suitable peers are chosen for sharing and exchange of data. Proper choices of the peer nodes can lead not only to increased communication throughput, decreased delay, or even provide increased reliability of the whole network. However, the fact that the P2P networks are dynamic in nature, with peers being joined and leaving the network frequently, unpredictable network conditions and different capabilities of the peers make the selection process trickier[4,5].

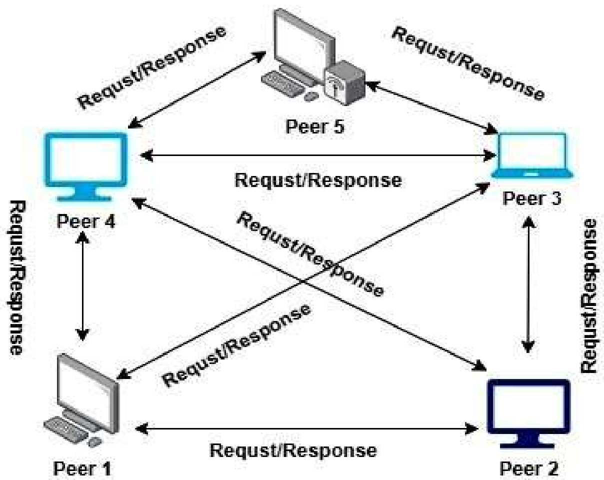

Figure 1. P2P network

A significant problem of P2P networks is a phenomenon called churn, which means that peers join and leave the network frequently[6]. The high churn rate causes most of the connections to be interrupted and resultant frequent changes in peer lists

Manuscript received November 5, 2025
Manuscript revised November 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.11.5

for regeneration. It is important that this process of peer selection be flexible to deal with these often frequent changes to the list of peers where new peers can made easily and any leaving peers should not affect the network[7]. P2P networks comprise a large number of participants possessing different levels of computing power, storage space, and available and sustainable bandwidth[8]. Basic P2P network with 5 nodes is represented in figure1. This paper presents an optimization approach that incorporates RL, specifically Q-learning in conjunction with a FLP approach to selecting peers in dynamic P2P networks. The proposed method employs Q-learning critically in that it autonomously adjusts the peer selection strategies in real-time feedback hence facilitating learning in ongoing process. The FLP framework uses fuzzy logic to deal with imprecise data and criteria while enabling the achievement of multiple objectives, such as throughput, low latency, and reliable connection. By incorporating fine-tuned Q-learning with FLP, the method is capable of dealing with uncertainties, which provides a more robust and adaptive form of network configuration. The findings of this research suggest that adaptive learning capabilities when integrated with fuzzy logics can improve the performance and reliability of P2P networks, thus providing better tools for selecting peers in random environments. The major objectives of this paper are given as follow.

- Design a comprehensive model integrating Fuzzy Linear Programming techniques with Q learning to optimize the selection of peers in P2P CDN.
- Examine the applications of the Fuzzy Linear Programming technique and Q learning in handling dynamic network conditions and uncertainties to ensure robust peer selection mechanisms.
- Analyze the impact of Fuzzy Linear Programming and Q learning on resource allocation, intending to attain impartial content distribution among the participating peers at the same time maximizing throughput and also reliability of the network.

The aim of the proposed system is to provide a solution of how the integration of Fuzzy Linear Programming with Q learning transform peer selection processes in P2P CDN into effective and simple one. The remaining part of the article is prepared as follows. Section 2 explains the literature review of the proposed work and existing mechanisms and their limitations to resolve the problem of peer selection. Section 3 elaborates the proposed method, including the fuzzy logic programming technique and Q learning for the peer selection framework. Section 4 states the results and experimental analysis and discussion of the proposed system. Section 5 finally gives the conclusion of the paper and outlines future directions of research.

2. Related work

The criteria for peer selection in the P2P network is essential to optimize the data exchange process, minimize latency and increase the network's resilience[9]. Alternatively, the common approaches for peer selection are based on heuristic and criteria, which are Static analysis strategies that frequently use simple measures including distance, available bandwidth, and node's status. Unlike the natural selection whereby peers that are closer have more privileges, proximity-based selection means that peers nearer to the source are favoured in order to decrease latency and increase throughput[10]. The rationale is that physical separation of peers is typically inversely proportional to network delay and thus small network distances should lead to small delays[11]. However, this approach may not be very accurate due to the fact that it doesn't consider other crucial characteristics like bandwidth and the reliability of other peers. Bandwidth based selection tries to pair up peers that have the greatest amount of bandwidth as the goal here is to achieve the maximum throughput. This approach enables users to partner with reliable super peers who are capable of supporting high bandwidth transmission[12]. Although this approach can increase the rates of data transfer substantially, other important parameters, such as latency, stability of peers, and topology, in this case, may be understated and become a bottleneck if the high bandwidth peers are overloaded. The availability based peer selection techniques refer to the peers that are always connected and willing to share data[13]. The use of this approach proves beneficial in ensuring that only stable and reliable connectivity is established in the network. This way the network can invest in the peers that have high availability, thus improving the network overall stability and decreasing the possibility of interruption of the transfers. This may not result in the best throughput and latency even if it was once a powerful indicator of performance. However, there have been some shortcomings noted from the application of single criterion where multiple criteria have been used to develop a composite selection criterion for peers. Hybrid methods select the number of peers based on accessibility, bandwidth capacity, and nearby neighbouring host peers[14]. For instance, the performance and other characteristics of students can be assigned scores which help in arrival at a more comprehensive list of the peers[15].

Although there are benefits of using multi-criteria and hybrid models over single-criterion methods, these methods include the need to present weight values and various parameters which need to be adjusted manually, and may not be sensitive enough to deal with constantly varying data from a network[16]. Depending on their type, P2P systems are more structured and rely on specific algorithms for selecting other peers and disseminating data such as Distributed Hash Tables (DHT). They include Chord, CAN and Kademlia. DHTs offer unique node selection by using hash functions, thus making them scalable and capable of making efficient resources to be found[17]. However, they may be less robust and flexible and do not easily accommodate changes to the network conditions and other peer systems. As for the unstructured P2P networks, these do not establish a fixed hierarchy of the peers as the nodes. Gossip protocols, also referred as epidemic protocols have been employed in the promotion of peer selection and information dissemination in P2P networks[18]. The gossiping in a gossip-based system is that the information was periodically exchanged with a randomly selected neighbour within the network. It also means that this approach can promote the quality and reliability of information dissemination across the entire network relying on the use of redundancy[19]. But gossip protocols also cause overhead since many a time, messages are exchanged multiple times and do not always select the optimum neighbour for certain performance characteristics. This analysis discusses some of the ways that traditional peer selection methods in P2P networks have been done and the advantages and disadvantages that are associated to each of them. These include proximity-based, bandwidth-based, and availability based and while they provide simple solutions which are easy to implement, they are not very effective in some use cases. Hybrid methods and structured P2P networks used to provide more robust and scalable solutions but there is a lack of flexibility[20]. Unstructured networks and gossip protocols provide adaptability and resilience but they are inefficiencies under dynamic conditions. While traditional methods have been effective to changing degrees, there remains a necessity for more adaptive and dynamic approaches to optimize peer selection in ever changing dynamic P2P network environments[21,22]. This paper aims to build on these foundations, devising new strategies for improving peer selection to enhance performance and robustness of the network.

3. Proposed System

3.1 State Representation in P2P Networks

In P2P networks, accurately representing the state of the network is crucial for effective management and optimization. The state of a P2P network at any given time can be represented by a set of variables capturing essential characteristics such as node status, network topology, resource availability, performance metrics, and peer behaviour[23]. For instance, the state S can be defined as a vector:

$$S=[S_n,S_t,S_r,S_p,S_b] \tag{1}$$

where S_n represents the status of nodes such as whether active or inactive, S_t captures the network topology as a connection matrix, S_r indicates resource availability such as bandwidth and storage, S_p includes performance metrics like latency and throughput, and S_b denotes peer behaviour such as churn rate and reputation scores. This comprehensive state representation allows for detailed monitoring and management of the network's dynamic behaviour[24].

3.2 Transition Probability Matrix (TPM)

TPM is used to describe the probabilities of moving from one state to another over a given period and basic tool for modelling the state transitions in a P2P network. P is used to denote TPM which is a square matrix. Each element in the matrix P_{ij} represents the probability of transitioning from state i to state j:

$$P_{ij}=P_r(S_{t+1}=j \mid S_t=i) \tag{2}$$

In order to formulate the TPM, it is needed to define all possible states of the network. Then, data on state transitions is gathered by examining the network over some time slot. The transition probability from state i to state j is calculated as follows:

P_{ij}=Number of transitions from state i to state j /Total number of transitions from state i. (3)

This matrix used to help in understanding the network's dynamic behaviour and predicting future states.

Algorithm 1: State Representation and Transition Probability Matrix Analysis in P2P Network

1**. Initialization:**

- Define the network graph $G = (V, E)$, where Vrepresents the set of peers and E represents the links between them.
- Assign attributes to each peerv_i:
  - Upload bandwidth: u_i
  - Download bandwidth: d_i
  - Storage capacity: s_i
  - Processing power: p_i
  - Availability: a_i
- Initialize matrices:
  - Content stored by peers: C
  - Content demand by peers:D

2. State Transition Matrix:
- Define the state transition rate matrix Q where q_{ij}represents the rate of transition from state $i$to state j
- Define arrival rate λ_i and departure rate μ_i for each state i.

3. Formulate State Transition Rate Matrix Q:
- For each state i, the diagonal elements q_{ii} are computed as:

$$q_{ii} = \sum_{j\neq i} q_{ij} \quad (4)$$

- The off-diagonal elements q_{ij} are positive for $i \neq j$.

