Sasikumar et al. Journal of Cloud Computing (2023) 12:139
https://doi.org/10.1186/s13677-023-00519-2

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH **Open Access**

A Hierarchical Optimized Resource Utilization based Content Placement (HORCP) model for cloud Content Delivery Networks (CDNs)

M. Sasikumar[1*], P. Jesu Jayarin[2] and F. Sangeetha Francelin Vinnarasi[3]

Abstract

Content Delivery Networks (CDNs) have grown in popularity as a result of the ongoing development of the Internet and its applications. The workload on streaming media service systems can be significantly decreased with the help of the cooperative edge-cloud computing architecture. In the traditional works, a different types of content placement and routing algorithms are developed for improving the content delivery of cloud systems with reduced delay and cost. But, the majority of existing algorithms facing complexities in terms of increased resource usage, ineffective delivery, and high system designing complexity. Therefore, the proposed work aims to develop a new framework, named as, Hierarchical Optimized Resource Utilization based Content Placement (HORCP) model for cloud CDNs. Here, the Chaotic Krill Herd Optimization (CKHO) method is used to optimize the resource usage for content placement. Then, a Hierarchical Probability Routing (HPR) model is employed to enable a dependable end-to-end data transmission with an optimized routing path. The performance of the proposed HORCP model is validated and compared by using several performance metrics. The obtained results are also compared with current state-of-the-art methodologies in order to show the superiority of the proposed HORCP model. By using the HORCP mechanism, the overall memory usage of the network is reduced to 80%, CPU usage is reduced to 20%, response is minimized to 2 s, and total congestion cost with respect to the network load level is reduced to 100.

Keywords Cloud System, Content Delivery Networks (CDNs), Hierarchical Optimized Resource Utilization based Content Placement, Chaotic Krill Herd Optimization (CKHO), Hierarchical Probability Routing (HPR) model, Optimized Cost

*Correspondence:
M. Sasikumar
msasikumarcsephd@gmail.com
[1] Department of Information Technology, Anand Institute of Higher Technology, Kazhipattur, Chennai, Tamil Nadu 603103, India
[2] Department of Data Analytics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu 602105, India
[3] Department of Information Technology, St. Joseph's Institute of Technology, Chennai, Tamil Nadu 600119, India

Introduction

Over the past few decades, the urban population growth has been unprecedented throughout the world [1]. According to studies, the number of people living in cities worldwide has increased by roughly 60 million annually, and by 2050, it's predicted that 70% of people intend to do so. The quality of experience (QoE) [2, 3] of consumers as well as the data needs of cities may both be met by content delivery networks (CDNs). A network of millions of internet-connected devices is known as a content delivery network [4]. Here, a service provider's limited supply of servers are linked together all over the world. Due to the employment of numerous servers, the load distribution should be preserved, significantly assisting the user in receiving high-quality material faster. A CDN [5] is made up of an origin server and a set quantity of proxy servers. In a CDN, duplicate copies of the content are created on the origin server and kept on the proxies of those servers [6, 7]. Proxy servers are built in a number of different places throughout the world. If a client simply requests a specific piece of online content, the request information is first transmitted to the geographically local proxy server, which then checks the content's availability. Typically, they have emerged as a result of the growth in Internet users and their desire for low-latency delivery of content. The goal of CDN [8–10] is to put the physical responsive server as close to the user as possible.

This concept is implemented utilizing an arrangement of proxy servers and cache providers offered by the content provider, with the last level often situated at the edge of the user network. One of the most important user demands is for quick content delivery, and CDN is often referred to as an improved kind of web caching that satisfies this requirement. However, due to their high deployment costs, traditional CDN systems [11, 12] are difficult to scale. The emergence of cloud computing has made it possible to create CDNs in the cloud, or cloud CDNs, by leasing resources (such memory and bandwidth). It can make use of the cloud's versatility to disseminate content over the Internet swiftly and easily.While reducing the cost of data storage and their delivery, cloud CDNs can increase capacity, adaptability, and elasticity [13]. One of the fundamental technologies of cloud CDNs is content placement. By creating multicast delivery models, the content providers typically decrease the number of replicas in order to cut down the content placement costs. However, the majority of content delivery methodologies are focused on static networks and, are unable to migrate the dynamic features of cloud proxy servers in CDNs [14, 15].

Furthermore, one of the primary causes of the conventional content placement methods areincreased network congestion and high cost. Determining how to lower the cost of content placement is still a difficult topic in light of these dynamic cloud CDNs. Furthermore, the conventional content placement methods do not take into account the global dynamics of the congestion in cloud CDNs and only give delivery paths based on local decision-making [16–18]. Because of these dynamic properties, it remains a difficult task to figure out how to lower the cost of content placement. Therefore, the proposed work aims to use an effective content placement model for cloud CDNs [19]. Also, the proposed content placement methodology can effectively distribute material and balance the load on dynamic networks better than the classic routing techniques. The major objectives of the proposed work are as follows:

- In this paper, a novel Hierarchical Optimized Resource Utilization based Content Placement (HORCP) model is developed for CDNs.
- The optimal content placement in cloud is performed in this research work with reduced cost and latency.
- The Chaotic Krill Herd Optimization (CKHO) algorithm is implemented to obtain the best value for optimizing the resource usage load for content placement.
- Moreover, the Hierarchical Probability Routing (HPR) model is employed to enable the reliable end-to-end data transmission with optimized routing path.
- Various performance measurements are used to compare and validate the suggested HORCP model's performance. In order to demonstrate the superiority of the suggested content placement model, the acquired results were also contrasted with current state-of-the-art model techniques.

In the proposed work, the HORCP mechanism is developed with the CKHO technique for enabling a successful and reliable content delivery in cloud systems. The performance and effectiveness of the proposed HORCP model is validated and assessed by using a variety of parameters including congestion cost, response time, and hit ratio. Moreover, the findings demonstrate that the proposed HORCP provides the reduced congestion cost and time while enabling a successful content delivery in cloud networks.

The remaining sections of the paper are divided into the following categories: The thorough literature overview on the current content delivery techniques used in traditional cloud systems, along with their positive and negative aspects, is provided in Sect. 2. The suggested HORCP paradigm is clearly explained in Sect. 3 along with an overall flow diagram and stage-by-stage justifications. Additionally, various parameters are used to validate the proposed HORCP model's performance and comparative outcomes.

Related works

This section presents an overview of existing content placement and routing algorithms utilized in cloud CDNs. Also, it analyses the advantages and disadvantages of each model based on how each model performs routing activities.

Zolfaghari, et al. [20] presented a comprehensive survey to examine the recent trends and challenges in the CDN. Here, the clear lifecycle of CDN is provided that includes the stages of system design, implementation, deployment, operation, and evaluation. Moreover, the authors investigated the major operations of content distribution, requisition-based routing and performance management. In addition, the authors intended to improve the parameters of cost, scalability, security, and privacy while designing an effective CDNs [21]. Implemented a Lyapunov optimization algorithm for ensuring a reliable information sharing in cloud based CDNs. In this framework, the goal of each cloud provider is to increase its expected payout, which is calculated as the reciprocal of the weighted average of the anticipated data transfer cost and latency for its customers [22]. The computerized infrastructure of the CDN is managed by an individual content provider, which tracks the subscriber content requests and, assigns wireless channels/nodes to send the required content for minimizing the overall content transfer costs and delays. As a result, the distribution of material within a CDN can be described as an optimization challenge [23]. Formulated a new CDN architecture for enhancing the Quality of Experience (QoE) of smart city networks. The authors suggested a brand-new cellular network infrastructure for a smart city that includes content delivery and resource management. Due to the restricted processing capacity of mobile edge computing, the resource allocation gets an adequate computing equilibrium for an array of services and application. The content delivery portion is more concerned with network cache constraints and transmission delay. The greedy algorithm and the suggested framework can adapt to a variety of user requests and communication situations as demonstrated in the numerical outcomes [24]. Additionally, a comprehensive system for multiple content delivery has been developed for different service models with guaranteed user satisfaction.

