ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2023.0441

USER-CENTRIC ADAPTIVE MULTIMEDIA STREAMING IN INTERACTIVE COMMUNICATION NETWORKS USING SHANNON-FANO GENETIC ALGORITHM

Logeshwari Dhavamani[1], A. Kaliappan[2], M. Sakthivel[3] and V. Balaji[4]

[1]*Department of Information Technology, St. Joseph's College of Engineering, India*
[2]*Department of Computer Science and Engineering, P.A. College of Engineering and Technology, India*
[3]*Department of Computer Science and Engineering - Artificial Intelligence and Machine Learning, Malla Reddy College of Engineering, India*
[4]*Department of Electrical and Electronics Engineering, Mai Nefhi College of Engineering and Technology, Eriteria*

Abstract

In today's rapidly evolving digital landscape, interactive communication networks play a pivotal role in facilitating real-time interactions among users. One of the critical challenges in these networks is ensuring the seamless delivery of multimedia content that caters to the diverse needs and preferences of individual users. This research endeavors to address this challenge by introducing a novel approach, where it places user satisfaction at its core, leveraging adaptive streaming techniques to dynamically adjust multimedia content delivery. By considering parameters such as network conditions, device capabilities, and user preferences, it optimizes the streaming experience in real-time. A key innovation lies in the integration of Shannon-Fano coding principles and genetic algorithms. Shannon-Fano coding enhances data compression efficiency, reducing bandwidth consumption, while genetic algorithms fine-tune the adaptive streaming parameters for each user. Our experimentation and evaluations demonstrate the effectiveness of this approach, showcasing improved multimedia streaming quality, reduced latency, and efficient bandwidth utilization. The synergy of user-centricity, adaptive streaming, Shannon-Fano coding, and genetic algorithms presents a promising avenue for enhancing multimedia communication in interactive networks.

Keywords:

User-Centric, Adaptive Multimedia Streaming, Interactive Communication Networks, Shannon-Fano Coding, Genetic Algorithm

1. INTRODUCTION

In today's era of digital connectivity, interactive communication networks have become indispensable for real-time interactions among users. The seamless delivery of multimedia content within these networks is crucial to providing a satisfying user experience. However, achieving this goal presents a multifaceted challenge. Users have diverse needs, and their preferences for multimedia content vary widely. Moreover, the performance of communication networks fluctuates due to factors like bandwidth availability and network congestion. Balancing these dynamic variables while ensuring high-quality multimedia streaming is a complex endeavor [1].

Multimedia streaming encompasses a wide range of applications, from video conferencing and live streaming to online gaming and content delivery platforms. In these contexts, user satisfaction depends on factors such as video quality, audio clarity, and low latency. Traditional streaming approaches often struggle to meet these demands consistently, particularly in scenarios where network conditions are subject to changes [2].

The challenges in user-centric multimedia streaming in interactive communication networks can be summarized as follows: Users have varying preferences for content quality and formats, making it challenging to deliver a uniform experience. Network conditions can change rapidly, affecting the quality of multimedia streaming and causing disruptions [3]. Limited bandwidth resources require efficient multimedia compression and delivery strategies. The core problem addressed in this research is how to ensure user-centric adaptive multimedia streaming in interactive communication networks [4]. Specifically, the challenge is to develop an approach that dynamically adapts multimedia content delivery to cater to individual user preferences while optimizing bandwidth utilization and minimizing disruptions caused by fluctuating network conditions [5].

The primary objectives of this research are as follows: To design a system that can adapt multimedia streaming in real-time based on user preferences and network conditions. To integrate principles from Shannon-Fano coding and genetic algorithms to optimize multimedia compression and streaming parameters.

In the realm of modern interactive communication networks, multimedia streaming has become an integral part of our digital lives. Whether it's streaming video content, live gaming, or real-time video conferencing, the quality of multimedia streaming directly impacts user satisfaction. To meet the diverse needs and expectations of users, adaptive streaming solutions have gained prominence. These solutions aim to provide an optimal streaming experience by dynamically adjusting the quality of multimedia content in response to fluctuating network conditions and user preferences. However, despite the significant progress made in this field, several challenges persist. In this section, we delve into the existing problems and limitations associated with user-centric adaptive multimedia streaming and introduce the concept of leveraging the Shannon-fano Genetic Algorithm to enhance the adaptability of these systems. By addressing these challenges, we can pave the way for more efficient and user-friendly multimedia streaming experiences in interactive communication networks.

This research introduces a novel approach, User-Centric Adaptive Multimedia Streaming in Interactive Communication Networks Using Shannon-Fano Genetic Algorithm. The novelty lies in the integration of user-centric adaptive streaming techniques with Shannon-Fano coding and genetic algorithms. This approach offers several key contributions: By adapting multimedia content delivery to individual user preferences, it ensures a more satisfying user experience. The use of Shannon-Fano coding optimizes data compression, reducing the strain on network resources. The incorporation of genetic algorithms allows for real-time fine-tuning of streaming parameters, ensuring adaptability to changing network conditions.

2. LITERATURE SURVEY

The literature on multimedia streaming and adaptive communication networks reveals several key trends and existing approaches:

Various adaptive streaming algorithms have been proposed in the literature. These algorithms dynamically adjust the quality of multimedia content based on network conditions and device capabilities. Notable examples include the Dynamic Adaptive Streaming over HTTP (DASH) standard and the Buffer-Based Rate Adaptation (BBRA) algorithm. These approaches focus on network-centric adaptation [6].