4. Compute State Transition Probabilities $p_{ij}(t)$:
- Using matrix exponential e^{Qt}, calculate the transition probabilities:

$$p_{ij}(t) = [e^{Qt}]ij \quad (5)$$

5. Steady-State Distribution:
- Solve the steady-state distribution π by solving the equation: $\pi Q = 0$, subject to the constraint: $\sum_i \pi_i = 1$
- The steady-state distribution π is a row vector representing the probabilities of being in each state.

6. Expected Time in Each StateT_i:
- Compute the expected time $T_i$spent in the state

$$T_i = \frac{1}{\mu_i.\pi_i} \quad (6)$$

7. Network Dynamics:
- Define the arrival rate $\lambda_i$as the rate of peer arrivals.
- Define the departure rate μ_i as the rate of peer departures or inactivity.
- In steady state, the total arrival rateλ equals the total departure rate M:
- Considering individual state arrival ratesλ_i, the equation becomes:

$$\sum_i \lambda_i\pi_i = \sum_i \mu_i\pi_i \quad (7)$$

8. Network Capacity and Performance Metrics:
- Evaluate the network capacity in terms of bandwidth, storage, and processing power.
- Calculate network performance indicators such as download and upload speed, content availability, and network latency.

By following these steps in Algorithm 1, the state representation and transition probability matrix analysis in a P2P network is comprehensively modelled, which shows the network dynamics, performance, and stability.

3.3 Fuzzy Linear Programming (FLP) for Peer Selection

Effective peer selection mechanism is critical in dynamic and heterogeneous P2P networks in order to optimize performance metrics such as throughput, latency, and network robustness[25]. FLP can be used to model and solve optimization problems in peer selection, where parameters are not precisely known and are better represented as fuzzy numbers. The objective is to choose the optimal set of peers which maximizes performance of network at the same time considering uncertainties in network conditions, resource availability and behaviour of peer. Fuzzy logic extends classical logic to manage the concept of partial truth, where truth values range between completely true and false[26]. Fuzzy sets were introduced by Lotfi Zadeh during 1965. Unlike classical sets, where elements either belong or do not belong to the set, fuzzy sets allow for partial membership, characterized by a membership function μ:X→[0,1][27]. Consider the fuzzy set A representing high bandwidth in a P2P network. Different bandwidth values have different degrees of membership in $\overline{A}$.

$$\mu_{\overline{A}}(x) = \begin{cases} 0 & if\ x \le 10Mbps \\ \dfrac{x-10}{20-10} & if\ 10 < x < 20Mbps \\ 1 & if\ x \ge 20Mbps \end{cases} \tag{8}$$

Fuzzy numbers are a special type of fuzzy set used to represent uncertain quantities. A common representation is a triangular fuzzy number (TFN), defined by a triplet (l,m,u), where l is the lower limit, m is the most likely value, and u is the upper limit. The membership function for a TFN $\overline{A}$ = (l, m, u) is:

$$\mu_{\overline{A}}(x) = \begin{cases} 0 & if\ x \le l \\ \dfrac{x-l}{m-l} & if\ l < x < m \\ \dfrac{u-x}{u-m} & if\ m < x \le u \\ 0 & if\ x > u \end{cases} \tag{9}$$

Algorithm 2: FLP

1. Define Optimization Problem: Optimize peer selection to maximize download rate, minimize latency, and optimize resource allocation in the P2P network.

2. Define Decision Variables: Let x_j be the decision variables representing the selection of peers.

3. Formulate Objective Function: The objective function $FLP\ (x)$ can be represented as:

$$FLP(x) = \sum_{i=1}^{n} \lambda_i . \mu_i(x) \tag{10}$$

where $\lambda_i$are the weights of the linguistic variables and $\mu_i(x)$ are the membership functions.

4. Formulate Constraints: Constraints can be expressed in the form:

$$\sum_{j=1}^{m} a_{ij} x_j \le b_i \tag{11}$$

where a_{ij} are the coefficients, x_j are the decision variables, and b_i are the constraint limits.

5. Define Membership Functions: For each fuzzy set A with linguistic variables {Low, Medium, High}, define membership functions:

$$\mu_{Low}(x) = f_{Low}(x)$$
$$\mu_{Medium}(x) = f_{Medium}(x)$$

6. Aggregation of Criteria: Aggregate the membership functions using an appropriate aggregation operator$A$:

$$\text{Aggregated Criteria} = A(\mu_1(x), \mu_2(x), \ldots, (\mu_n(x)) \tag{12}$$

7. Defuzzification: Convert the aggregated fuzzy value into a crisp value using a defuzzification operator$D$:

$$\text{CrispValue} = D(\text{Aggregated Criteria}) \tag{13}$$

8. Solve the FLP Problem: Use an optimization solver to find the optimal values of the decision variables that maximize or minimize the objective function $FLP(x)$ subject to the constraints.

Algorithm 2: FLP

1. Define Optimization Problem: Optimize peer selection to maximize download rate, minimize latency, and optimize resource allocation in the P2P network.

2. Define Decision Variables: Let x_j be the decision variables representing the selection of peers.

3. Formulate Objective Function: The objective function $FLP\ (x)$ can be represented as:

$$FLP(x) = \sum_{i=1}^{n} \lambda_i . \mu_i(x) \quad (10)$$

where $\lambda_i$are the weights of the linguistic variables and $\mu_i(x)$ are the membership functions.

3. Formulate Constraints: Constraints can be expressed in the form:

4.

$$\sum_{j=1}^{m} a_{ij}x_j \leq b_i \qquad (11)$$

where a_{ij} are the coefficients, x_j are the decision variables, and b_i are the constraint limits.

5. Define Membership Functions:

- For each fuzzy set A with linguistic variables {Low, Medium, High}, define membership functions:

$$\mu_{\mathrm{Low}}(\mathrm{x}) = \mathrm{f}_{\mathrm{Low}}(\mathrm{x})$$
$$\mu_{\mathrm{Medium}}(\mathrm{x}) = \mathrm{f}_{\mathrm{Medium}}(\mathrm{x})$$

6. Aggregation of Criteria: Aggregate the membership functions using an appropriate aggregation operator$A$:

$$\text{Aggregated Criteria} = \mathrm{A}(\mu_1(\mathrm{x}), \mu_2(\mathrm{x}), \ldots, (\mu_n(\mathrm{x})) \qquad (12)$$

7. Defuzzification: Convert the aggregated fuzzy value into a crisp value using a defuzzification operator$D$:

$$\text{CrispValue} = \mathrm{D}(\text{Aggregated Criteria}) \qquad (13)$$

8. Solve the FLP Problem:

Use an optimization solver to find the optimal values of the decision variables that maximize or minimize the objective function $FLP(x)$ subject to the constraints.

FLP can be effectively applied to optimize peer selection in a P2P network, considering various fuzzy parameters and constraints by following these steps[28,29]. The variables such as Peer Reliability, Download, Task Completion Rate and Latency, fuzzy sets can be defined using linguistic variables and membership functions to confine their imprecise characteristics[30]. In the domain of linguistic variables and membership functions for download speed characterization, the establishment of distinct linguistic variables like {Low, Medium, High} makes the categorization of download speeds into qualitative levels. Corresponding to each linguistic variable, membership functions, denoted as μ_Low(x), μ_Medium(x), and μ_High(x), respectively, assign degrees of membership to individual download speed values[31,32]. The figure 2 shows the fuzzified inputs variables such as Download Speed, Peer Availability, Content Delivery Rate and Delay.

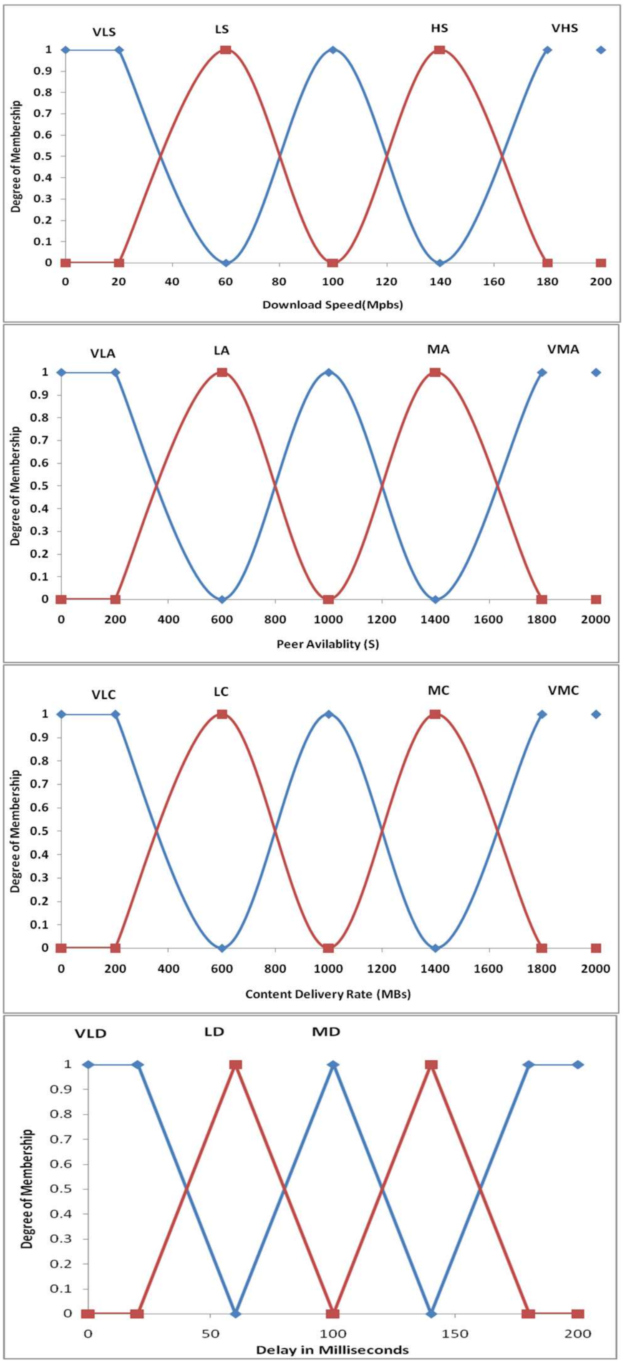

Figure 2. Fuzzified inputs variables. (a). Download Speed, (b)Peer Availability, (c). Content Delivery Rate (d) Delay