Sadeghi, et al. [25] utilized a deep reinforcement learning algorithm for enhancing the content delivery of hierarchical networks. Caching is expected to be a key component of internet topologies, mobile networks, and next-generation CDNs. Caching can help the network infrastructure as well as end users by strategically storing the majority of popular contents at the storage-enabled network entities during off-peak demand instances. The authors of this paper indicated that the caching techniques are more useful for the next-generation networks, hence it is essential to take into account for solving the problems network caching, complex dynamics, and state space modeling. The major drawbacks of this work are increased complexity in system design, and low efficiency. Sinky, et al. [26] formulated a content centric framework for enhancing caching and placement operations in the smart city networks. To be more precise, the authors make use of the collaborative filtering theory for providing precise and effective content popularity estimations that support proactive in-network caching of web data. Here, the popularity driven content caching model is used to improve the QoE, where the following functions are computed content popularity, availability, population density, storage capability of node, and delay [27]. Implemented a Q-Learning based content placement algorithm for cloud systems. Here, an adaptive delivery tree is formulated to take the better routing decisions in the cloud CDNs. The learning packets in this approach can deduce the local and nonlocal congestion information. The regional traffic data shows how long it takes a packet to go from one cloud proxy server to another. The worldwide perspective of the latency issues from the present cloud proxy server to the terminating proxy server is provided by the non-local traffic data. Every cloud proxy server keeps the data in a Q-table and updates it using learning packets.

Alghamdi, et al. [28] Designed a new fog-cloud architecture model to improve the performance of CDNs. Here, a popularity-based caching strategy is used along with an Optimized Link State Routing (OLSR) protocol for CDNs. This study indicated that the factors such as security, node mobility, node failure rate, and scalability should be addressed for assuring a high quality of content delivery in the cloud systems. The primary advantages of this work are increased scalability and efficiency. Zhao, et al. [29] investigated about some of the recent developments in the field of CDNs. When designing a CDN, the operation costs are a crucial element that should be taken into account. Although some tactics, like expanding server cache capacity, can enhance CDN performance. But, they undoubtedly raise the service providers' operating costs, adding to their burden. Therefore, it is crucial for CDN providers to lower costs and delay. Moreover, the different types of caching mechanisms are reviews in this study for an effective server cache memory management [30]. Introduced a new Information Centric Networking (ICN) paradigm for improving the storage performance of cloud CDNs. The suggested content popularity mechanism increases the content availability at the proximity devices with reduced transfer time and packet loss ratio. Moreover, the clustering based efficient caching mechanism offers a suitable solution to the issue of the existing hash and on-path caching mechanisms.

The literature research [31–33] revealed that the current content placement and routing algorithms mostly focused on lowering CDN costs and enhancing content delivery. The bulk of techniques are having issues with excessive resource use, dependability issues, and accelerated delay. Therefore, the proposed research work aims to develop a new framework for cloud content placement in CDNs. Table 1 discusses about the list of existing methodologies used in the field of CDN with their findings.

Table 1 Literature review of the previous works

Ref	Methods	Findings
Hasan, et al.2019 [30]	New Information Centric Networking (ICN) paradigm	Increased content availability, reduced transfer time and packet loss ratio
Liu, et al.2019 [27]	Q-Learning based content placement algorithm	Better routing decisions, reduced local and nonlocal congestion information
Asheralieva, Aet al.2021	Lyapunov optimization algorithm	Reliable information sharing, minimized data transfer cost and latency
Chen, et al.2019 [23]	Greedy algorithm	Enhanced Quality of Experience (QoE), reduced transmission delay, and enabled multiple content delivery
Sadeghi, et al.2019 [25]	Deep reinforcement learning algorithm	Increased complexity in system design, and low efficiency
Sinky,H et al. 2019 [26]	Content centric framework using collaborative filtering theory	Optimized storage capability of node, and minimized delay in transmission
Alghamdi, et al.2019 [28]	Optimized Link State Routing (OLSR) for CDN	Increased scalability and efficiency

Proposed methodology

This section provides the clear explanation for the proposed cloud content delivery system with the workflow and appropriate illustrations. The original contribution of this paper is to perform an effective content placement and routing with the low time and cost complexities. For this purpose, a novel framework, named as, Hierarchical Optimized Resource Utilization based Content Placement (HORCP) model is developed in the proposed system. It also aims to construct a better and reliable cloud delivery environment with minimized congestion. The overall flow of the proposed HORCP model is shown in Fig. 1, which encompasses the following operations:

- System modelling
- Content placement
- Cloud resource usage analysis using Chaotic Krill Herd Optimization (CKHO) algorithm
- Hierarchical Probability based Routing (HPR)
- Performance evaluation

Figure 2 shows the Workflow model of the proposed content delivery system. After system modeling, an effective content placement in cloud is performed with the use of the HORCP model. During this process, the optimized content is placed in the environment according to the number of edge servers, steaming video, and