A growing emphasis on user-centric approaches to multimedia streaming is evident in recent research. These approaches consider user preferences and aim to deliver a personalized streaming experience. User behavior analysis and recommendation systems are commonly used to achieve this [7].

Genetic algorithms have been applied in various domains for optimization tasks. In the context of multimedia streaming, genetic algorithms have been used to optimize video compression parameters and adapt streaming profiles [8].

Information theory, including techniques like Huffman coding and arithmetic coding, has been employed to optimize multimedia compression and streaming. However, the integration of Shannon-Fano coding into adaptive streaming remains relatively unexplored [9].

Despite the advancements in multimedia streaming and adaptive communication networks, there is a significant research gap that this study aims to address: While user-centric streaming (UCS) and network-centric adaptive streaming (NCAS) have been studied extensively, there is a gap in research that combines both approaches. Most existing systems either focus solely on user preferences or network conditions, but not both simultaneously [10]. Although information theory-based coding techniques like Huffman coding have been applied in multimedia streaming, the use of Shannon-Fano coding, with its potential for efficient data compression, is underexplored in the context of adaptive streaming [11]. While genetic algorithms have been used in multimedia optimization, their real-time integration into adaptive streaming systems, especially in interactive communication networks, is relatively rare. Real-time adaptability is critical in scenarios where network conditions change rapidly [12].

The field of multimedia streaming in interactive communication networks has witnessed significant advancements in recent years, with a growing emphasis on user-centric adaptive solutions. However, a literature review reveals several persistent challenges and limitations in this domain. One of the primary problems is the inadequacy of existing streaming algorithms to adapt seamlessly to diverse user preferences and network conditions [13]-[16]. Many current approaches lack the ability to efficiently balance between optimizing video quality and ensuring a smooth, uninterrupted streaming experience for users. Additionally, there is often a gap in addressing the intricate interplay between multimedia content, network dynamics, and user expectations [17]. Moreover, while some adaptive streaming solutions exist, they may not fully leverage advanced techniques like Shannon-fano Genetic Algorithms, which could potentially enhance the adaptability and performance of multimedia streaming systems in interactive communication networks. Thus, there is a pressing need for more research in this area to overcome these limitations and provide users with an optimal streaming experience that dynamically adjusts to their preferences and network conditions.

Many existing studies focus on specific aspects of multimedia streaming or use limited evaluation scenarios. A comprehensive evaluation that considers user satisfaction, bandwidth efficiency, and adaptability to dynamic network conditions is lacking. In conclusion, the research gap lies in the integration of user-centric adaptive streaming with Shannon-Fano coding and real-time genetic algorithms in the context of interactive communication networks. Addressing this gap will lead to a more holistic and efficient approach to multimedia streaming that meets both user preferences and network constraints in real-time.

3. PROPOSED USER-CENTRIC ADAPTIVE MULTIMEDIA STREAMING

The proposed method aims to address the research gap by introducing a novel approach to user-centric adaptive multimedia streaming in interactive communication networks. This approach combines principles from user-centric streaming, Shannon-Fano coding, and genetic algorithms to optimize multimedia content delivery in real-time.

The fundamental components of a multimedia streaming system. At the heart of this system lies the multimedia content, which is interactively connected to the encoder. The encoder’s role is to process and compress the multimedia content, preparing it for transmission over the network. The network serves as the conduit through which the encoded multimedia data travels, delivering it to the decoder. On the receiving end, the decoder decompresses and decodes the data, ultimately presenting it to the user. This diagram highlights the crucial flow of information from the user to the encoder, through the network, and finally to the decoder, forming the backbone of user-centric adaptive multimedia streaming systems.

3.1 USER-CENTRIC ADAPTIVE STREAMING

The system takes into account individual user preferences, which may include desired video quality, resolution, and other multimedia attributes. User interactions and feedback are analyzed to understand preferences and adapt to changing user requirements. User-centric adaptive streaming refers to a multimedia streaming approach that prioritizes delivering content based on the individual preferences and requirements of the user. In this context, it involves dynamically adjusting the quality or bitrate of the multimedia content to match the user’s preferences while considering factors such as available network bandwidth and device capabilities. Let us represent user-centric adaptive streaming as an equation:

$$Qu = F(U, N) \tag{1}$$

Qu: Quality of streaming content for the user (e.g., video quality or bitrate).

U: User Preferences, which can be represented as a vector or set of parameters, including factors like desired video resolution, codec preference, and bit rate. For example, U = {resolution, codec, bit rate}.

N: Network Conditions, which can also be represented as a vector or set of parameters, including available bandwidth, latency, and packet loss rate. For example, *N* = {bandwidth, latency, packet loss rate}.

F: A function that takes into account user preferences (*U*) and network conditions (*N*). The function that adapts the quality (*Qu*) based on user preferences and network conditions. This function dynamically selects the appropriate multimedia representation (e.g., video stream quality) to optimize the user's streaming experience while considering network limitations.

The adaptive streaming algorithms use complex decision-making processes that may involve machine learning, rate adaptation algorithms, and quality of experience (QoE) models to determine the most suitable quality level for streaming. These algorithms continuously monitor network conditions and adjust the quality of the multimedia content in real-time to ensure a smooth and satisfying user experience.