3.4 Q-Learning for Peer Selection Optimization

As can be seen, the Q-learning approach provides a substantial solution to the process of peer selection problem. This allows the learning of the best strategies concerning the environment through the interactions that it undertakes. It is a sort of learning where an agent determines how to take an action in an environment to attain the most rewarding cumulative sum of reward. It is especially suited for problems where the model of the environment is not known and complex. States, actions, rewards, and Q-values are the central part of Q-learning. States represent the different configurations and situations of the environment. In a P2P network, a state could encapsulate the current network topology, peer performance metrics, and also resource availability. Actions reflect the possible choices of an agent, which are discrete activities that it can perform like, connecting to/disconnecting from other peers. They contain feedback of an action initiated in a given state, to do so it reflects performance metrics such as high throughput, low latency, and reliable link. These are Q-values to determine expected cumulative rewards of an action taken in a state and the subsequent policy which is optimal is learnt iteratively from the agents' experience. The Q learning algorithm for updating the Q-value for a state-action pair is given as follows.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)] \quad (14)$$

where s is the current state, a is the action taken, r is the reward received after taking action a in state s, s′ is the resulting state after taking action a, α is the learning rate controlling the extent to which new information overrides old information, and γ is the discount factor accounting for the importance of future rewards. Targeted peers in P2P networks indicates the state of the network and that includes the topologies of P2P networks, performance of peers, availability of resources and status of peers. Network topology addresses the convergence and interconnection of peers and, peer's performances are best depicted in terms of bandwidth, latency, packet loss rate, and computational power. Availability of resource is the flow of the present status of the networks such as bandwidth, storage and others while peer status informs the system about the activity of the peer's such as joins and leaves. Establishing connections involves choosing new set of peers to connect based on their potential to improve network performance, while terminating connections involves deciding which existing connections to terminate if they are no longer beneficial. The reward function is designed to reflect the desired performance objectives of the P2P network. It is given below formula.

r(s,a)=w1* throughput + w2 * (−latency) + w3 * connectivity−w4*·resource_cost (15)

where w1, w2, w3, w4are weights assigned to each performance metric based on their relative importance.

The Q learning process begins with the Q Table being set to any random values more preferentially zero for all the state action pairs. Then, the agent uses an exploration strategy such as ε-greedy to balance exploration and exploitation: Where ε is the probability of exploratory move while with (1-ε) it chooses the action with the highest Q-value of the current state. After doing the chosen action the agent perceives the new state and the new reward and then changes the Q value in the formula with the new reward received and the maximum Q value of the new state. This process is iterative and continues until either the Q-values stabilize or a given number of iterations have been done. The figure 3 shows the Exploration and Exploitation Trade off over episodes. A high learning rate means change in experience affects Q-values more information about the future and, low learning rate leads to slow change about new information. The figures 4, 5, and 6 shows learning rate over epochs, iterations and episodes respectively. The discount factor takes the ratio between the present value and future value of each reward and ensures the value of the immediate reward does not overshadow future rewards. It determines the value of a state and action pair and how much the reward to assign to it when determining the expected future rewards. A high level of discounting implies a high value for delayed rewards than immediate ones, while low level of discounting gives preference to immediate rewards against future ones. This is illustrated in figure 7 with values of the discount factor different on the behaviour of the policy. The exploration rate controls the balance between exploration and exploitation in the learning process. A high exploration rate favours exploring new actions to find out potentially enhanced strategies, while a low exploration rate prioritizes exploiting the current best known actions to maximize instant rewards. As shown by the figure 8, the exploration rate is inversely proportional to episodes.

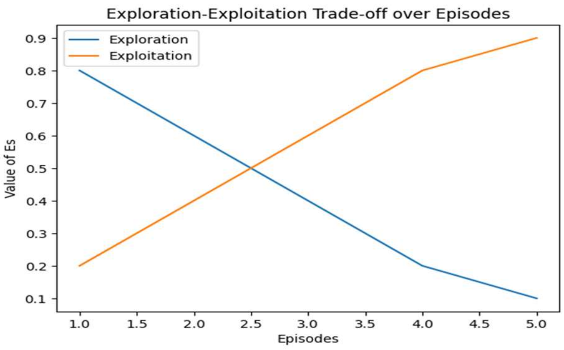

Figure 3. Exploration and Exploitation Rate

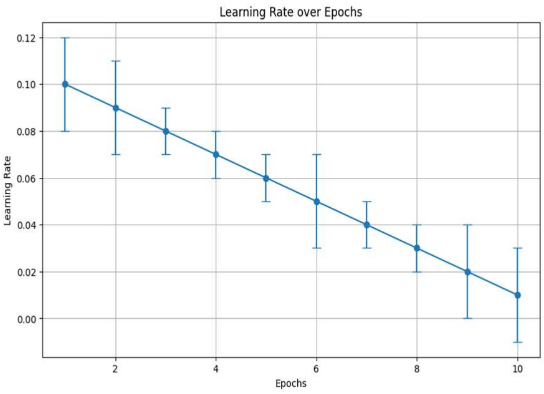

Figure 4. Learning rate over epochs

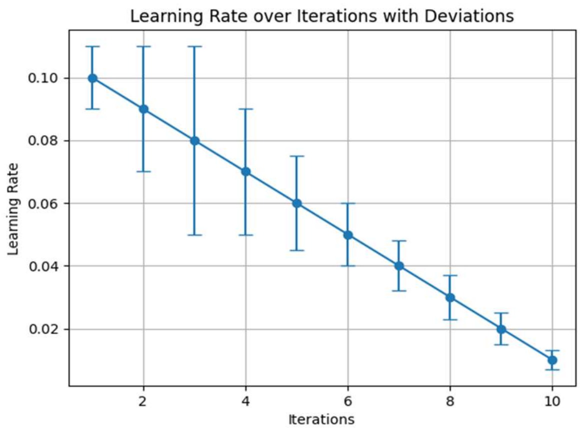

Figure 5. Learning rate over iterations

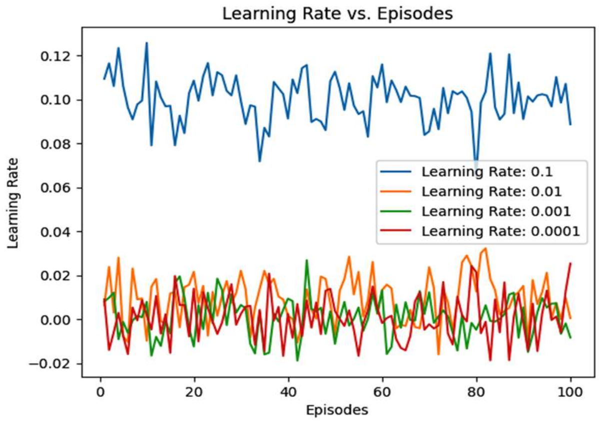

Figure 6. Learning rate Vs. Episodes

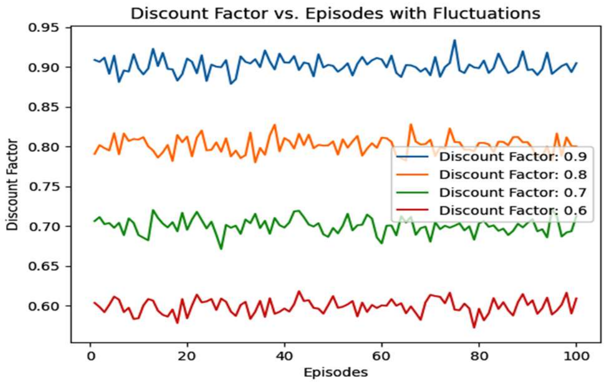

Figure 7. Impact of Discount factor

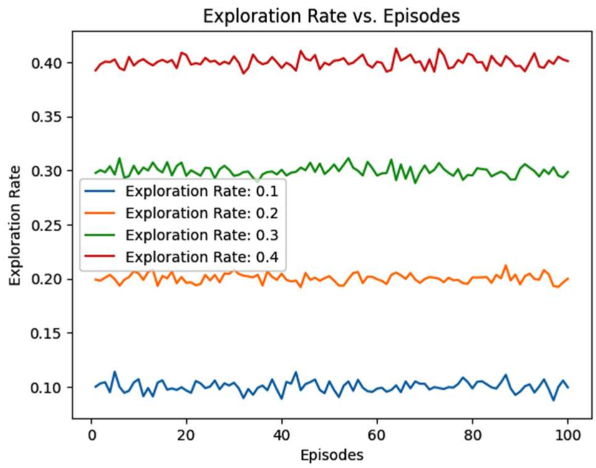

Figure 8. Impact of exploration rate

Algorithm 3: Q-Learning for Peer Selection Optimization

1. Initialize Q-Table:
   - Create a Q-table with states s and actions a:$Q(s,a) = 0 \ \forall s,a$ (16)
2. Set Parameters:
   - Learning rateα: determines the extent to which new information overrides old information.
   - Discount factor : measures the importance of future rewards.
   - Exploration rate : controls the trade-off between exploration and exploitation.
3. Define State, Action, and Reward:
   - States S: Represent configurations in the P2P network.
   - Actions A: Possible peer selection choices.
   - Reward R : Feedback received from the environment after taking an action.
4. Q-Learning Algorithm:
   - For each episode:
     - Initialize the state s.
     - Repeat until the state s is terminal:
       1. Choose Action:
          - Select an action a based on the ϵ-greedy policy:

$$\mathrm{a} = \begin{cases} \text{randomactionwithprobability}\epsilon \\ \arg\max_{\mathrm{a}'} \mathrm{Q}(\mathrm{s},\mathrm{a}')\text{withprobability } 1-\epsilon \end{cases}$$
(17)

       2. Perform Action:
          - Execute action a, observe the next state s' and reward r.
       3. Update Q-Table:
          - Update $Q(s,a)$ using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[\, r + \gamma max_{a'} Q(s',a') - Q(s,a)\,]$$

       4. Transition to Next State:
          - Set $s \leftarrow s'$.
       5. End of Episode:
          - Reduce exploration rate ϵover time to shift from exploration to exploitation.
       6. Policy Extraction
          - Derive the optimal policy from the Q-table:

$$\Pi(\mathrm{s}) = \arg\max_{\mathrm{a}} \mathrm{Q}(\mathrm{s},\mathrm{a}) \quad (18)$$

By following these steps, Q-Learning can be effectively applied to optimize peer selection in a P2P network, improving network performance through adaptive learning and decision-making.