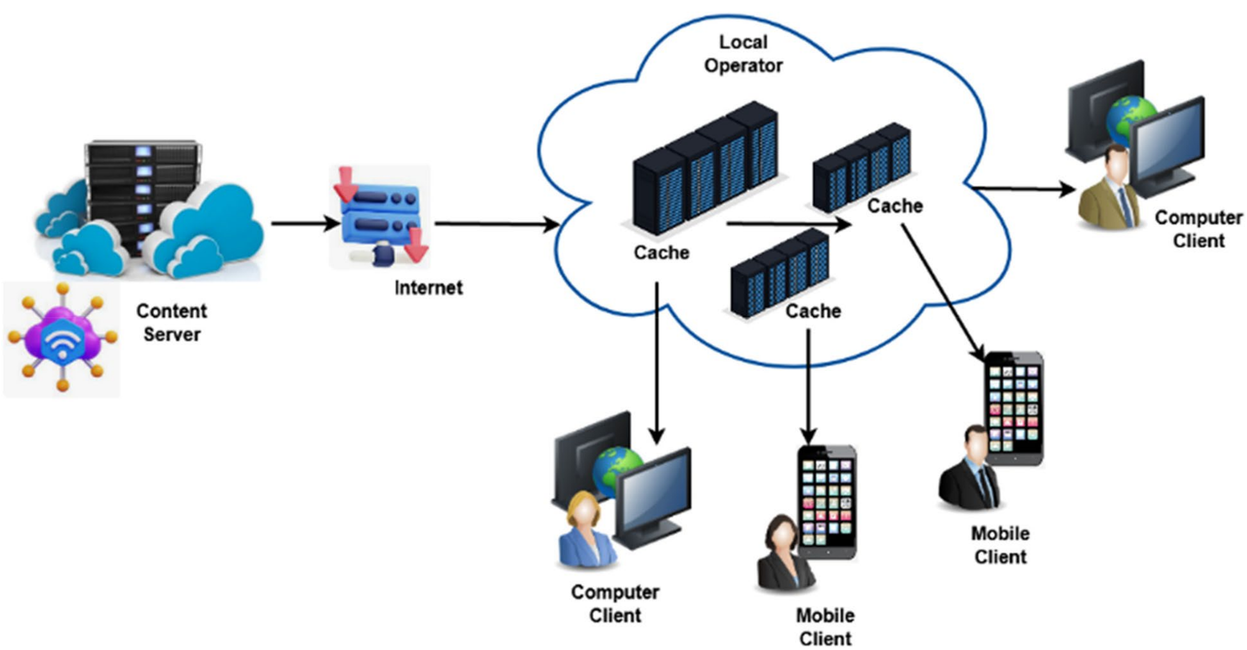

Fig. 1 Architecture model information flow in CDN

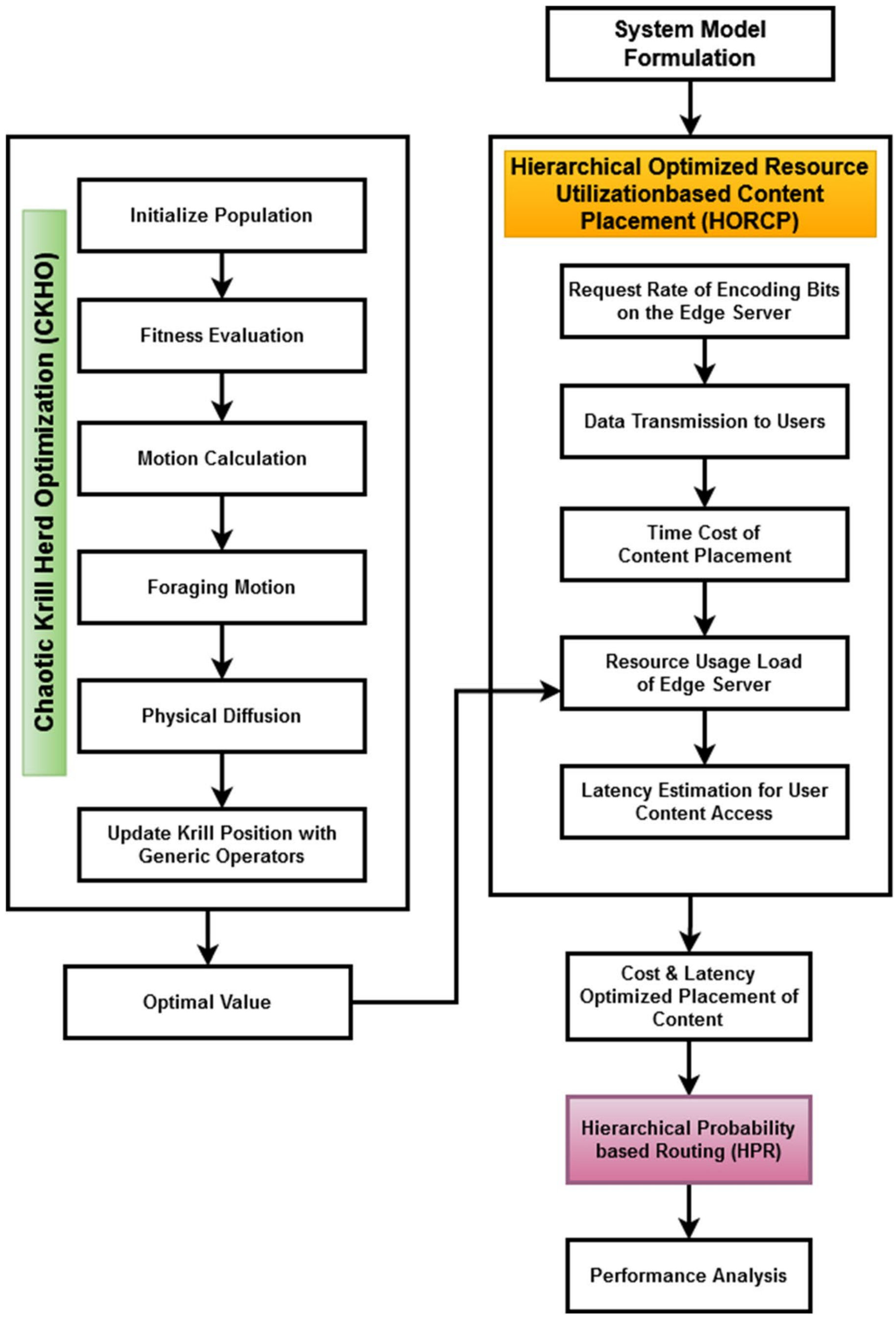

Fig. 2 Workflow model of the proposed content delivery system

set of users. Here, the resource usage load on the edge server is estimated based on the optimal value obtained from the Krill Herd Optimization (CKHO) algorithm. This technique helps to place the contents on the cloud systems with reduced latency and cost. Then, the Hierarchical Probability based Routing (HPR) mechanism is implemented to enable data transmission through the optimized route. The key benefits of this framework are increased efficiency, reduced cost, efficient resource utilization, and minimal delay. In the proposed work, the new content placement model is developed with the resource allocation and routing methodologies. After system modeling, an effective content placement model is implemented to successfully stream the multimedia services. During this process, the time cost estimation, resource usage allocation, and latency estimation are performed with the use of hierarchical optimization based routing model. In order to optimally the resource usage, a CKHO technique is implemented, which provides the best optimum solution to simplify the process of resource allocation. By using these processes, the optimized content placement is done in the proposed system with reduced cost and latency parameters.

Content placement

In the proposed HORCP framework, an effective content placement is performed with the reduced cost and latency factors. Typically, the content placement in cloud is one of the most prominent issues need to be addressed for enabling a reliable and effective data transmission. The system for streaming media services has faced enormous hurdles as a result of the increased demand from customers for video streaming services [34, 35]. The workload on streaming media service systems can be significantly reduced by the cooperative edge-cloud computing design, which combines edge computing and cloud computing. The cooperative edge-cloud computing architecture's streaming content caching is crucial for enhancing user service quality. This led the authors of this research to put forward a HORCP strategy based on user mobility and popular content. In this model, the mobile users submit enormous amounts of data to cloud data centers and edge servers as the source data generators. To decrease data transmission delay and traffic load, the edge servers use a portion of the storage space to cache specific information, particularly for video stream files with strict latency requirements. In this architecture, the user is able to access the contents that are cached in the edge server. The user will send the content request to the edge servers whenever they need to access the data. The content requests made by mobile users can be handled by the edge servers, which have constrained computation and storage capabilities, or they can be forwarded to the cloud data center. Users can access robust but delayed compute and storage capabilities from the cloud data center by sending content requests to it from edge servers. The cooperating cache domain, which is used to exchange the contents of each other's caches, is formed by pooling the cache spaces of all edge servers.

In this technique, the number of edge servers $\mathrm{E_s}$, steaming video $\mathrm{S_M}$, and available set of users U are considered as the input, and the predictive value for content placement is delivered as the output. At first, the input parameters are initialized $\mathrm{E_s} = \{1, 2, \ldots, \mathrm{S_r}\}$, $\mathrm{S_M} = \{1, 2, \ldots, \mathrm{M}\}$, and $\mathrm{u} = \{1, 2, \ldots, \mathrm{U}\}$, where each file is split into multiple encoding segments with the fixed length. After that, the request rate of encoding bits of the edge server are estimated, where the total number of times $\mathbb{R}_\mathrm{s}^\mathrm{u}$ that each user accesses the edge server's encoding segments $\mathrm{S_M}$ as shown in the following model:

$$\mathbb{R}_\mathrm{s}^\mathrm{u}(\mathrm{t}) = \sum_{\varphi=1}^{\mathrm{U}} \sum_{\mathrm{f}=1}^{\phi} \mathbb{R}_{\varphi,\mathrm{f}}^\mathrm{u}(\mathrm{t}) \tag{1}$$

where, φ denotes each user, f denotes each content, ϕ is the number of contents, $\mathbb{R}_\mathrm{s}^\mathrm{u}$ represents the encoding segments of the file at number of times, and $\mathrm{S_M}$ denotes the encoding segments. Consequently, the frequency for user u in the accessing file $\mathrm{S_r}$ from the edge server s is estimated by using the following equation:

$$F_j^u(s) = \frac{\mathbb{R}_{\varphi,f}^u(t)}{\mathbb{R}_s^u(t)} t = 1 to N_\rho \tag{2}$$

where, N_ρ represents the time duration and F_j^u is the frequency of user. Furthermore, the average popularity of file f at the edge server is estimated based on the following model:

$$F_j^u(s) = \frac{\sum_{u=1}^{U} F_j^u(s)}{U}, S_M = \{1, 2, \ldots, M\} \tag{3}$$

After computing the average population of file, the probability of user located on the edge server at the time period $(t+1)$ is also estimated as shown in below:

$$PR_j^u(t+1) = PR_c^u(t) * PR_{c\rightarrow s}^u(t+1) \tag{4}$$

where, c represents the count of probability for each user U. Finally, the predictive value for the content placement is obtained according to the number of users moving to the edge server at the time period $(t+1)$ as shown in below:

$$\aleph_{nu}^s(t+1) = \sum_{u=1}^{U} F_j^u(t+1) \tag{5}$$

$$F_j^u(t+1) = \begin{cases} 1 if\, max\left(PR_j^u(t+1)\right) \\ 0 else \end{cases} \tag{6}$$

Based on the $F_j^u(t+1)$ value, the content is placed in the cloud systems for an efficient and reliable routing. After computing the time cost, the resource usage local on the edge server is optimally computed by using the KHO algorithm, which helps to obtain the low latency and cost factors while placing contents on the cloud.