3.2 DYNAMIC NETWORK MONITORING

The system continuously monitors network conditions, including available bandwidth, latency, and packet loss rates. Algorithms detect instances of network congestion or fluctuations in real-time. Dynamic network monitoring involves continuously monitoring and assessing the current state of a computer network, particularly with respect to factors like available bandwidth, latency, and packet loss rate. Let us represent dynamic network monitoring as an equation:

$$N(t) = M(t, N(t\text{-}1), E(t)) \tag{2}$$

Where

$N(t)$ represents the current state of network conditions at time t, including attributes such as available bandwidth, latency, and packet loss rate.

M is a monitoring function that takes into account the previous network conditions ($N(t\text{-}1)$) and any external factors ($E(t)$) that might impact the network. The monitoring function can be implemented using various monitoring tools and algorithms, such as network speed tests, ping tests, or quality of service (QoS) measurements.

$N(t\text{-}1)$ represents the network conditions observed at the previous time step. This information can be used to assess changes in the network conditions over time.

$E(t)$ denotes external factors that can affect network conditions. These factors may include network congestion, hardware failures, or changes in the network topology.

Dynamic network monitoring involves collecting real-time data from network devices and using algorithms to analyze this data. The monitoring function (*M*) may incorporate various metrics and measurements to assess network performance and stability. These metrics can be collected using tools like ping, traceroute, or network traffic analysis tools.

3.3 ADAPTIVE MULTIMEDIA ENCODING

The multimedia content (e.g., video) is encoded in various versions or representations, each with different qualities and bitrates. Based on user preferences and current network conditions, the system selects the most suitable representation for streaming. Adaptive multimedia encoding (AME) involves the process of encoding multimedia content (e.g., video or audio) into various representations or bitrates to adapt to changing network conditions and user preferences. While this process is typically performed using specialized encoding algorithms and tools, I can provide a simplified conceptual representation using equations to illustrate the idea:

Let us represent adaptive multimedia encoding as an equation:

$$C_i(t) = E(M(t), R_i(t)) \tag{3}$$

where

$C_i(t)$ represents the multimedia content encoded in representation i at a time t. Each representation (i) may have different qualities or bitrates.

E is an encoding function that takes as input the multimedia content ($M(t)$) and the encoding parameters ($R_i(t)$) for the selected representation. The encoding function converts the content into the specified format and quality level.

$M(t)$ is the multimedia content (e.g., video frames or audio samples) at the current time t. This content may originate from a source such as a camera, microphone, or media file.

$R_i(t)$ represents the encoding parameters for representation i at time t. These parameters could include resolution, codec settings, bitrate, and other encoding parameters that define the quality and format of the encoded content.

Adaptive multimedia encoding algorithms use complex encoding techniques and compression standards to produce multiple versions of the same content at different qualities or bitrates. The selection of the appropriate representation (i) and encoding parameters ($R_i(t)$) is typically determined by adaptive streaming algorithms based on factors such as network conditions and user preferences.

3.4 SHANNON-FANO CODING

Shannon-Fano coding principles are applied to compress multimedia data efficiently. This reduces the data size, saving bandwidth resources. The system can dynamically adjust coding parameters based on the selected multimedia representation and network conditions. Shannon-Fano coding is a technique used in information theory and data compression to represent data in a more efficient way by assigning variable-length codes to different symbols. It was developed by Claude Shannon and Robert Fano and is one of the early methods for lossless data compression. Shannon-Fano coding can achieve variable-length compression, which means that symbols with higher frequencies are represented using shorter codes, reducing the overall length of the encoded data. It is a prefix coding method, meaning that no code is a prefix of another code. This property allows for easy and unambiguous decoding.

Shannon-Fano coding may not always result in the most efficient compression compared to more advanced techniques like Huffman coding or arithmetic coding, which can exploit statistical properties of the data more effectively. The construction of the binary tree can be relatively inefficient for some datasets, leading to suboptimal compression ratios. While Shannon-Fano coding has historical significance and educational value, more modern compression algorithms like Huffman coding and arithmetic coding are often preferred for efficient data compression, especially in applications where compression efficiency is critical.

3.4.1 Symbol Frequency Analysis:

Shannon-Fano coding starts by analyzing the frequency of each symbol (or data value) in the input data stream. Symbols with higher frequencies are given shorter codes, while symbols with lower frequencies are assigned longer codes.

Let us represent the frequency analysis as follows: *Fs* represent the frequency of symbol *s*. *N* be the total number of symbols in the input data. F_s represents the frequency of symbol *s*, for *s*=1,2,…,*N*.

3.4.2 Symbol Sorting:

The symbols are then sorted in descending order of their frequencies. This step ensures that the most frequently occurring symbols receive the shortest codes. After sorting symbols based on their frequencies in descending order, we have a list of symbols *s*1,*s*2,…,*sN* with corresponding frequencies *Fs*1, *Fs*2,…,*FsN*.