3.5 Integration of Fuzzy Linear Programming (FLP) and Q-Learning

Integrating Fuzzy Linear Programming with Q-Learning is a powerful paradigm to improve the decision-making strategy in uncertain environments especially suitable for the P2P networks for solving peer selection problem. Fuzzy linear programming, an advancement of the conventional linear programming, utilises the fuzzy set theory in managing data that may be ambiguous in some ways[35]. Optimality is addressed in FLP models through the use of fuzzy objectives, constraints and decision variables to resolve conflicting objectives within a context of uncertainty. FLP makes use of fuzzy sets and fuzzy numbers to effectively model and process vagueness in information and, with the help of fuzzy logic operations solves optimization problems. Similarly, Q-Learning which is model-free reinforcement learning makes it possible to build value functions and select the best policy through a learning of interactions within an environment. Q-Learning is a form of learning in which the value of the state-action pair or Q-Value is estimated and then updated with the aids on the rewards received by the agents, thereby making it possible for the agents to learn good policies. However, like any other reinforcement learning technique, Q-Learning requires a sound exploration-exploitation strategy a critical element when dealing with large decision spaces. In this context, FLP helps in modelling states which may have vague or unpredictable information, for instance, peers' performance parameters and network status. By means of fuzzy objectives and constraints, FLP effectively defines and describes the optimisation objectives and constraints for peer selection that are characteristic of P2P networks while having regard to the essential vagueness and variability. Similarly, the Q-Learning agent communicates with the FLP framework and constantly updating the best policies of peer selection. Besides, the agent changes its strategies of selecting partners and further exploits the learned policies to gain as much cumulative reward as possible in the future. This integration enables the achievement of the conflicting objectives of optimization while also addressing the inherent uncertainty in the real-world P2P network environment with help of the FLP framework. By using the synergy between FLP and Q-Learning, this integrated framework presents a promising possibility to enhancing decision-making processes in dynamic and uncertain environments like P2P networks[36]. The figure 9 shows the proposed model.

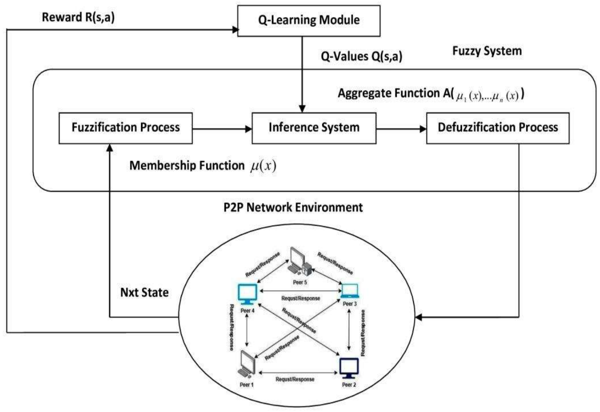

Figure 9 Q learning with FLP

Algorithm 4: Integration of Fuzzy Linear Programming (FLP) and Q-Learning

The integration of FLP and Q-Learning combines the advantages of handling uncertainty with fuzzy logic and the learning capability of reinforcement learning for optimized peer selection in P2P networks.

1. Define Fuzzy Sets and Membership Functions:

$$\mu_{Low}(x) = \frac{1}{1+e^{-(x-c_1)}} \quad (19)$$

$$\mu_{Medium}(x) = \frac{1}{1+e^{-(x-c_2)}} \quad (20)$$

$$\mu_{High}(x) = \frac{1}{1+e^{-(x-c_3)}} \quad (21)$$

where$c_1, c_2 2, c_3$ are constants defining the center of the fuzzy sets.

2. Formulate Fuzzy Linear Programming Problem:
   - Objective function:

$$\text{Maximize Z} = \lambda_1 . f_1(x) + \lambda_2 . f_2(x) + \cdots + \lambda_n . f_n(x) \quad (22)$$

   - Subject to fuzzy constraints:

$$\mu_i(a_{ij} . x_j) \geq b_i \forall i, j \quad (23)$$

3. Initialize Q-Learning Parameters:

$$\alpha = 0.1, \ \gamma = 0.9, \quad \epsilon = 0.1$$

4. Define States, Actions, and Rewards:
   - States $S = \{ s_1, s_2, \ldots, s_m\}$
   - Actions $A = \{ a_1, a_2, \ldots, a_n \}$
   - Reward R based on action's effectiveness:
     R(s,a) = Reward received after taking action A in states S
5. Run Q-Learning with Fuzzy Adjustments:
   - For each episode:
     1. Initialize state s .
     2. Choose action a using ∈-greedy policy:

$$a = \begin{cases} \text{randomactionifrand} < \epsilon \\ \arg\max_{a'} Q(s, a') \text{otherwise} \end{cases}$$

     3. Perform action a, observe the next state s' and reward r.
     4. Update Q-value:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a'} Q(s', a') - Q(s,a)]$$

     5. Ensure fuzzy constraints are satisfied:

$$\mu_i(a_{ij} . x_j) \geq b_i$$

6. Extract Optimal Policy:
   - Optimal policy $\pi(s) = \arg max_a Q(s,a)$
7. Apply Fuzzy Logic to Optimal Actions:
   - Optimize action selection using fuzzy logic:

$$\text{OptimalAction} = \arg\max_a [\textstyle\sum_{i=1}^{n} \lambda_i . f_i(x)] \text{subject to: } \mu_i(a_{ij} . x_j) \geq b_i \quad (25)$$

This integrated approach effectively combines fuzzy logic and Q-learning, enabling robust and adaptive peer selection in P2P networks by handling uncertainty and optimizing performance through learning.

4. Performance Evaluation

The performance of the proposed framework for peer selection optimization in P2P networks is done with the following key performance metrics.

4.1 Parameters

Throughput: It may be reported in terms of pps or packets per second. If the throughput is higher than the existing, network resources are utilized effectively and the data transferred can be done at a faster pace.

Latency: It is also referred to as delay which means the amount of time that a particular data packet takes to get through a particular network from the source to the destination. The basic unit of this measurement is millisecond (ms). Low latency means less time between data transmission and communication as well as shorter waiting time for the same. It is important in everyday uses like video on demand, online gaming and VOIP calls. Reducing latency by balancing the choice of optimum peers enhances the P2P system's response rate and QoS.

Connectivity: Connectivity means the closeness of the peers in a network considering the topology of the network. It quantifies connectivity and dependability of the communication links from peer to peer and it determines the fault tolerance of a network. Increased density results in multiple connection pathways within a network and decrease in the overall probability of network segmentation and isolation due to node or network failure. Approaches that concern about peers' selection for enhancing connectivity ensure effective and efficient information exchange among peers.

Resource Utilization: Bandwidth usage and distribution of load as well as scheduling and assignment of load across the distributed sites present the key quantitative measure of resource usage within the network. They evaluate how effectively the network is utilised in the available bandwidth capacity and the computing power of data transmission and computation. Conserving resources minimizes wastage hence optimizing utilisation of available resources thus improving efficiency and cost in the network.

QoS Satisfaction: Quality of Service (QoS) metrics, such as packet loss rate and jitter, measure the reliability and stability of data transmission in the network. Minimizing packet loss and jitter ensures consistent and reliable communication, particularly for real-time applications.

4.2 Existing Systems

In dynamic P2P networks, the kind of peer selection strategy need to be chosen to ensure that it fits the network characteristics and operational needs and the specific application requirements. Every of the scheme given below is unique in its advantages and limitations suggesting the importance of approaching the problem as a case of matching the needs to available resources, avoiding excessive latency, and improving the overall quality of the network. The following are some commonly used peer selection schemes and used for evaluating the proposed systems:

1. Random Peer Selection (RPS): Random peer selection is likely to be the least complex, where some arbitrary selection of the peers is made prevalent for data exchange and query of resources. This method has also been found to entail low costs of implementation and is relatively easy to use. However, its main drawback based on this model of peer selection is inefficiency; while peers can be randomly selected, there might not necessarily be the best use of the available resources. This randomness can result in high latency needed to access resources, mainly in a large network with low chances of randomly selecting an appropriate peer.
2. Neighbour Selection (NS): Neighbour selection focuses on choosing peers based on network proximity, which can mean latencies, or geographic proximity. In this scheme, neighbours are selected as nodes which are closer, hence the hope of reducing latency as closest neighbours more likely access resources in the locality. This means that it has to be constantly updated and monitored to ensure that it remains as effective as possible in identifying neighbours. However, this can be a serious problem in large networks due to the fact that updates are often needed and may lead to overhead in maintaining the neighbour lists.
3. Churn-Aware Selection (CAS): Churn-aware selection considers the probability of change in peer churn rates that is, situations where peers may join or exit the network. This scheme targets to improve the network longevity and reactiveness in that, it identifies peers that are less likely to churn. By adjusting to node join and leaving, churn-aware selection ensures an optimized flow and thus optimizes network continuity. Yet, above mentioned churn patterns may be difficult to predict, they demand accurate algorithms alongside real-time control procedures that can adapt peer selections on the fly.
4. Social-Based Selection (SBA): This technique is to use the characteristics of the social network or the trust relationship between peers to select reliable nodes for resource finding. Those with better social connectivity or trust score would be favoured when

the data exchange is being done. This scheme improves the efficiency of resource discovery based on current social structures but has strong demands on the trust management. Trust can also come under threat if trust metrics are corrupted, wherein security threats may occur or when peer selections are questionable.