Input: Edge server E_s, Streaming video S_M, set of user$U$;
Output: Predicted value $\aleph_{nu}^{s}$;
Procedure:
Step 1: Let $E_s = \{1,2,\ldots,S_r\}$,denotes the set of edge servers, $S_M = \{1,2,\ldots,M\}$ denotes the set of streaming media files stored in the edge servers,$u = \{1,2,\ldots,U\}$denotes the set of users. Each file in $S_M$is divided into multiple encoding segments with a fixed length.
Step 2: Let,$\mathbb{R}_s^u$denotes the number of times that the encoding segments of the file, the total number of times that all users access the encoding segments in edge servers can be modelled S_M. Here, $\mathbb{R}_s^u(t)$ is estimated based on the number of contents and user as shown in equ (1);
Step 3: Hence, the frequency $F_j^u$of user u accessing file $S_r$in edge server$s$is estimated by using equ (2);
Step 4: Compute the average popularity of file f at edge server $s$by using equ (3);
Step 5: Estimate the probability that user PR_j^u is located on the edge server $S_M$at time period$(t+1)$by using equ (4);
Step 6: Obtain the predicted value of the number of users moving to the edge server $s$at time$(t+1)$ as shown in equ (5) and (6);

Algorithm 1. Hierarchical Optimized Resource Utilization based Content Placement (HORCP) model

Chaotic Krill-Herd Optimization (CKHO)

The resource utilization load on the edge is optimized during content placement using the CKHO algorithm's solution. Different optimization algorithms are used in conventional works to solve complex problems. The CKHO technique has the major benefits of increased convergence rate, reduced local optimum, simplicity, adapts for parallel computing and high searching speed. In this technique, each krill's distance from the density of the krill swarm and food determines the objective function for krill movement. Moreover, each krill's position is made up of three components: natural dispersion, foraging motion, and motion induced by other krills. The three motions mentioned above in CKHO can be simplified according to the following Lagrangian model. Here, the target effect, local effect, and repulsive effect are roughly calculated as the three factors that will determine the trajectory of the first motion. The flow of the CKHO algorithm is shown in Fig. 3. In this technique, the parameters such as generalization counter G_c, and krill population ρ_k are considered as the inputs, and the optimal value ρ_k is produced as the output. According to the initial position, the fitness function is estimated based on the following model:

$$\acute{\eta} = fitness(\rho_k) \tag{7}$$

$$fitness(.) = \{_u^t + \mathfrak{a}_u^t \tag{8}$$

where, $\{_u^t$—frequency of user u and $\mathfrak{a}_u^t$—Average popularity of content. Until reaching the maximum number of iterations, the population is sorted as shown in below:

$$S_\rho = sort(\rho_k) \tag{9}$$

Then, the motion updated is performed by using the following equation:

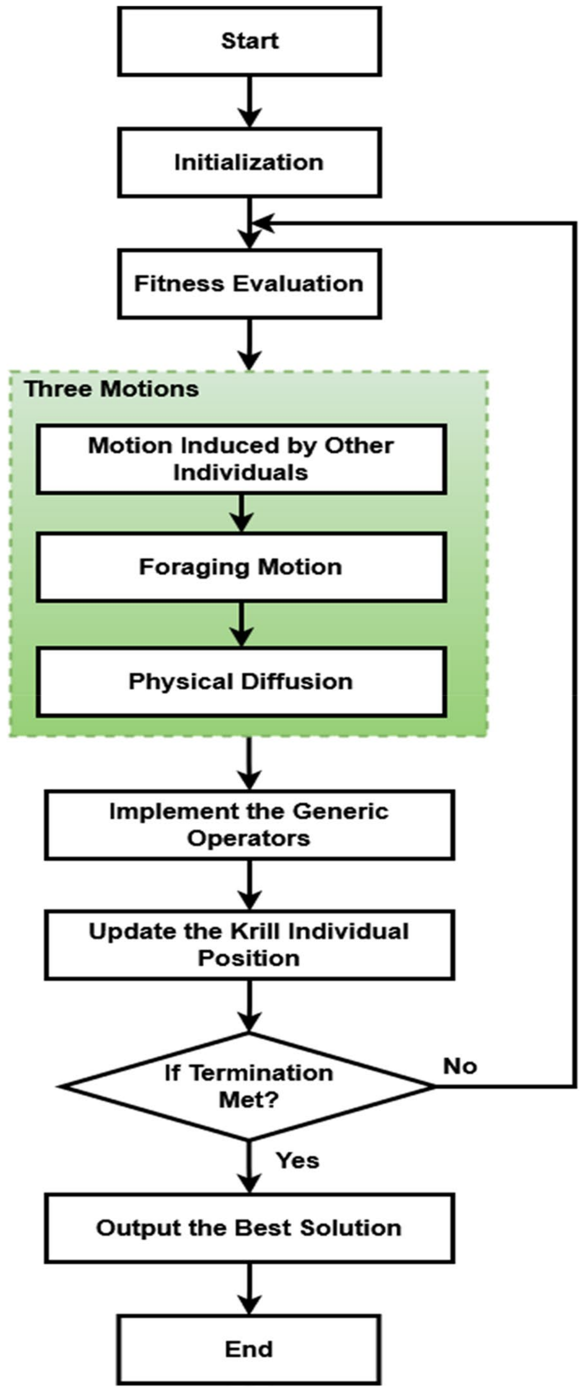

Fig. 3 Flow of CKHO algorithm

$$\mathcal{M}_j^{new} = \mathcal{M}^{max}\varepsilon_j + \mathfrak{w}_n\mathcal{M}_j^{old} \tag{10}$$

$$\varepsilon_j = \varepsilon_j^{total} + \varepsilon_j^{target} \tag{11}$$

Where, $\mathcal{M}^{max}$ indicates the maximum speed, $\mathfrak{w}_n$ represents the inertia weight in [0, 1], $\mathcal{M}_j^{old}$ is the previous motion, ε_j^{total} and ε_j^{target} are the local and target effects respectively. Moreover, the motion induced by other individuals according to the foraging behavior is shown in below:

$$F_j = S_f \delta_j + \omega_j F_j^{old} \tag{12}$$

$$\delta_j = \delta_j^{food} + \delta_j^{best} \tag{13}$$

Where, S_f is the foraging speed, ω_j is the inertia weight between 0 and 1, F_j^{old} is the previous foraging motion, δ_j^{food} is the food attraction and δ_j^{best} is the effect of the best fitness. Then, the physical diffusion is performed, where the model of motion is represented based on the maximum diffusion speed and random vector as shown in the following models:

$$Đ_j = Đ^{max}\gamma \tag{14}$$

where,$Đ^{max}$ is the diffusion speed, and γ is the random vector in [1,1] . Furthermore, the krill position is updated in the searching space, and the fitness is estimated according the new position of krill as represented in the following model:

$$\mathcal{M}_j^{best} = \mathcal{M}^{max}\varepsilon_j + \mathfrak{w}_n \mathcal{M}_j^{old} \tag{15}$$

Based on the $\mathcal{M}_j^{best}$, the best optimal value is obtained as the output of CKHO algorithm, which helps to optimize the resource load based on their usage.

Input: Initialize generation counter G_c, Initialize krill population ρ_k;
Output: Optimal value O_v;
Procedure:
Step 1: After parameter initialization, the fitness value is estimated for each krill according to its initial position as shown in equ (7) and (8);
Step 2: $while m < M_x$ //M_x- maximum generation
Sort the population$S_\rho$according to their fitness as shown in equ (9);
Step 3: $for j = 1\ to N_\rho$ // N_ρ – number of population

- The motion $\mathcal{M}_j^{new}$ calculation is performed using equ (10) and (11);
- Motion induced by other individuals Foraging motion is estimated using equ (12) and (13);

End for;
Step 4: Perform physical diffusion;
Step 5: The maximum diffusion speed and random vector are estimated according to the model of this motion;
Step 6: The model of this motion can be expressed according to two factors: a maximum diffusion speed$Đ_j$with a random vector using equ (14);
Step 7: Update the krill position in the searching space;
Step 8: Estimate the fitness value $\mathcal{M}_j^{best}$ for each krill according to its new position as shown in equ (15);
Step 9: Return the optimal value as the output $O_v = \mathcal{M}_j^{best}$;
Step 10: End while;

Algorithm 2. Chaotic Krill Herd Optimization (CKHO)

Hierarchical Probability Routing (HRP)