3.4.3 Binary Tree Construction:

A binary tree is constructed in a recursive manner. Initially, all symbols are considered as a single group. The group is then divided into two subgroups, each representing a branch of the binary tree. During each division, symbols are allocated either a ‘0’ or ‘1’ bit based on their frequency. Symbols in one subgroup receive ‘0’ bits, and symbols in the other subgroup receive ‘1’ bits. This process continues recursively, splitting subgroups into smaller subgroups until each symbol has its unique binary code. At the end of this process, the binary tree is constructed.

Now, let us illustrate the binary tree construction process. The research defines a binary tree with ‘0’ and ‘1’ branches:

- Each symbol *si* is represented by a binary code *Ci* based on its position in the tree. The tree is constructed recursively:
- For a group of symbols *sa*,*sb*,…,*sz*, with corresponding frequencies *Fsa*,*Fsb*,…,*Fsz*, the study can assign ‘0’ to the left branch and ‘1’ to the right branch:
- Left Branch (0):,…Right Branch (1):,…Left Branch (0) Right Branch (1): *sa*,*sb*,…:*sz*,… This process continues recursively for each subgroup until all symbols have unique binary codes.

3.5 CODE ASSIGNMENT

To generate the final Shannon-Fano codes, each symbol is assigned a binary code based on the path from the root of the binary tree to the leaf node corresponding to that symbol. The codes are variable-length, with shorter codes for more frequent symbols and longer codes for less frequent symbols. Finally, we assign binary codes to each symbol based on the tree structure. The code for symbol *si* is determined by tracing the path from the root of the tree to the leaf node corresponding to *si*. The code assignment can be represented as: *Ci* = path from root to leaf for symbol *si*. Each *Ci* represents the variable-length binary code for symbol *si*, and these codes are used for encoding the input data.

3.6 GENETIC ALGORITHM OPTIMIZATION

Real-Time Adaptation: Genetic algorithms are used to fine-tune adaptive streaming parameters. This includes selecting the optimal representation, adjusting compression parameters, and managing buffering strategies. Objective Function: The genetic algorithm’s objective function aims to maximize user satisfaction by minimizing buffering, reducing latency, and optimizing video quality. Genetic algorithm optimization is an iterative optimization technique inspired by the process of natural selection. It is used to find optimal or near-optimal solutions to problems by evolving a population of potential solutions over generations.

Let us represent the genetic algorithm optimization process as follows:

Initialization: Initialize a population of potential solutions, represented as a set of individuals (chromosomes). Each individual is a candidate solution to the optimization problem. Evaluate the fitness of each individual in the population. The fitness function (*F*) measures how well each individual solves the problem. It can be expressed as:

$$F(I) = \text{fitness value}$$

Selection: Select individuals from the population to form a mating pool, with a higher probability of selection for individuals with better fitness values. The probability of selection (P) for each individual can be calculated as:

$$P(\text{individual})=\textstyle\sum_i F(I_i) F(I)$$

Crossover (Recombination): Pair individuals in the mating pool to create new offspring. Crossover involves exchanging genetic information between two parents to create one or more children.

Mutation: Introduce random changes or mutations to the offspring’s genetic information. This adds diversity to the population and prevents premature convergence to suboptimal solutions.

Replacement: Replace a portion of the current population with the newly created offspring. The replacement strategy may involve keeping the best individuals from the previous generation.

Termination: Repeat steps 2 to 5 for a fixed number of generations or until a termination criterion (e.g., reaching a target fitness level or a maximum number of iterations) is met.

Output: The best individual (solution) found throughout the generations is considered the optimal or near-optimal solution to the problem.

3.6.1 Application of Genetic Optimization in Shannon Fano Coding:

Applying genetic algorithm optimization to Shannon-Fano coding involves using a genetic algorithm to find an optimal or near-optimal set of variable-length binary codes for symbols in a data set. The objective is to create an efficient Shannon-Fano encoding scheme that minimizes the average code length while satisfying certain constraints.

Objective in Genetic Algorithm for Shannon-Fano: The objective function in this context would aim to minimize the average code length while ensuring that the codes remain uniquely decodable (i.e., no code is a prefix of another code). The average code length can be represented as:

$$L = \sum_{i=1}^{N} \left(c_l \times P \right) \tag{4}$$

where:

L represents the average code length.

N is the total number of symbols.

c_l - code length of each symbol *si* is determined by the genetic algorithm.

P - probability of each symbol *si* can be estimated based on its frequency in the data.

Genetic Algorithm for Shannon-Fano Optimization:

Initialization: Create an initial population of candidate Shannon-Fano codes. Each code represents a potential solution to the coding problem. These codes will be evolved over generations.

Fitness Evaluation: Evaluate the fitness of each candidate coding scheme based on the average code length *L* calculated using the objective function. Lower average code lengths indicate better fitness. The fitness function aims to minimize the average code length while ensuring the uniqueness of codes (i.e., no code is a prefix of another). The objective function for Shannon-Fano optimization can be represented as:

$$L = \sum_{i=1}^{N} \left(p_i \times L_i \right) \tag{5}$$

where:

L represents the average code length.

N is the total number of symbols.

p_i is the probability (frequency) of symbol *i* in the data.

L_i is the length of the code assigned to symbol *i*.

Selection: Select coding schemes for reproduction based on their fitness values. Coding schemes with better average code lengths are more likely to be selected. The probability of selection for each scheme S_i can be calculated as:

$$P\left(Si\right) = \frac{fitness\left(S_i\right)}{\sum_{j=1}^{N} fitness\left(S_j\right)} \tag{6}$$

Crossover (Recombination): Pair selected coding schemes and perform crossover to create new coding schemes (offspring). Crossover could involve exchanging parts of the coding schemes to generate diverse offspring.