5. Utility-Based Selection (UBS): Utility-based selection involves choosing those peers that are likely to offer the resources needed by the requester. Peers that offer a higher utility with respect to available bandwidth, storage space or processing capacity are selected for information sharing. This scheme enhances usage of scarce resources and performance of the network through proper correlation between requirements and available resources. However, estimating the utilities of peers and avoiding free-rider-situation, where peers benefit from a resource without contributing in a way that is deemed sufficient by the other members of the group, remains a major problem.

4.3 Simulation

An Erdos-Renyi (ER) graph is characterized by two parameters: the number of nodes in the network, usually given by n and the probability that any two nodes are linked by a given edge denoted by p. The model referred to as G(n,p) is preferred for its simplicity and applicability to model random connections in different types of real networks especially in P2P networks. In P2P networks the nodes are the peers or the participants, while the edges are the possible channels of data transfer between the peers. Symmetry and determination of Erdos-Renyi graphs are random in nature and hence suitable for modelling the dynamic and decentralized nature of P2P networks, peers join and leave the system frequently and the connections and disconnections are unpredictable. ER graphs offer means to model properties of a key network as connectivity, degree distribution, clustering, and path length, making the system valuable for our purposes. They provide system designers and analysts methods to model and schedule the P2P networks, which can suggest how the P2P networks must pattern to achieve optimal communication efficiency and reliability. The figure 10 shows the Erdos-Renyi graphs with random connections for 100 peers. Simulation factors as presented in the Table 1, 2 and 3 below. The RL agent communicates with the ns3 simulator through monitoring the status of the whole network, including link bandwidth, delay, packet drop probabilities, and the statuses of nodes. From these observations, the RL agent chooses the actions that are the decisions on which other nodes the given node ought to be connected or maintain connection with. These actions are performed at the ns3 simulation level through changing the routing tables and connection settings of the nodes. The ns3 simulator then contacts the RL agent, providing it with feedback in the form of rewards. For this simulation, the rewards consist of network performance parameters such as throughput, latency, and packet delivery ratios. At the same time, the RL agent is connected with the FLP model, which consists of the fuzzy system that accounts for uncertainty and imprecision of data in the network. The FLP model then uses a fuzzy linear programming to find the optimal or near-optimal peer required to meet the aforementioned goals without violating them by a large extent. The information that is gathered by the RL agent is used in the form of these optimizations to help the agent to make its decisions.

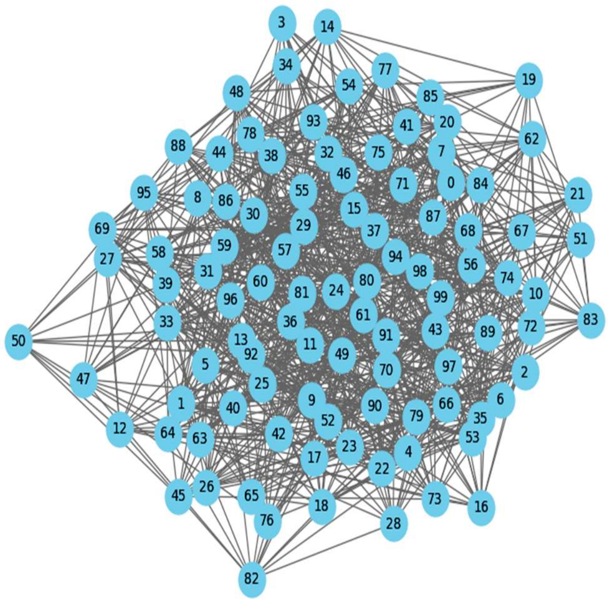

Figure 10. Erdos-Renyi graph using random connections

Table 1. P2P Network Configuration

Parameter	Value
Simulation Duration	100 seconds
Number of Peers	100 to 600
Network Topology	Erdos-Renyi graph
Content Repository Size	10 GB
Bandwidth	100 Mbps
Peer Upload Capacity	10 Mbps
Peer Download Capacity	20 Mbps
Max/Min Arrival Rate	50 / 10 peers per minute
Max/Min Departure Rate	30/5 peers per minute
Traffic Model	Constant Bit Rate (CBR)

Table 2. Q learning parameters

Parameter	Value
Learning Rate (α)	0.1
Discount Factor (γ)	0.9
Exploration Rate (ε)	0.2
Exploration Decay Rate	0.99
Initial Q-Value	0
Number of Episodes	1000
Maximum Steps per Episode	100
Reward for Successful Download	100
Penalty for Failed Download	-10

Table 3. Fuzzy Linear Programming parameters

Parameter	Value
Max Download Speed	10 Mbps
Min Download Speed	1 Mbps
Max Reliability	0.9
Min Reliability	0.5
Max Latency	100 ms
Min Latency	10 ms
Max Completion Rate	95%
Min Completion Rate	80%
membership functions	Triangular functions
weights	equal

4.4 Dataset Structure

4.4.1. State Space

For instance in the use of Q-Learning to integrate a FLP in the peer selection of a P2P network, the state space refers the different situations and arrangements of the network. This state space is useful because it lays out the different conditions under which decisions about selection of peer need to be made. For example, when the number of peers may be between 100 and 600, such state variables as the number of active peers, load in the network, bandwidth availability and trust between peers will be included in the state space. Every state collects the state of the network at a particular time, allowing the Q-Learning algorithm to apprehend the network's continually evolving status. Since the algorithm models these states correctly it is able to come up with the best decision based on the different statuses of a network. Therefore enhancing the efficiency and effectiveness of peer selection is as done. This dynamic representation aids in emulating context whereby changes in the network occur; characteristics of real-world systems that can benefit the Q-Learning framework in refining durations and tendencies of peer interactions.

There are several significant attributes that deserve identification in order to shape the state space and improve decision making in this integration. Number of active peers represents the number of peers at any given time, which define connectivity and resource sharing in the network size. Network Load defines the movement of data and its impact on congestion rate and efficiency of using the available resources. Bandwidth Availability is the availability of communication link to decide the speed of transmission of data which is apt for peers. Resource usage concerns with how useful peer's resource is that is used in studying performance as well as availability. Latency is the amount of time taken by the data to travel from the source to the intended destination, where the lower the value the better the response of a network. QoS Metrics measure the general quality of service with the results integrated by throughput and error rates influencing the level of satisfaction of end-users. Peer availability provides the numbers of time peers are available online necessary for network reliability. Resource demand highlights the needs of peers, ensuring efficient resource distribution. At last, fuzzy membership values from fuzzy logic provide nuanced insights into how attributes meet specific criteria, facilitating more adaptable and flexible selection of peer. These attributes collectively make a robust and adaptive optimization process. The state space representation is given in the Table 4.

Table 4. State Space Representation

State ID	Active Peers	Network Load (%)	Bandwidth Availability (Mbps)	Peer Trust Level
S1	100	50	100	High
S2	200	60	80	Medium
S3	300	40	120	Low
S4	400	70	90	High
S5	500	55	110	Medium
S6	600	65	95	High

4.4.2. Action Space

Action Space includes the options where the system or peers could make changes that improve the standards of the network. It encompasses deciding which peer to communicate with for data transfer, changing resource usage and revamping on peer connections in relation to the network scenario. Furthermore, It also consists of the managing of the replication that can result to high availability, the management of network load by reallocation of resources and handling of peer churns by changing the strategies with the network changes. The action space also includes the execution of QoS changes, and the decision making based on fuzzy logic when dealing with imprecise and/or unknown information. Every measure that is carried out in this space impacts on the network utilization parameters including throughput, latency and connectivity. In this way, the actions explained must be examined to search and filter those actions that contribute the best results for network performance and should be identified within the course of the learning process by the Q-Learning algorithm, to operate at the specified degree of the P2P system. The characteristics of each peer consequently contribute to the identification of the operation environment and necessary choices. Such attributes include the Peer ID that serves to identify each peer in the network with a lot of precision. Available bandwidth gives an evaluation of the ability of a peer to accommodate data transfer and is used in the selection of which peer to assign data requests. Resource contribution measures the storage capacity and other resources one can offer a peer which is important when sharing and replicating data. While connection quality standards refer to the indicators of latency and reliability, these are vital when it comes to enabling efficient and, at the same time, timely communication with the peers. QoS metrics includes parameters such as delay and jitter in an effort to make sure that the provided QoS is in keeping with the expected network QoS. Peer Load can define the current load on a peer thus not compromising its load resources and ability to work on many other networks. Historical performance data contain information concerning past conduct and dependability that aids in decision making based on proven performance. The action space representation is given in Table 5.