After placing the contents in cloud with optimized cost and latency, the routing is performed for enabling the reliable and successful data transmission. For this purpose, the Hierarchical Probability based Routing (HPR) mechanism is implemented, which provides the optimized route to enable the data transmission. A unique router architecture called as, Service-oriented Routing (SoR) is put forth to enable a router to offer content-based reliable networking services. It uses the deep packet inspection to examine the packet's header and payload, and also it records the essential data in a database. In the proposed framework, the CDN is constructed with the routing based system, where the mathematical modeling of routing request is computed for enabling the quality improved data delivery. For improving QoE in CDNs, some of the essential parameters need to be concentrated such as total response time, packet arrival rate, load on server, and latency. By using the HPR mechanism, the aforementioned parameters are improved in the proposed CDN framework. Here, the number of users N_u and content file $\complement_f$ are taken as the inputs for processing, and the optimized route $\mathbb{R}_r$ is produced as the output. At the beginning, the total link delay is estimated as shown in below:

$$\mathcal{D}_{SoR} = \sum_{i=1}^{d}\sum_{j=1}^{e} \vartheta_{ij}\rho_{ij}^{CS} \tag{16}$$

$$\vartheta_{ij} = min\left\{1 - \rho_{1j}, \frac{\omega_j^2}{\omega_j^1}\rho_{2j}\right\} \tag{17}$$

Where, $\mathcal{D}_{SoR}$ indicates the surrogate server, ϑ_{ij} represents the optimal route request probability, ρ_{1j} denotes the end to end delay between $user1^{st}$ and $j^{th}user$, ρ_{2j} is the end to end delay between $user2^{nd}$ and $j^{th}user$, ρ_{ij} defines the end to end delay between $useri^{th}$ and $j^{th}user$, ω_j^2 represents the bandwidth between $user2^{nd}$ and $j^{th}user$, and ω_j^1 is the bandwidth between $user1^{st}$ and $j^{th}user$. Consequently, the total waiting time is estimated according to the end-to-end packet transfer rate (T_{SoR}) as shown in below:

$$T_{SoR} = \sum_{a}\sum_{b}\frac{1}{\delta}\left(\mathcal{G}^{SoR} + \mathcal{G}^{R} + \mathcal{G}^{RR}\right) \tag{18}$$

where, $\mathcal{G}^{SoR}$ is the end-to-end packet transfer in surrogate server, $\mathcal{G}^{R}$ represents end-to-end packet transfer in route nodes, and $\mathcal{G}^{RR}$ denotes the end-to-end packet transfer in redirection probability. Moreover, the end-to-end packet transfer rate is estimated with respect to the surrogate server utilization rate $\mathcal{G}^{SoR}$ as shown in below:

$$\mathcal{G}^{SoR} = \frac{\tau_1}{\mu_1^{SoR} - \tau_1} + \frac{\tau_1\beta_{11} + \tau_{11}}{\mu_{11}^{SoR} - (\tau_1\beta_{11})} \tag{19}$$

where, τ_1 and τ_{11} are the surrogate server utilization, β_{11} represents the higher surrogate server loads, and μ_1^{SoR} indicates the service rate of a router. In addition, the end-to-end packet transfer rate in route nodes is estimated by using the following model:

$$\mathcal{G}^{R} = \frac{\tau_1\beta_{11} + \tau_{11}}{\mu_{11}^{SoR} - (\tau_1\beta_{11} + \tau_{11})} + \frac{\tau_1\beta_{12} + \tau_{12}}{\mu_{12}^{SoR} - (\tau_1\beta_{12} + \tau_{12})} \tag{20}$$

where, τ_1,τ_{11} and τ_{12} are the surrogate server utilization. Then, the end-to-end packet transfer rate according to the redirection probability is computed by using the following model:

$$\mathcal{G}^{RR} = \frac{\tau_1\partial + \tau_{DNS}}{\mu_{DNS} - (\tau_1\partial + \tau_{DNS})} \tag{21}$$

where, τ_{DNS} represents the domain name server utilization, and μ_{DNS} is the service rate of a router for the domain name server. At last, the reduced total link delay is estimated for the server centers by using the following model:

$$\mathcal{D}_{SoR} = \tau_{11}\rho_{11}^{CS} + \tau_{12}\rho_{12}^{CS} \tag{22}$$

where, ρ_{11}^{CS} indicates the end to end delay between user1st and server center, and ρ_{12}^{CS} represents the end to end delay between user2nd and server center. The final optimized route is computed as follows:

$$\mathbb{R}_r = \mathrm{route}(\mathcal{D}_{SoR}) \tag{23}$$

This optimal route is used for enabling the routing operation with reduced delay and ensured data quality.

Input: Number of userN_u, Content file C_f
Output: Optimized route $\mathbb{R}_r$
Procedure:
Step 1: The total link delay $\mathcal{D}_{SoR}$by using equ (16);
// $\mathcal{D}_{SoR}$ –surrogate server;
Step 2:Estimate the total waiting time for end-to-end packet transmission (T_{SoR}) by using equ (18);
Step 3: Compute the end-to-end packet transfer rate $\mathcal{G}^{SoR}$ by using equ (19);
Step 4: Compute the end-to-end packet transfer $\mathcal{G}^{R}$ in route nodes according to the surrogate server utilization using equ (20);
Step 5: Estimate the end-to-end packet transfer based on the redirection probability$\mathcal{G}^{RR}$ by using equ (21);
Step 6: Obtain the final reduction of total link delay $\mathcal{D}_{SoR}$ by using equ (22);
Step 7: Obtain the optimal route $\mathbb{R}_r$ for data transmission with reduced delay by using equ (23);

Algorithm 3. Hierarchical Probability based Routing (HPR)

Results and discussion

This section validates the simulation and comparison results of the existing and proposed content placement methodologies used in the cloud CDNs. For this assessment, the different types of parameters such as total congestion cost, memory usage, response time, and etc. are considered in this study. The Internet or a transmission network are examples of the real-world systems that are conceptually represented as communication networks. It is made up of links connecting a set of identical nodes. The communication network architecture utilized by Boyan and Littman, which includes an irregular 66 network and a 116 node LATA communication network, is employed in this research to test our approach. An intangible representation of a real-world system, like the Internet or a transmission network, is a communication network. It consists of a collection of isomorphic nodes connected by links. We put the experiment to the test in various network topologies. As shown in Figs. 4 and 6, the traffic congestion cost is computed for the conventional and proposed content placement methods with respect to different network load level. Table 2 compares the total congestion cost of the conventional and proposed content delivery models.

Here, both an irregular network architecture and LATA communication networking structures have been considered for analysis. The network comprises a pair of segments that are closely connected to one another as well as a bridge link. Network congestion is more likely to occur on the middle bridge link. The average number of injected packets per time unit serves as the load in this study and is a parameter value of the Poisson arrival process. Based on the analysis, it is observed that the total congestion cost for the proposed HORCP model is effectively reduced under varying network load level. Figure 5 and Table 3 shows the total congestion cost for LATA communication network and proposed content placement algorithms under varying time step values and iterations respectively.

Figures 6 and 7 shows the total congestion cost of the existing and proposed content placement algorithms under varying time step values and iterations respectively. We contrast the differences in the congestion cost over time between the suggested and present approaches using the LATA network design. We carry out tests with heavy as well as light loads. Low load indicates that there is little network congestion and fewer data packets are injected per unit of time. When there is a high load, there is severe network node congestion and more data packets are injected per unit of time. As seen in Fig. 6 and Table 4, when the time step reaches 2000, the HORCP congestion cost converges to a particular value with low load. The outcomes show that the HORCP approach can reduce the cost of congestion for content delivery along the route request model.