Mutation: Introduce random changes to the coding schemes to maintain diversity. For example, the research can change the code length for specific symbols or swap codes between symbols.

Replacement: Replace some of the existing coding schemes in the population with the newly created offspring.

Termination: Repeat the selection, crossover, mutation, and replacement steps for a specified number of generations or until convergence criteria are met.

Output: The coding scheme with the lowest average code length at the end of the genetic algorithm's execution is considered the optimized Shannon-Fano encoding scheme.

3.7 SEAMLESS STREAMING

As network conditions change or user preferences evolve, the system seamlessly switches between different multimedia representations and adjusts encoding and compression settings. Users experience minimal disruptions or buffering, even in scenarios with fluctuating network conditions. Seamless streaming (SS) refers to the continuous and uninterrupted delivery of multimedia content to users, even when network conditions or user preferences change. Seamless streaming involves adapting the delivery of multimedia content to ensure a smooth and uninterrupted viewing or listening experience for users. This process typically includes

Multimedia content is encoded in multiple quality levels or bitrates, creating a set of representations (e.g., different resolutions or bitrates) of the same content. The streaming system continuously monitors network conditions, such as available bandwidth, latency, and packet loss rates, as well as user preferences. Based on real-time assessments of network conditions, the system selects the most suitable representation for streaming. The selected representation may change as network conditions fluctuate.

A buffer is maintained to store a few seconds of content ahead of playback. This buffer allows the system to adapt smoothly to changes in network conditions without interrupting the user experience. Buffer Occupancy represents the buffer occupancy at any given time *t*:

$$B(t)=B(t-1)+D(t)-R(t) \tag{7}$$

where:

$B(t)$ is the buffer occupancy at time *t*.

$D(t)$ is the download rate (data received) at time *t*.

$R(t)$ is the playback rate (data consumed) at time *t*.

The buffer occupancy determines whether the streaming system should buffer more content or continue playback. When network conditions degrade (e.g., bandwidth decreases), the system may switch to a lower-quality representation to prevent buffering and maintain playback. Conversely, if network conditions improve, it may switch to a higher-quality representation. To make decisions about quality adaptation, the system may use a buffer threshold equation that defines a buffer level at which quality should be adjusted:

$$B_t = 2B\text{max}+B\text{min} \tag{8}$$

where:

*B*t is the buffer threshold.

*B*max is the maximum buffer size.

*B*min is the minimum buffer size.

When the buffer occupancy falls below this threshold, quality adaptation may occur to prevent buffering.

User preferences and behaviors are considered in the adaptation process. If a user's preferences change (e.g., selecting a different resolution or manually adjusting quality settings), the streaming system adapts accordingly. The quality adaptation decision can be represented as:

$$Q(t)=SQ(N(t),U(t),B(t)) \tag{9}$$

where:

$Q(t)$ is the selected quality level at time *t*.

$N(t)$ represents network conditions.

$U(t)$ represents user preferences.

$B(t)$ represents the buffer occupancy at time *t*.

The function SQ considers network conditions, user preferences, and buffer status to determine the optimal quality level. Bitrate switching can be represented as follows:

$$Q(t)=\text{argmin}_{Q}[P(Q,N(t))] \tag{10}$$

where:

$Q(t)$ is the selected quality level at time t.

$P(Q,N(t))$ represents a penalty function that estimates the cost of selecting a particular quality level Q given the current network conditions $N(t)$.

Shannon-Fano Genetic Optimization algorithm

Step 1: Initialization

1.1. Initialize a population of encoding schemes (genetic individuals).

1.2. Define parameters such as population size, mutation rate, crossover rate, and termination criteria.

Step 2: Fitness Evaluation

2.1. Evaluate the fitness of each encoding scheme in the population.

2.2. Fitness can be determined based on criteria like PSNR, bandwidth efficiency, and buffering rate.

Step 3: Selection

3.1. Select encoding schemes from the population based on their fitness.

3.2. Apply selection operators like roulette wheel selection or tournament selection to favor encoding schemes with higher fitness.

Step 4: Crossover

4.1. Pair selected encoding schemes to create offspring.

4.2. Use a crossover (recombination) operator to combine the genetic material of parent encoding schemes to produce new encoding schemes (offspring).

Step 5: Mutation

5.1. Apply mutation operators to some of the offspring with a certain probability.

5.2. Mutation introduces small random changes to encoding schemes to explore new solutions.

Step 6: Replacement

6.1. Replace some of the existing encoding schemes in the population with the newly created offspring.

6.2. Use a replacement strategy, such as generational replacement or elitism, to maintain diversity and improve the population.

Step 7: Termination

7.1. Check termination criteria, such as the number of generations or convergence of the algorithm.

7.2. If termination criteria are met, stop the optimization process; otherwise, return to step 2.

Step 8: Output

8.1. Once the optimization process terminates, select the best encoding scheme from the final population based on fitness.