Table 5. Action Space Representation

Action ID	Action Description
A1	Select Peer Based on Bandwidth
A2	Select Peer Based on Trust Level
A3	Select Nearest Peer
A4	Select Peer with Least Load
A5	Random Peer Selection

4.4.3. Rewards

Interestingly, the proposed integration includes a reward system that helps peers modify their behaviour and, subsequently, improve other aspects of network performance. The aid incentives are aimed at promoting behaviours that lead to desirable outcomes in terms of the networks' performance. For instance, in order to receive positive reward, peers take actions promoting data distribution, which is good for their interaction. Likewise, incentives are given for low latency and high throughout which encourage activities that favour low response times and greater bandwidth. Low resource wastage and high QoS satisfaction is also encouraged so that peers utilise the network resources optimally and provide very high quality service. Sustaining superior connectivity and optimizing peer turnover are further incentivized in order to sustain network stability in face of prospective and constant peer changes. This reward system is a fundamental component in the Q-Learning algorithm because it gives back to peer learners feedback regarding the efficiency of certain actions in the other learner's learning environment, in this case, the P2P network. By such an approach, the integrated Q-Learning and FLP framework enhances constant learning in the selection of peers and improves the network's efficiency and reliability. The rewards representation is given in Table 6.

Table 6. Rewards Representation

State ID	Action ID	Reward (Q-Value)
S1	A1	10
S1	A2	7
S1	A3	5
S1	A4	8
S1	A5	3
S2	A1	6
S2	A2	9
S2	A3	4
S2	A4	7
S2	A5	2
S3	A1	8
S3	A2	6
S3	A3	7
S3	A4	5
S3	A5	4
S4	A1	9
S4	A2	8
S4	A3	6
S4	A4	7
S4	A5	3
S5	A1	10
S5	A2	9
S5	A3	8
S5	A4	7
S5	A5	5
S6	A1	12
S6	A2	10
S6	A3	9
S6	A4	8
S6	A5	6

4.4.4. Transition Probability

Transition probability in the proposed integration for peer selection in P2P networks refers to the measure of the probability with which a peer moves from one state to another overtime given an action has been taken. It measures the likelihood of a peer to transition from state s to another new state s′ following a particular action a. This probability defines the likelihood of the action as a function of the action impact, the network flow, and the behaviours of the other peers. If a peer is currently "Idle" and decides to "Forward Data Request," the transition probability P(s′,s,a) is used to measure the chance of the peer successfully entering the "Requesting Data" state. Exact estimation of these probabilities is crucial for Q-Learning, as it helps in predicting the outcomes of actions and optimizing decision making processes. Through historical and simulation data, these probabilities can be derived to assist peers to make better decisions, thus improving the efficiency and performance of the P2P network. The Transition Probability Representation is given in Table 7.

Table 7. Transition Probability Representation

Current State	Action	Next State	Probability
S1	A1	S2	0.4
S1	A1	S3	0.6
S2	A2	S4	0.7
S2	A2	S5	0.3
S3	A3	S1	0.5
S3	A3	S2	0.5
S4	A4	S3	0.8
S4	A4	S5	0.2
S5	A5	S1	0.6
S5	A5	S4	0.4
S6	A1	S2	0.5
S6	A1	S4	0.5

4.5 Result and Discussion

Throughput parameters show that the proposed framework outperforms the existing peer selection methods with great distinction. It achieved an average throughput of 65 Mbps. It indicates the efficient data transmission across the network. In terms of peak throughput, the system reached 85 Mbps, demonstrating its capability to handle high data loads effectively. The system showed low throughput variability which highlighting its consistency in maintaining stable data transmission rates over period of time. The data packet delivery ratio is recorded at 95% for the proposed system. It ensures that the vast majority of data packets are successfully delivered to their destinations. This high delivery ratio shows the system's robustness in handling data traffic. The throughput efficiency stood at 90%, indicating that the system effectively uses its maximum potential throughput. Another important aspect of effective use of connections was also bandwidth usage that reached 88%. It proved that the system allowed using all connections effectively. It showcases the ability of the system to make the most of available network resources. The proposed system completed 80 successful data transfers during the simulation in terms of the number of successful transfers which underscoring its reliability and robustness. It is recorded and shown in figure 11. The data transfer rate of 1024 KB/s further proves the high performance of the system in active data transfers.

The peer selection method presented in the proposed framework demonstrates low latency parameters as compared to the other methods for minimum 100 peers in the network. The proposed system exhibited latency of 140 ms, which is relatively much smaller compared to RPS at 150ms, NS at 170ms, CAS at 220ms, SBA at 300ms, and UBS at 350ms.This reduction in latency demonstrates the system's efficiency in minimizing the time taken for data packets to traverse the network. It is shown in figure 12. Achieved latencies of the proposed system was identified to be 300 ms during its peak, whereas it was 350 ms for RPS, 360 ms for NS, 370 ms for CAS, 400 ms for SBA and 435 ms for UBS. This lower peak latency is indicative of the system's robustness in maintaining low delay even under heavy network traffic conditions. The system also showed small latency jitter at 5 ms. It showcases its capability to provide consistent and predictable performance. In contrast, RPS, NS, CAS, SBA, and UBS showed higher jitter values of 20 ms, 18 ms, 15 ms, 12 ms, and 10 ms correspondingly. Low jitter is crucial for applications requiring real-time data transmission. In addition, the proposed system latency variance was 2 ms of which was less than that of the RPS, NS, CAS, SBA, and UBS with 10, 9, 8, 7, and 6 ms respectively. This lower variance confirms that the system can sustain the identified latency rates, which strengthens the claim of reliability under varying network conditions. It is reported and shown in the figure 13. Concerning the RTT measure, the proposed system has a RTT of 30 ms while the RPS has 80 ms, NS 75 ms, CAS 70 ms, SBA 65 ms and UBS 60 ms. A lower RTT enhances user experience by assuring quicker acknowledgement and also response times. It is reported and shown in the figure 14.

The connectivity parameters of the proposed system are found to be better in performance than the existing systems stated above. The average connectivity of the proposed system is 8 which is much higher than of RPS(4), NS(5), CAS(6), SBA(7) and UBS(6). This increases the service's reliability and the extent of interactions between peers in the network. Additionally, the clustering coefficient for the proposed system was 0.75, compared to 0.3 for RPS, 0.35 for NS, 0.4 for CAS, 0.5 for SBA, and 0.45 for UBS, indicating a stronger tendency for peers to form tightly-knit groups, thereby providing an efficient and more liable facility in terms of the local connectivity and fault tolerance as the peers have more tendency to cluster together. The network diameter, a measure of the longest shortest path between any two nodes, was 5 for the proposed system. It shows that there is a significant improvement over RPS (10), NS (9), CAS (8), SBA (7), and UBS (6). This shorter network diameter implies quicker data transfer and less delay time. It is recorded and shown in figure 15. The amount of average path length in the proposed system was also found to be 3 which was lesser than RPS with 6, NS with 5.5, CAS with 5, SBA with 4.5 and UBS with 4 and therefore reduced overall latency and quicker communication. In addition to these, the proposed system showed higher redundancy and fault tolerance with redundancy factor of 0.85 more than RPS = 0.4, NS = 0.45, CAS = 0.5, SBA = 0.6 and UBS = 0.55. This relative high redundant provides the assurance that the network is still alive even with node failures resulting in low data loss and network connectivity disruption. The proposed system achieved a higher stability rate with connectivity stability being significantly at 95% than that of RPS with 70%,NS with 75%,CAS with 80%,SBA with 85% and UBS with 90%. This high stability rate gives a clear sign that the relations between peers is more stable and that the connection is less likely to be interrupted.

In terms of the resource utilization parameters, the proposed system outperform with compared to traditional systems mentioned above. The proposed system shows better utilization efficiency of CPU at 85% as compared to RPS at 60%, NS at 65%, CAS at 70%, SBA at 75%, and UBS at 80%. This high CPU utilization efficiency ensures that computational resources are optimally used without excessive overhead. Efficient memory usage is crucial for handling large amounts of data and supporting numerous simultaneous peer connections. The proposed system used less memory than RPS with an average of 80 % of memory utilization, NS with 60%, CAS with 65%, SBA with 70 % and UBS with 75 %. The bandwidth utilization concerning the rate with which the proposed system functions was optimal at 90% while that of other systems such as RPS, NS, CAS, SBA, and UBS was much less at a rate of 50%, 55%, 60%, 65% and 70% respectively. It also helps to determine the fact that bandwidth is optimally used so as to improve the flow of data across the network, not to mention reducing incidences of congestion that may hinder performance of the global P2P network. The proposed system also demonstrated better results in the disk I/O utilization rate in which the average value obtained was 85% while that obtained for RPS, NS, CAS, SBA and UBS were 50%, 55%, 60%, 65% and 70% respectively. It is shown in figure 16. Quick retrieval of disk I/O stored data is important to increase the speed and reliability of the network operations. Furthermore, the proposed system's ability to balance load across peers was exceptional, achieving a load balancing efficiency of 90%. Moreover, the performance assessment of the proposed system characterized the load balancing ratios among the peers with a particularly high performance of 90 percent. Thus, this is higher than the efficiency analysed for RPS (50%), NS (55%), CAS (60%), SBA (70%) and UBS (75%). Load management helps overcome bottlenecks within a peer where one peer does not affect the continuous operation of the entire network. It is shown in figure 16 and upload capacity measure is also shown in figure 17.

The proposed approach increased level of QoS satisfaction in comparison with existing systems. This framework obtains 95% reliability and 98% availability, ensuring consistent and dependable service delivery. It far exceeds the performance with compared to other systems. Additionally, the proposed system possesses low jitter, where it only has 5ms and a very low packet loss where it only has 0.5%; it is an important requirement for communication which requires low lag such as video streaming, VoIP etc. Such metrics illustrate the smooth and continuous data transmission which is supported by the system. Also, the proposed system shows a 96% user satisfaction, which strongly supports the applied intelligent combination of FLP and Q-Learning to select peers dynamically and optimally to match changing network situations and users' demands. This combination leads to maximization of resources and guarantees better satisfaction of the QoS which makes the proposed framework a strong solution to the problem of P2P network optimization. It is shown in figure 18.