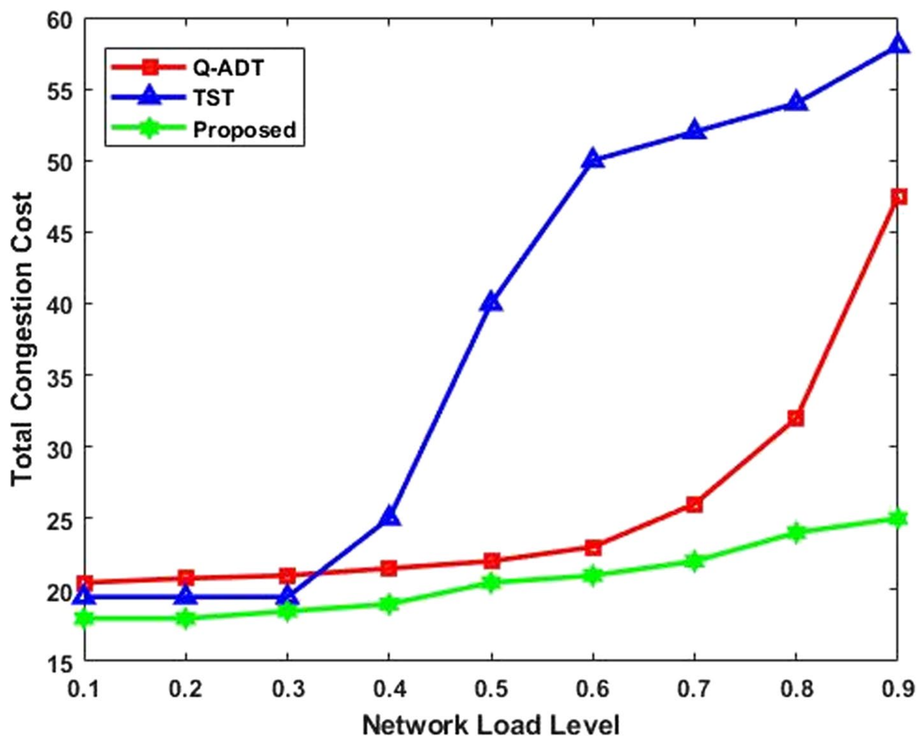

Fig. 4 Total congestion cost for irregular network architecture

Figure 8 and Table 5 validates the total congestion cost of the proposed HORCP model with respect to changing optimization based iterations such as 30, 50 and 90. In this analysis, the congestion cost is estimated according to the time step value. The obtained performance outcomes show that the proposed HORCP model effectively reduces the congestion cost with the use of CKHO and HPR mechanisms. Since, the content placement and routing processes are carried out in the proposed framework based on the optimized resource usage and reduced latency. Therefore, the proposed HORCP model outperforms the other existing content placement algorithms.

Table 2 Comparison based on total congestion cost

Network load level	Q-ADT	TST	Proposed
0.1	21	20	18
0.2	21.5	20	18
0.3	22	20	19
0.4	23	25	20
0.5	23.5	40	21
0.6	24	50	21.5
0.7	26	51	22
0.8	31	53	23
0.9	47	58	25

Figure 9 validates Cumulative Distributive Function (CDF) of existing and proposed content placement algorithms used for the CDNs with respect to changing response time. Typically, the response time is the most essential parameter used to determine that how effectively the proposed HORCP model could provide response to the given requests. It is clear that the algorithm's initial reaction time is practically identical to that of the greedy method. The performance does start to improve, though, as the response time goes down as the number of queries increases. The partitioning of the server's cache and consequent improvement in response quality can be employed to clarify this drop in response time. Similarly, the cache hit ratio and byte hit ratio are estimated for the conventional [8] and proposed models as shown in Figs. 10 and 11 respectively. In this instance, the cache will take 20% of the entire storage capacity. As can be observed, the hit rate is similar in the beginning when the number of requests is low, but as traffic and the number of requests rise, the HORCP method performs better than the standard models.

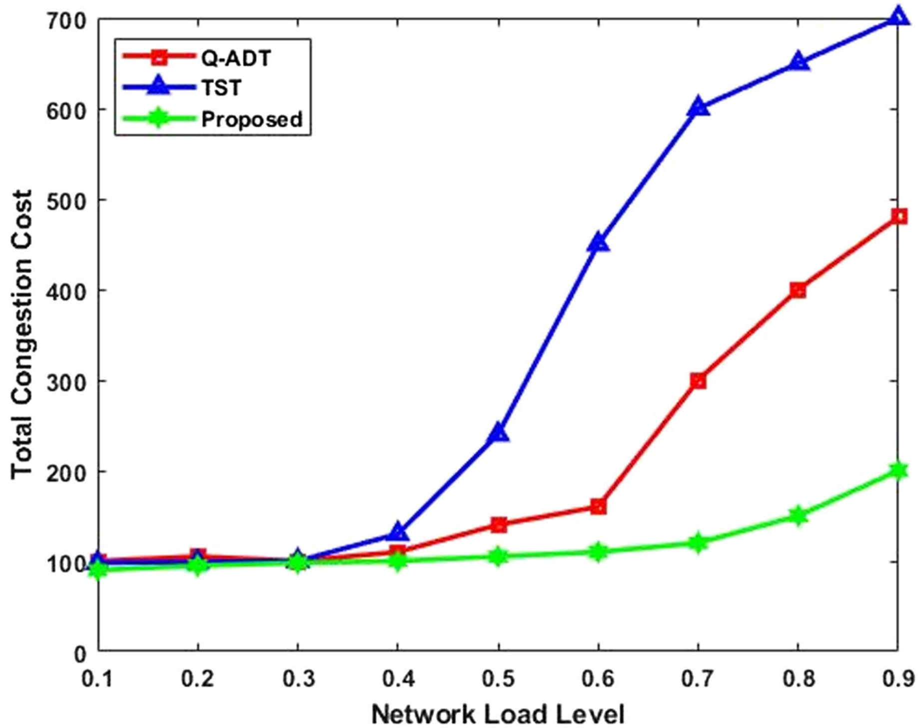

Fig. 5 Total congestion cost for LATA communication network

As can be observed, the hit rate has increased as a result of the great popularity of the items. In the proposed approach, the hit rate for the cache increases as more queries are made. This implies that the HORCP model has improved CDN performance. The segregation of the cache space allows for the storage of both dynamic and static data in the cache region, which is the major reason for the performance improvement. The hit rate will initially be low since the cache space has been used up, but as the number of requests rises, the space is filled by previously processed requests, increasing the hit rate as well as efficiency. Moreover, Figs. 12 and 13 validates the request and response time of the proposed HORCP model under varying unit time respectively. For this analysis, the functions executed with the available local data, remotely available data, and remotely executed data have been considered. Overall, the response time is effectively improved in the proposed model with the use of HPR mechanism, since which provides the optimized routing path for content delivery. In addition, the CPU and memory usage rates are validated for the aforementioned functions as shown in Figs. 14 and 15. In order to assure the reliable data transmission in CDNs, the cloud resource consumption such as CPU and memory should be effectively reduced. Because, an increased usage may degrade overall performance of the content delivery system. Therefore, the CKHO algorithm is implemented in the proposed to optimize the resource usage load.

Table 3 Comparison based on total congestion cost for LATA communication network

Network load level	Q-ADT	TST	Proposed
0.1	100	100	100
0.2	100	100	100
0.3	100	100	100
0.4	110	120	100
0.5	130	220	105
0.6	150	430	110
0.7	300	600	115
0.8	400	650	120
0.9	480	700	200

Figures 16 and 17 validates the traffic congestion cost of proposed HORCP model under varying network load level and time step respectively. In this analysis, the congestion cost before applying optimization and after applying optimization technique is estimated. In order to