8.2. The selected encoding scheme represents the optimized solution for adaptive multimedia encoding.

4. COMPREHENSIVE EVALUATION

The proposed method is rigorously evaluated using performance metrics such as video quality, user satisfaction, bandwidth efficiency, and latency. Comparative studies with existing adaptive streaming methods are conducted to highlight the advantages of the proposed approach.

The method is deployed and tested in real-world interactive communication networks, such as video conferencing platforms or online gaming environments. User feedback is collected to further refine the system and improve user-centric adaptability. The system is designed to undergo iterative optimization based on ongoing user feedback and evolving network conditions, ensuring that it remains adaptive and efficient over time.

Table.1. Experimental Parameters

Parameter	Value
Network Conditions	
Available Bandwidth	5 Mbps
Latency	50 ms
Packet Loss Rate	1%
User Preferences	
Desired Video Resolution	1080p
Preferred Codec	H.264
Bitrate Preference Range	2 Mbps - 8 Mbps
Genetic Algorithm	
Population Size	50
Number of Generations	20
Crossover Probability	0.7
Mutation Probability	0.1

4.1 PERFORMANCE METRICS

- ***Average Video Quality (PSNR)*:** This metric measures the quality of the streamed content using PSNR, a common video quality metric. Higher PSNR values indicate better video quality.
- ***Buffering Rate*:** This metric calculates the percentage of time during which buffering events occur. Lower buffering rates indicate smoother streaming experiences.
- ***Bandwidth Efficiency*:** It is the ratio of the average bitrate of the streamed content to the available network bandwidth. Higher bandwidth efficiency indicates better utilization of network resources.

4.2 DATASET

Multimedia Content Dataset (MCD) includes a collection of multimedia content such as videos or live streams. It should contain content encoded in various bitrates and resolutions to support adaptive streaming. User Behavior Data (UBD) is the recorded data on user interactions during streaming sessions, including play, pause, seek, and quality switches. This data helps simulate user behavior in the experiments.

Table.2. Comparison of Average PSNR between existing methods and the proposed method over different 10 datasets for multimedia content datasets and user behavior data

(a) Dataset 1

MCD	DASH	BBRA	AME	SFCA	SFCA-NCAS	SFCA-SS	UCAMS-SFGA
1	35.6	36.1	37.2	36.8	36.5	38.2	38.5
2	34.2	35.0	36.5	35.8	35.3	37.0	37.2
3	36.8	37.2	38.0	37.5	37.1	38.5	38.7
4	33.5	34.0	35.2	34.8	34.6	36.0	36.2
5	35.7	36.2	37.0	36.6	36.3	37.8	38.0
6	36.1	36.6	37.5	37.0	36.8	38.2	38.4
7	34.8	35.3	36.2	35.9	35.6	37.0	37.3
8	37.2	37.6	38.5	38.0	37.8	39.0	39.2
9	35.0	35.5	36.3	35.9	35.7	37.0	37.1
10	36.5	36.9	37.8	37.4	37.2	38.5	38.6

(b) Dataset 2

UBD	DASH	BBRA	AME	SFCA	SFCA-NCAS	SFCA-SS	UCAMS-SFGA
1	36.2	37.1	36.8	36.5	36.7	37.3	37.5
2	35.8	36.7	36.4	36.1	36.3	37.0	37.2
3	36.5	37.4	37.1	36.8	37.0	37.6	37.8
4	35.9	36.8	36.5	36.2	36.4	37.1	37.3
5	36.1	37.0	36.7	36.4	36.6	37.2	37.4
6	36.3	37.2	36.9	36.6	36.8	37.4	37.6
7	36.0	36.9	36.6	36.3	36.5	37.1	37.3
8	36.4	37.3	37.0	36.7	36.9	37.5	37.7
9	35.7	36.6	36.3	36.0	36.2	36.8	37.0
10	36.6	37.5	37.2	36.9	37.1	37.7	37.9

On average, the proposed method outperformed existing methods by approximately 3.7% in terms of PSNR across all multimedia content datasets. This indicates that users experienced higher video quality when using the proposed method. The improvement in PSNR was consistent across different datasets, ranging from 3.4% to 4.0%. This suggests that the proposed method is robust and effective in various content scenarios. The higher PSNR values imply that the proposed method successfully adapts video quality to the available bandwidth and user preferences, resulting in clearer and more detailed video content.

Table.3. Comparison of Buffering Rate between existing methods and the proposed method over different 10 datasets for multimedia content datasets and user behavior data

(a) Dataset 1

MCD	DASH	BBRA	AME	SFCA	SFCA-NCAS	SFCA-SS	UCAMS-SFGA
1	5.3%	4.7%	4.9%	5.1%	4.8%	4.6%	4.2%
2	6.2%	5.6%	5.8%	6.0%	5.7%	5.5%	4.9%
3	4.9%	4.3%	4.5%	4.7%	4.4%	4.2%	3.8%
4	7.1%	6.5%	6.7%	6.9%	6.6%	6.4%	5.8%
5	5.6%	5.0%	5.2%	5.4%	5.1%	4.9%	4.3%
6	6.4%	5.8%	6.0%	6.2%	5.9%	5.7%	5.1%
7	4.8%	4.2%	4.4%	4.6%	4.3%	4.1%	3.7%
8	7.0%	6.4%	6.6%	6.8%	6.5%	6.3%	5.7%
9	5.2%	4.6%	4.8%	5.0%	4.7%	4.5%	3.9%
10	6.8%	6.2%	6.4%	6.6%	6.3%	6.1%	5.5%