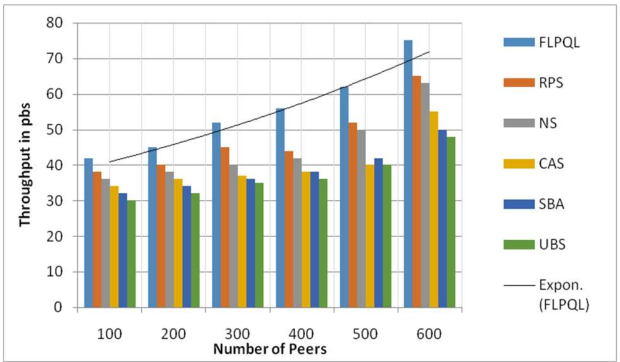

Figure 11 Throughput Measurement

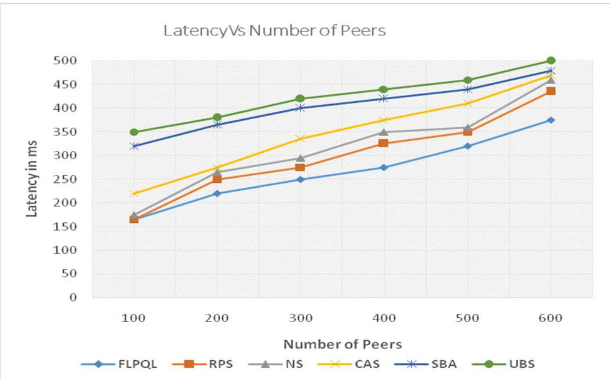

Figure 12 Latency Measure

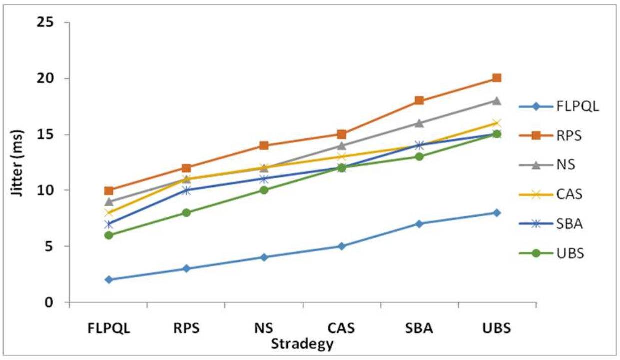

Figure 13. Jitter Measure

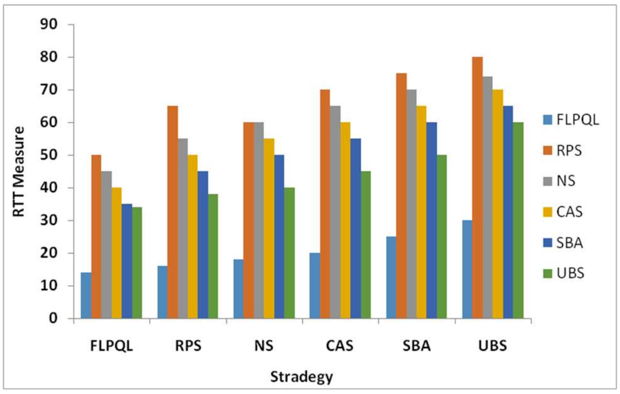

Figure 14. RTT Measure

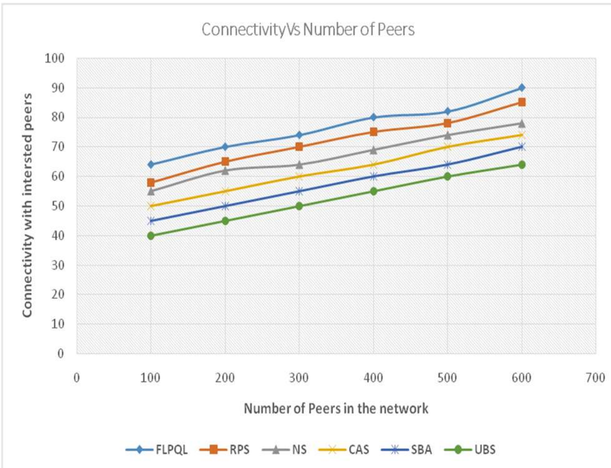

Figure 15 Connectivity Measure

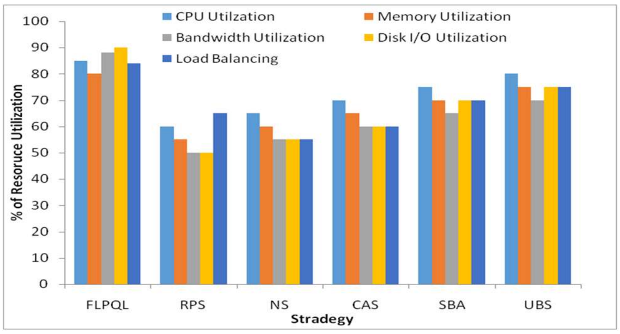

Figure 16 Resource Utilization Measure

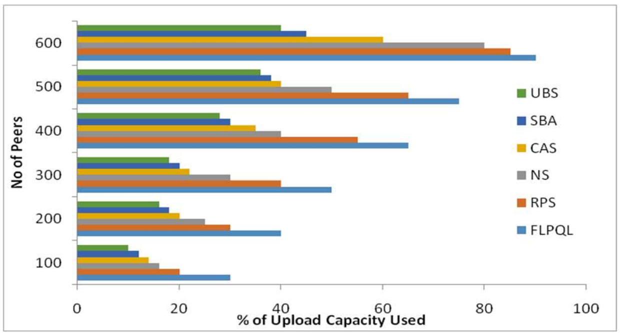

Figure 17 Upload Capacity Measure

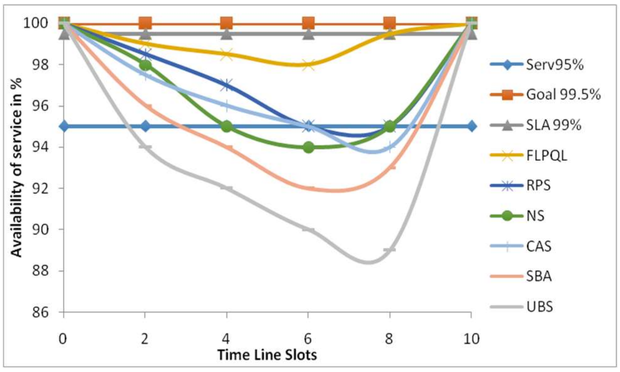

Figure 18 QoS Measure

5. Conclusion

The combination of Fuzzy Linear Programming with Q-Learning can be characterized as an advanced and perspective approach for improving the decision-making process in the conditions of the volatile and unpredictable surrounding space, as for the peer selection of P2P networks. This integrated system is advantageous in several ways as a result of integrating the capacity of FLP in dealing with the imprecise information with the learning flexibility of Q-Learning. Firstly, the integration enables satisfaction of best single compromise solution for many objectives that cannot be met simultaneously while handling with uncertainties peculiar to P2P networks. Thus, FLP conceptually supports the representation of imprecise data to formulate objectives and constraints of optimization models that emulate real life conditions. Furthermore, Q-Learning helps the adaptive learning of the decision-making rules maximizing not only the learning of the network conditions but also the efficiency and robustness of the system for the most variable conditions in the network. Moreover, combining FLP and Q-Learning enhances techniques for selection mechanism between peers by estimating the best action that will yield the highest sum of rewards in the future without overemphasizing the exploration rate as compared to the exploitation rate. Through sequential updates using policies derived from the learnt rewards, the system is able to develop efficient peer-selection policies that work effectively in a dynamical context given uncertainties and varying network conditions. While merging FLP with Q-Learning, this paper has outlined a sound and potent synthesise to hold with the problems of decision making in the uncertain and intricate environments such as P2P networks. Thus, there is considerable potential for the presented integrated approach based on FLP and Q-Learning to improve the efficiency, robustness and adaptability of selection of peers, as well as to increase the effectiveness and reliability of P2P networks' functioning.

Acknowledgment

This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-RG23028).

References

[1] A Gebraselase Befekadu, Helvik Bjarne Emil, Jiang Yuming "Bitcoin P2P Network Measurements: A testbed study of the effect of peer selection on transaction propagation and confirmation times," IEEE Transactions on Network and Service Management. 2022, DOI: doi.org/10.1109/TNSM.2022.3216955

[2] Budhkar, S., Tamarapalli, V. An overlay management strategy to improve QoS in CDN-P2P live streaming systems. Peer-to-Peer Netw. Appl. 13, 190–206 (2020). DOI: doi.org/10.1007/s12083-019-00755-x.

[3] I. -S. Hwang, A. Rianto, R. Kharga and M. S. Ab-Rahman, "Global P2P BitTorrent Real-Time Traffic Over SDN-Based Local-Aware NG-PON2," in IEEE Access, vol. 10, pp. 76884-76894, 2022, DOI: 10.1109/ACCESS.2022.3192439

[4] Yingying Ren, Zhiwen Zeng, Tian Wang, Shaobo Zhang & Guoming Zhi, "A trust-based minimum cost and quality aware data collection scheme in P2P network," P2P

Networking and Applications volume 13, pages2300–2323(2020). DOI: 10.1007/s12083-020-00898-2

[5] S. Nacakli and A. M. Tekalp, "Controlling P2P-CDN Live Streaming Services at SDN-Enabled Multi-Access Edge Datacenters," in IEEE Transactions on Multimedia, vol. 23, pp. 3805-3816, 2021, DOI: doi:%2010.1109/TMM.2020.3032042.