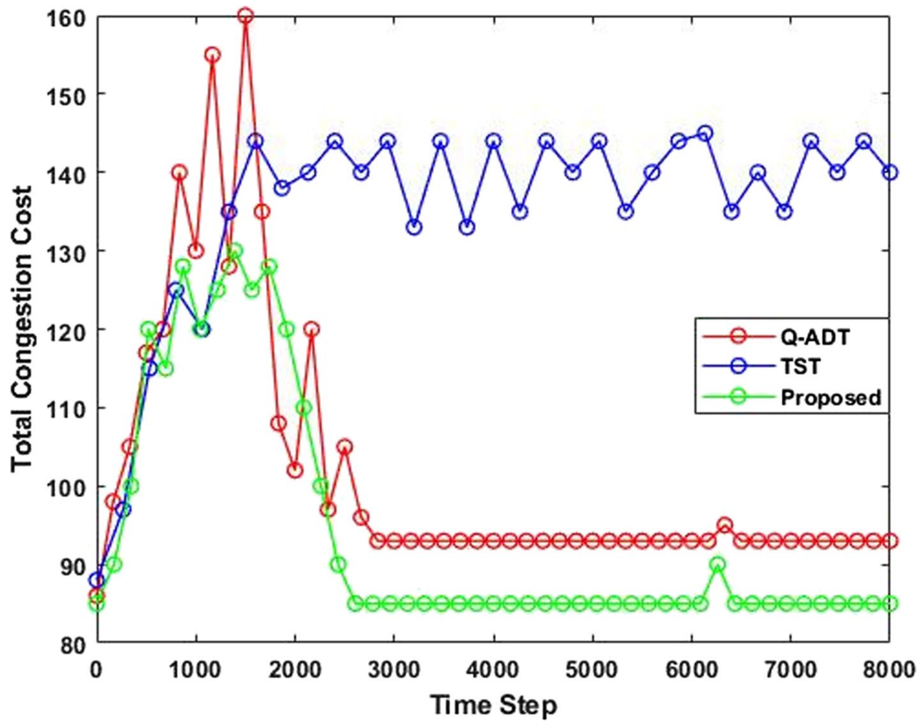

Fig. 6 Total congestion cost Vs time step

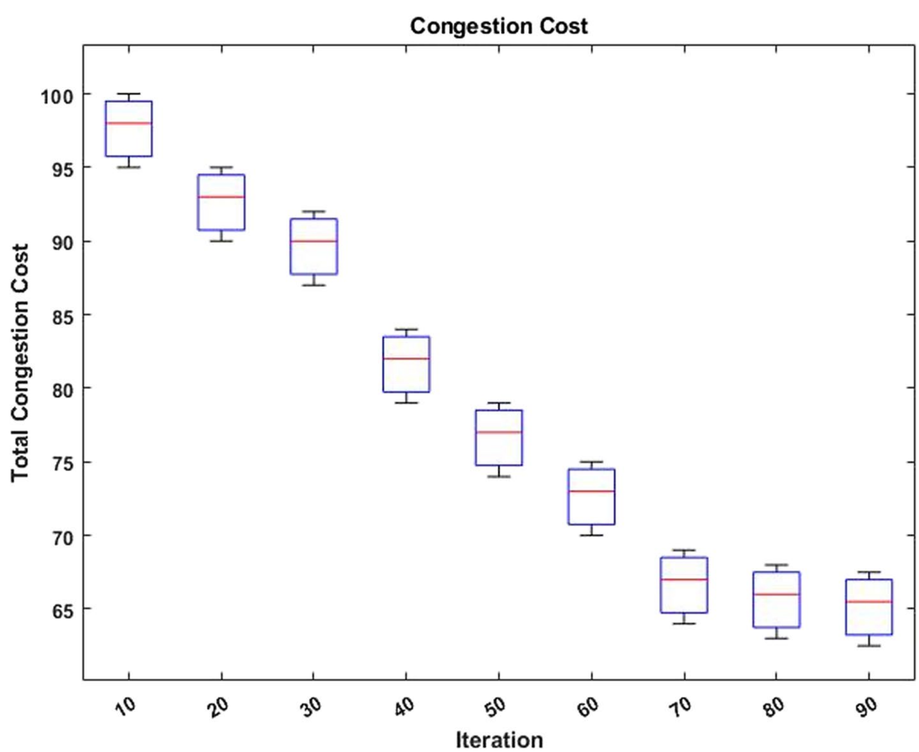

Fig. 7 Total congestion cost Vs iterations

Table 4 Congestion cost analysis for the proposed model

Iterations	Total congestion cost
10	98
20	94.5
30	90
40	82
50	78
60	73
70	68
80	67
90	67

demonstrate the efficiency of CKHO algorithm, the congestion cost with and without optimization processes is evaluated as shown in Fig. 18. The obtained results indicate that the proposed HORCP model could effectively minimize the congestion cost, when it incorporates with the CKHO algorithm.

Discussion

In order to ensure the reliable content delivery and enable data transmission in the cloud systems, the following factors must be satisfied:

- Self-service on demand
- Access to a wide network
- Resource aggregation
- Fast elasticity
- Measurable service

Content delivery networks have had a big influence on how content is provided to end users over the Internet. In the past, content providers have used independent CDNs to distribute their material to end users. It is difficult for content providers who either completely rely on third-party providers to understand and monitor the performance of their service due to the constantly changing landscape of content types, such as the transition from standard definition video to high definition to full high definition. Additionally, the geographic accessibility of the third-party infrastructure affects the CDN's speed. It offers a versatile approach that enables content producers to dynamically match and arrange material on one or more cloud storage servers based on preferences for coverage, cost, and quality of service. The pay-as-you-go model's advantages and economies of scale are the main implications. By utilizing clouds, content providers may manage circumstances like flash crowds more quickly without having to make infrastructure development investments. In addition to the CCDN-specific difficulties listed above, there are a number of critical

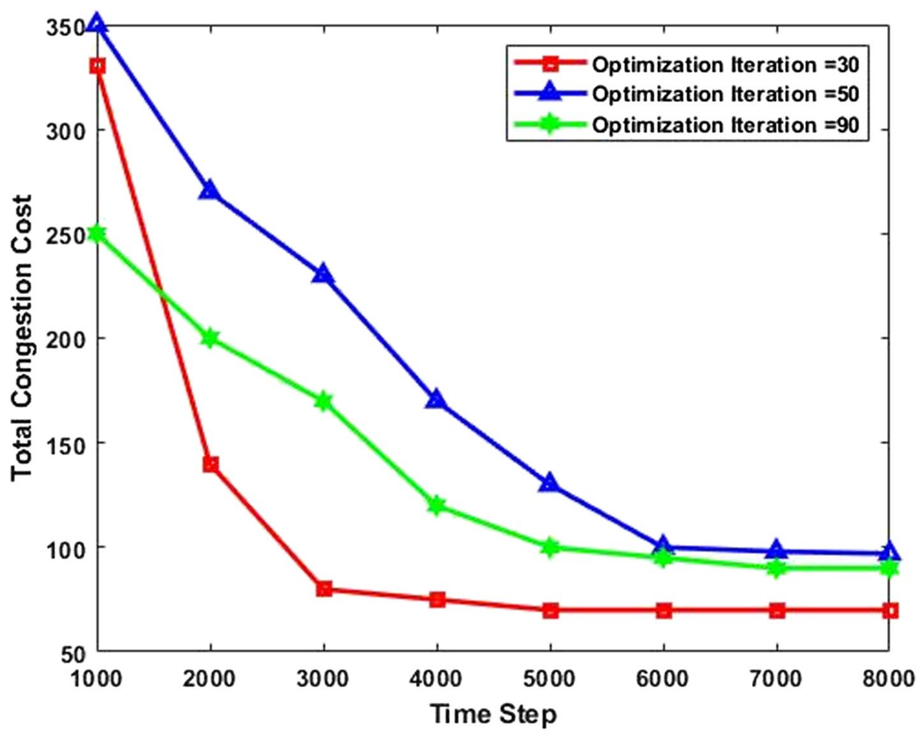

Fig. 8 Total congestion cost of HORCP

Table 5 Comparison based on total congestion cost of HORCP

Time step	Optimization iteration = 30	Optimization iteration = 50	Optimization iteration = 90
1000	330	350	250
2000	140	270	200
3000	80	240	170
4000	75	170	130
5000	70	140	110
6000	70	110	105
7000	70	110	100
8000	70	110	100

CDN-specific aspects that influence service performance within the cloud architecture. In the existing studies, a variety of classification techniques are implemented for ensuring a successful content delivery in cloud systems. Yet, the majority of techniques having the major problems of increased time for transmission, high cost consumption, lower efficiency, and increased congestion rate. Therefore, the proposed research study aims to develop a new framework for assuring an effective content delivery in cloud networks.

Conclusion

This paper presents a new framework, termed as, HORCP for an effective content placement and enabling optimized routing in cloud CDNs. It also aims to construct a better and reliable cloud delivery environment with minimized congestion. The content placement in the cloud tends to be one of the most important challenges that must be resolved in order to enable a dependable and efficient data transmission. Due to the rising demand from users for video streaming services, the system for streaming media services has encountered significant obstacles. The cooperative edge-cloud computing concept, which integrates edge computing and cloud computing, can greatly reduce the stress on streaming media service systems. Streaming content caching in the cooperative edge-cloud computing architecture is essential for improving user service quality. In this framework, the optimized content is placed in the environment according to the number of edge servers, steaming video, and set of users. Here, the CKHO algorithm's optimal result is used to assess the resource utilization load on the edge server. This method makes it easier and cheaper to upload content to cloud storage services. When the route has been optimized, the HPR mechanism is put into place to allow data transmission. This framework's