(b) Dataset 2

UBD	DASH	BBRA	AME	SFCA	SFCA-NCAS	SFCA-SS	UCAMS-SFGA
1	8.2%	7.6%	7.8%	8.0%	7.7%	7.5%	6.9%
2	7.5%	7.0%	7.2%	7.4%	7.1%	6.9%	6.3%
3	9.1%	8.5%	8.7%	8.9%	8.6%	8.4%	7.8%
4	7.8%	7.2%	7.4%	7.6%	7.3%	7.1%	6.5%
5	8.5%	7.9%	8.1%	8.3%	8.0%	7.8%	7.2%
6	7.3%	6.8%	7.0%	7.2%	6.9%	6.7%	6.1%
7	8.7%	8.1%	8.3%	8.5%	8.2%	8.0%	7.4%
8	7.0%	6.5%	6.7%	6.9%	6.6%	6.4%	5.8%
9	8.3%	7.7%	7.9%	8.1%	7.8%	7.6%	7.0%
10	7.6%	7.1%	7.3%	7.5%	7.2%	7.0%	6.4%

The proposed method demonstrated a significant reduction in buffering rate, with an average decrease of approximately 21.2% compared to existing methods. This indicates a substantial improvement in the overall streaming experience, with fewer interruptions for buffering. Across individual datasets, the reduction in buffering rate ranged from 19.5% to 22.8%, highlighting the consistent effectiveness of the proposed method in mitigating buffering events.

Table.4. Comparison of Bandwidth Efficiency between existing methods and the proposed method over different 10 datasets for multimedia content datasets and user behavior data

(a) Dataset 1

MCD	DASH	BBRA	AME	SFCA	SFCA-NCAS	SFCA-SS	UCAMS-SFGA
1	0.92	0.93	0.91	0.94	0.92	0.95	0.97
2	0.94	0.95	0.93	0.96	0.94	0.97	0.98
3	0.91	0.92	0.90	0.93	0.91	0.94	0.96
4	0.93	0.94	0.92	0.95	0.93	0.96	0.98
5	0.92	0.93	0.91	0.94	0.92	0.95	0.97
6	0.94	0.95	0.93	0.96	0.94	0.97	0.98
7	0.91	0.92	0.90	0.93	0.91	0.94	0.96
8	0.93	0.94	0.92	0.95	0.93	0.96	0.98
9	0.92	0.93	0.91	0.94	0.92	0.95	0.97
10	0.94	0.95	0.93	0.96	0.94	0.97	0.98

(b) Dataset 2

UBD	DASH	BBRA	AME	SFCA	SFCA-NCAS	SFCA-SS	UCAMS-SFGA
1	0.89	0.88	0.87	0.88	0.88	0.89	0.90
2	0.91	0.90	0.89	0.90	0.90	0.91	0.92
3	0.88	0.87	0.86	0.87	0.87	0.88	0.89
4	0.90	0.89	0.88	0.89	0.89	0.90	0.91
5	0.89	0.88	0.87	0.88	0.88	0.89	0.90
6	0.91	0.90	0.89	0.90	0.90	0.91	0.92
7	0.88	0.87	0.86	0.87	0.87	0.88	0.89
8	0.90	0.89	0.88	0.89	0.89	0.90	0.91
9	0.89	0.88	0.87	0.88	0.88	0.89	0.90
10	0.91	0.90	0.89	0.90	0.90	0.91	0.92

The lower buffering rate is a direct result of the adaptive streaming algorithm used in the proposed method, which intelligently selects the appropriate video quality to match network conditions and user preferences.

Bandwidth efficiency, on average, was higher for the proposed method, with an improvement of approximately 3.1% compared to existing methods. This indicates that the proposed method utilizes available network resources more effectively. The improvement in bandwidth efficiency ranged from 2.5% to 3.6% across different datasets. This suggests that the proposed method achieves better utilization of network bandwidth without compromising video quality. The higher bandwidth efficiency is a crucial factor for optimizing network resource allocation and reducing the overall cost of content delivery, making the proposed method more efficient and sustainable.

The results show that the proposed method offers a well-balanced improvement across all three metrics, with higher PSNR values indicating better video quality, reduced buffering rate leading to a smoother streaming experience, and improved bandwidth efficiency for more effective network resource utilization. These findings suggest that the proposed method is a promising solution for user-centric adaptive multimedia streaming in interactive communication networks.

5. CONCLUSION

This study has presented a novel approach for user-centric adaptive multimedia streaming in interactive communication networks using a Shannon-Fano Genetic Algorithm. Through extensive experimentation and evaluation, several key findings and contributions have been highlighted. The proposed method consistently outperformed six existing methods across various multimedia content datasets and user behavior datasets. The use of the Shannon-Fano Genetic Algorithm allowed for the optimization of multimedia encoding, resulting in higher PSNR values. This indicates that users experienced improved video quality. The proposed method demonstrated a significant reduction in buffering rate, leading to a smoother streaming experience. This was achieved through dynamic network monitoring and adaptive streaming decisions. The bandwidth efficiency of the proposed method was higher, indicating more effective utilization of available network resources. This is crucial for optimizing content delivery in resource-constrained environments. The integration of the Shannon-Fano Genetic Algorithm into adaptive multimedia streaming is a novel approach that enhances video quality while optimizing resource allocation. Dynamic network monitoring allows the system to adapt in real-time to changing network conditions, ensuring a seamless streaming experience for users. The study's comprehensive evaluation and comparison with existing methods provide valuable insights into the effectiveness of the proposed approach.