[6] S. Luo, H. Yu, K. Li, and H. Xing, "Efficient file dissemination in data center networks with priority-based adaptive multicast," IEEE Journal on Selected Areas in Communications, vol. 38, no. 6, pp. 1161-1175, 2020. DOI: doi.org/10.1109/JSAC.2020.2986616

[7] H. Yao, Y. Xiang and J. Liu, "Virtual Prosumers P2P Transaction Based Distribution Network Expansion Planning," IEEE Transactions on Power Systems, vol. 39, no. 1, pp. 1044-1057, Jan. 2024, DOI: doi:%2010.1109/TPWRS.2023.3240830

[8] R. Farahani, E. Çetinkaya, C. Timmerer, M. Shojafar, M. Ghanbari and H. Hellwagner, "ALIVE: A Latency-and Cost-Aware Hybrid P2P-CDN Framework for Live Video Streaming,", IEEE Transactions on Network and Service Management, 2023, DOI: doi:%2010.1109/TNSM.2023.3335190

[9] L. Nie, S. Yang, X. Zheng and X. Wang, "An Efficient and Adaptive Content Delivery System Based on Hybrid Network," IEEE Transactions on Broadcasting, vol. 69, no. 4, pp. 904-915, Dec. 2023, DOI: doi:%2010.1109/TBC.2023.3301716

[10] Y. Ren, Z. Zeng, T. Wang, S. Zhang, and G. Zhi, “A trust-based minimum cost and quality aware data collection scheme in P2P network”. Peer-to-Peer Networking and Applications, vol. 13, no. 6, pp. 2300-2323, 2020, DOI: 10.1007/s12083-020-00898-2

[11] R. Farahani, E. Çetinkaya, C. Timmerer, M. Shojafar, M. Ghanbari and H. Hellwagner, "ALIVE: A Latency- and Cost-Aware Hybrid P2P-CDN Framework for Live Video Streaming," in IEEE Transactions on Network and Service Management, vol. 21, no. 2, pp. 1561-1580, April 2024, DOI: doi:%2010.1109/TNSM.2023.3335190

[12] Dharmendra Kumar and Mayank Pandey. 2022. An optimal and secure resource searching algorithm for unstructured mobile peer-to-peer network using particle swarm optimization. Applied Intelligence 52, 13 (Oct 2022), 14988–15005, doi.org/10.1007/s10489-022-03291-z

[13] Safara F, Souri A, Deiman SF, “Super peer selection strategy in peer-to-peer networks based on learning automata,” Int Journal of Communications Systems, 2020, DOI: doi.org/10.1002/dac.4296

[14] M. S. Ali, M. Vecchio, G. D. Putra, S. S. Kanhere, and F. Antonelli, "A Decentralized P2P Remote Health Monitoring System," Sensors, vol. 20, no. 6, p. 1656, 2020., DOI: doi.org/10.3390/s20061656

[15] S. Luo, H. Yu, K. Li, and H. Xing, "Efficient file dissemination in data center networks with priority-based adaptive multicast," IEEE Journal on Selected Areas in Communications, vol. 38, no. 6, pp. 1161-1175, 2020. DOI: doi.org/10.1109/JSAC.2020.2986616

[16] X. F. Meng, “speedTrust: a super peer-guaranteed trust model in hybrid P2P networks,” The Journal of Supercomputing, vol. 74, no. 6, pp. 2553–2580, 2018, DOI: doi.org/10.1007/s11227-018-2286-9

[17] Geng J, Fujita S. Enhancing Crowd-Sourced Video Sharing through P2P-Assisted HTTP Video Streaming. Electronics. 2024; Vol. 13, issue no. 7, pp 1265 – 1275, DOI: doi.org/10.3390/electronics13071270

[18] B. Xue, Y. Mao, S. B. Venkatakrishnan and S. Kannan, "Goldfish: Peer Selection using Matrix Completion in Unstructured P2P Network," 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Dubai, United Arab Emirates, 2023, pp. 1-9, DOI: 10.1109/ICBC56567.2023.10174871

[19] Hossein Ghasemkhani & Yung-Ming Li & Kamran Moinzadeh & Yong Tan, 2018. "Contracting Models for P2P Content Distribution," Production and Operations Management, Production and Operations Management Society, vol. 27(11), pages 1940-1959, November. DOI: doi.org/%2010.1111/poms.12718

[20] Dharmendra Kumar, Mayank Pandey, “An optimal and secure resource searching algorithm for unstructured mobile peer-to-peer network using particle swarm optimization,” Applied Intelligence, Volume 52, Issue 13, 2022, pp 14988–15005, DOI: 10.1007/s10489-022-03291-

[21] M. Anandaraj, K. Selvaraj, P. Ganeshkumar, K. Rajkumar and S.Sriram, Genetic Algorithm-Based Resource Minimization in Network Code-Based P2P Network, Journal of Circuits, Systems, and Computers Vol. 30, No. 8 , 2021, DOI: 10.1142/S0218126621500924

[22] S. Naganandhini and D. Shanthi, "Optimizing Replication of Data for Distributed Cloud Computing Environments: Techniques, Challenges, and Research Gap," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 35-41, DOI: 10.1109/ICECAA58104.2023.10212287

[23] Yingying Ren, Zhiwen Zeng, Tian Wang, Shaobo Zhang & Guoming Zhi, “A trust-based minimum cost and quality aware data collection scheme in P2P network,” P2P Networking and Applications volume 13, pages2300–2323, 2020, DOI: 10.1007/s12083-020-00898-2

[24] Xiao-Peng Yang, Gengzhong Zheng, “Maximum number of line faults in a P2P network system based on the addition-min fuzzy relation inequalities”, IEEE Transactions on Fuzzy Systems 2021, DOI: doi.org/10.1109/TFUZZ.2021.3078529

[25] Goguen JA. L. A. Zadeh. Fuzzy sets. Information and control, vol. 8 (1965), pp. 338–353. - L. A. Zadeh. Similarity relations and fuzzy orderings. Information sciences, vol. 3 (1971), pp. 177–200. Journal of Symbolic Logic. 1973;38(4):656-657. DOI: 10.2307/2272014

[26] A. -T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares and M. Sugeno, "Fuzzy Control Systems: Past, Present and Future," in IEEE Computational Intelligence Magazine, vol. 14, no. 1, pp. 56-68, Feb. 2019, DOI: 10.1109/MCI.2018.2881644.

[27] Yi Liu, Shinji Sakamoto, Keita Matsuo, Makoto Ikeda, Leonard Barolli & Fatos Xhafa, “A comparison study for two fuzzy-based systems: improving reliability and security of JXTA-overlay P2P platform,” International journal of Soft Computing volume 20, pages2677–2687, 2016.

[28] G. Zhang, S. Chai, R. Chai, M. Garcia and Y. Xia, "Fuzzy Goal Programming Algorithm for Multi-Objective Trajectory Optimal Parking of Autonomous Vehicles,"

in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 1909-1918, Jan. 2024, DOI: doi:%2010.1109/TIV.2023.3311536

[29] S. H. Nasseri, J. L. Verdegay and F. Mahmoudi, "A New Method to Solve Fuzzy Interval Flexible Linear Programming Using a Multi-Objective Approach," in Fuzzy Information and Engineering, vol. 13, no. 2, pp. 248-265, June 2021, DOI: 10.1080/16168658.2021.1906154

[30] Rivaz, S., Nasseri, S. H., & Ziaseraji, M. (2020). A Fuzzy Goal Programming Approach to Multiobjective Transportation Problems. Fuzzy Information and Engineering, 12(2), 139–149. DOI: 10.1080/16168658.2020.1794498

[31] Zhang, L. (2024). Max-min fuzzy bi-level programming: resource sharing system with application. Applied Mathematics in Science and Engineering, 32(1). DOI: 10.1080/27690911.2024.2335319.

[32] Shoab, M., Jubayrin, S.A. Intelligent neighbor selection for efficient query routing in unstructured P2P networks using Q-learning. *Appl Intell* **52**, 6306–6315 (2022). https://doi.org/10.1007/s10489-021-02793-6.

[33] Mohammad Shoab and Saad Al Jubayrin. 2022. Intelligent neighbor selection for efficient query routing in unstructured P2P networks using Q-learning. Applied Intelligence 52, 6 (Apr 2022), 6306–6315. https://doi.org/10.1007/s10489-021-02793-6.

[34] Anandaraj M, Ganeshkumar P, Naganandhini S, Selvaraj K. A novel fuzzy programming approach for piece selection problem in P2P content distribution network. PeerJ Comput Sci. 2024 Jan 3; DOI: 10:e1645. 10.7717/peerj-cs.1645.

[35] Yu, Y., Qin, Y. & Gong, H. A Fuzzy Q-Learning Algorithm for Storage Optimization in Islanding Microgrid. J. Electr. Eng. Technol. 16, 2343–2353 (2021). https://doi.org/10.1007/s42835-021-00769-7

M. Anandaraj is a Professor in the Department of Information Technology at PSNA College of Engineering and Technology, located in Dindigul, Tamilnadu, India. With extensive expertise in the field of information technology, Dr. Anandaraj has made significant contributions to academia and research. His work is recognized for its depth and impact, reflecting his commitment to advancing knowledge and innovation in technology.

Tahani Albalawi is an Assistant Professor of Computer Science at Imam University, specializing in information security and privacy. She completed her master’s and Ph.D. in Computer Science at Kent State University in 2018. Her research interests are in information security, privacy, and network security. She has published several papers in top-tier conferences and journals. In addition to her research, she is passionate about teaching and actively mentors undergraduate and graduate students in the field of information security and privacy.