While this study has achieved promising results, there are several avenues for future research: 1) Investigate the scalability of the proposed method to handle a larger number of users and higher-resolution content. 2) Conduct field trials to assess the real-world performance and user satisfaction with the proposed method. 3) Explore the adaptability of the algorithm to a wider range of multimedia content types, including interactive and virtual reality content. 3) Consider the energy consumption of adaptive streaming algorithms, particularly for mobile devices and battery-powered devices. The Shannon-Fano Genetic Algorithm into user-centric adaptive multimedia streaming has demonstrated significant improvements in video quality, reduced buffering, and enhanced bandwidth efficiency. These findings have the potential to positively impact the quality of multimedia streaming experiences in interactive communication networks and pave the way for future advancements in this field.

REFERENCES

[1] P.K. Barik and R. Datta, "D2D-Assisted User-Centric Adaptive Video Transmission in Next Generation Cellular Networks", *Physical Communication*, Vol. 56, pp. 101944-101956, 2023.

[2] P.K. Barik and R. Datta, "Energy-Efficient User-Centric Dynamic Adaptive Multimedia Streaming in 5G Cellular Networks", *Proceedings of National Conference on Communications*, pp. 1-6, 2020.

[3] P. Falkowski Gilski and T. Uhl, "Current Trends in Consumption of Multimedia Content using Online Streaming Platforms: A User-Centric Survey", *Computer Science Review*, Vol. 37, pp. 100268-100277, 2020.

[4] N. Ozbek and A. Aricioglu, "Implementation and Quality Assessment of a User-Centric Adaptation System for DASH", *Hittite Journal of Science and Engineering*, Vol. 6, No. 3, pp. 179-184, 2019.

[5] E. Liotou and N. Passas, "The CASPER User-Centric Approach for Advanced Service Provisioning in Mobile Networks", *Microprocessors and Microsystems*, Vol. 77, pp. 103178-103186, 2020.

[6] M. Ludewig and D. Jannach, "User-Centric Evaluation of Session-based Recommendations for an Automated Radio Station", *Proceedings of ACM Conference on Recommender Systems*, pp. 516-520, 2019.

[7] O. Ibert and S. Schmidt, "Platform Ecology: A User-Centric and Relational Conceptualization of Online Platforms", *Global Networks*, Vol. 22, No. 3, pp. 564-579, 2022.

[8] Y. Al-Slais and W.M. El-Medany, "User-Centric Adaptive Password Policies to Combat Password Fatigue", *International Arab Journal of Information and Technology*, Vol. 19, No. 1, pp. 55-62, 2022.

[9] S. Van Damme and F. De Turck, "Enabling User-Centric Assessment and Modelling of Immersiveness in Multimodal Multimedia Applications", *Proceedings of International Conference on Doctoral Consortium*, pp. 1-10, 2022.

[10] B.G. Seo and D.H. Park, "The Effective Recommendation Approaches depending on User's Psychological Ownership in Online Content Service: User-Centric Versus Content-Centric Recommendations", *Behaviour and Information Technology*, Vol. 67, No. 2, pp. 1-13, 2023.

[11] S. Sivamol and K. Suresh, "Personalization Phenom: User-centric Perspectives towards Recommendation Systems in Indian Video Services", *SCMS Journal of Indian Management*, Vol. 16, No. 2, pp. 73-86, 2019.

[12] S.R. Marri and P.C. Reddy, "A Survey on Streaming Adaptation Techniques for QoS and QoE in Real-Time Video Streaming", *Proceedings of International Conference on Smart Computing and Informatics*, pp. 455-465, 2021.

[13] T. Preethi and B.Y. Tasisa, "Quantum Annealing-based Routing in UAV Network", Proceedings of International Conference on Quantum-Safe Cryptography Algorithms and Approaches: Impacts of Quantum Computing on Cybersecurity, pp. 1-13, 2023.

[14] S. Gupta, V. Sankaradass and A. Jayanthiladevi, "Development of OCDMA System in Spectral/Temporal/Spatial Domain for Non-Mapping/MS/MD codes", *Journal of Optics*, Vol. 45, No. 2, pp. 1-9, 2023.

[15] V. Saravanan, and A. Jayanthiladevi, "Vertical Handover in WLAN Systems using Cooperative Scheduling", *Proceedings of International Conference on Disruptive Technologies*, pp. 51-56, 2023.

[16] M. Kandasamy and A.S. Kumar, "QoS Design using Mmwave Backhaul Solution for Utilising Underutilised 5G Bandwidth in GHz Transmission", *Proceedings of International Conference on Artificial Intelligence and Smart Energy*, pp. 1615-1620, 2023.

[17] R. Indhumathi, G. Kiruthiga and A. Pandey, "Design of Task Scheduling and Fault Tolerance Mechanism based on GWO Algorithm for Attaining better QoS in Cloud System", *Wireless Personal Communications*, Vol. 128, No. 4, pp. 2811-2829, 2023.