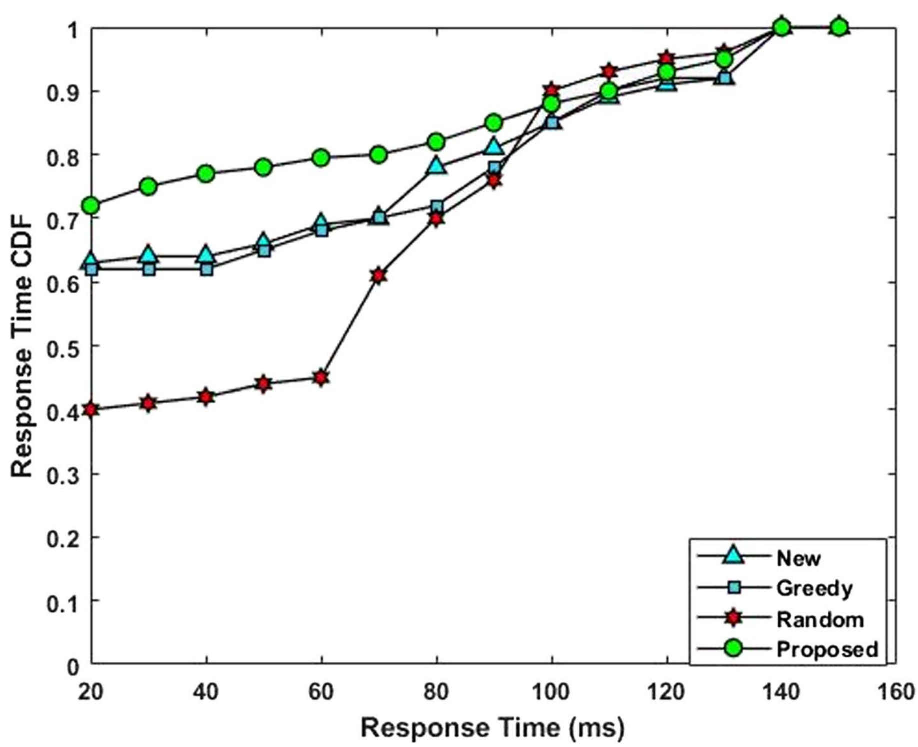

Fig. 9 Response time

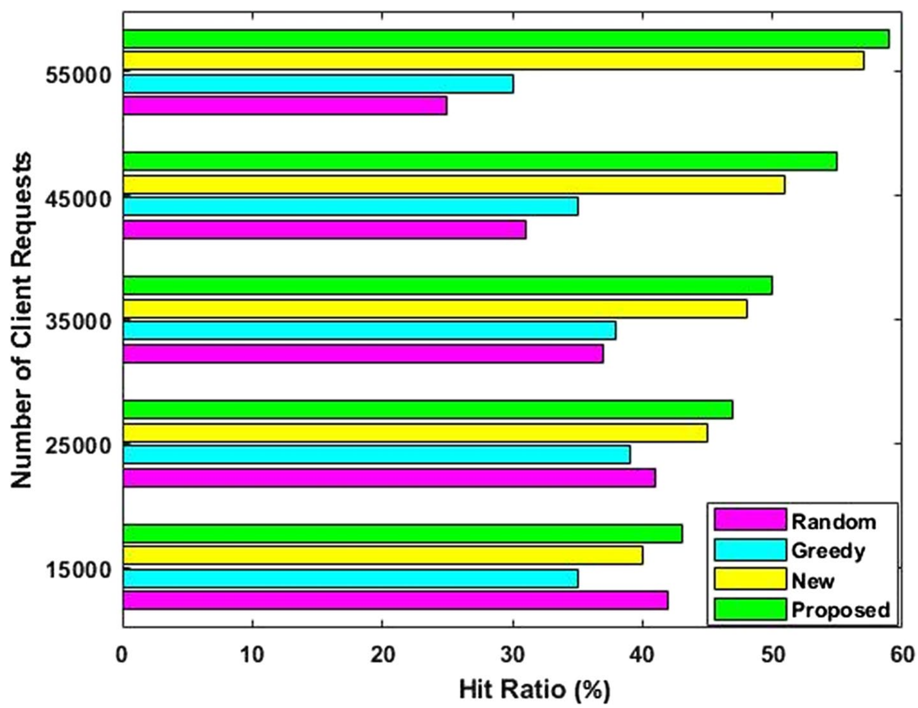

Fig. 10 Hit ratio

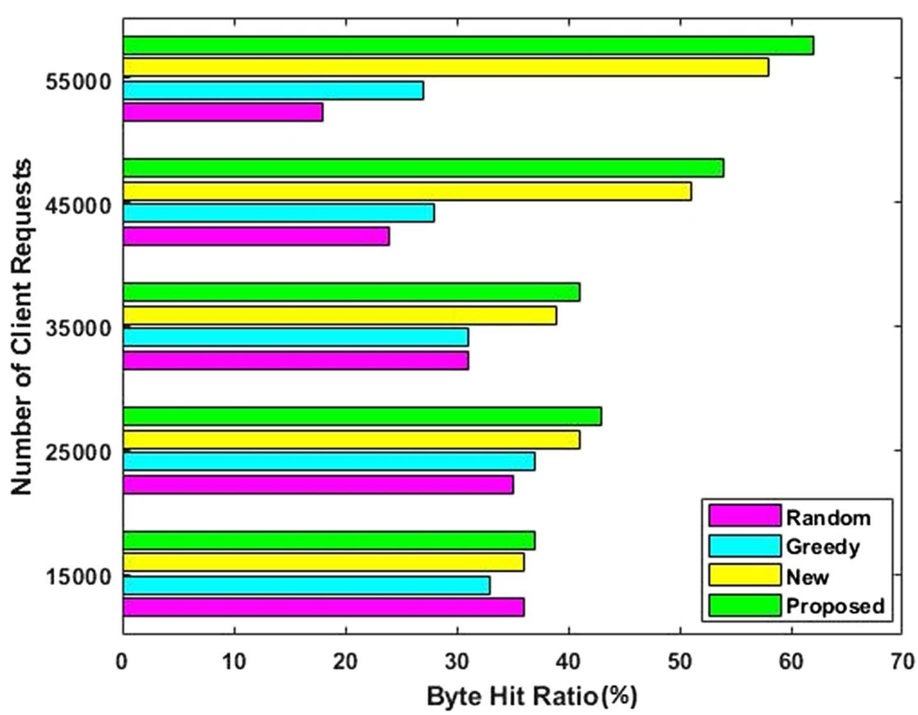

Fig. 11 Byte hit ratio

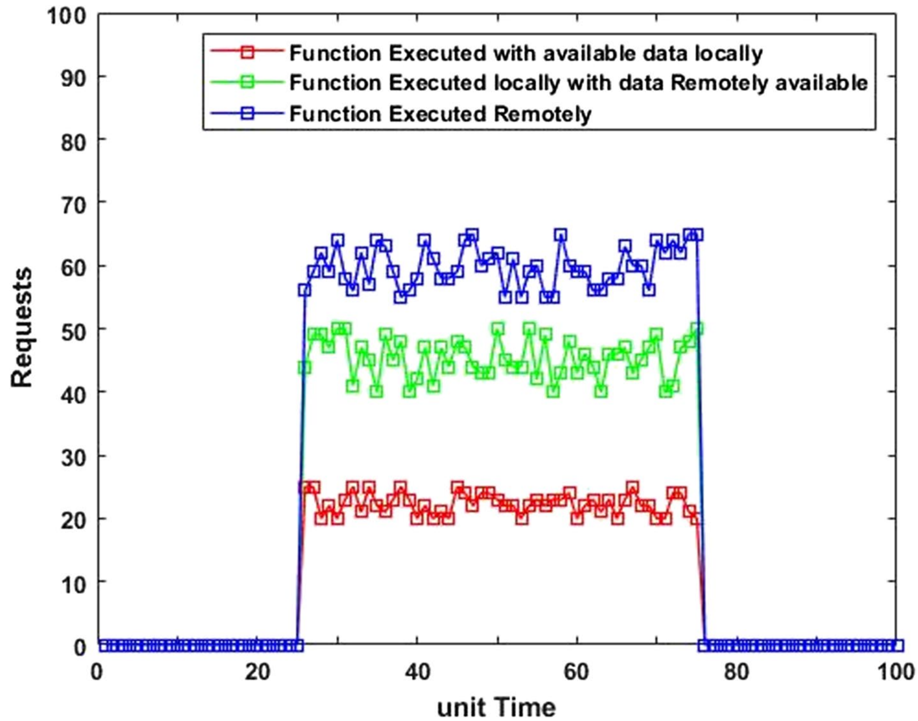

Fig. 12 Function execution analysis according to the route requests

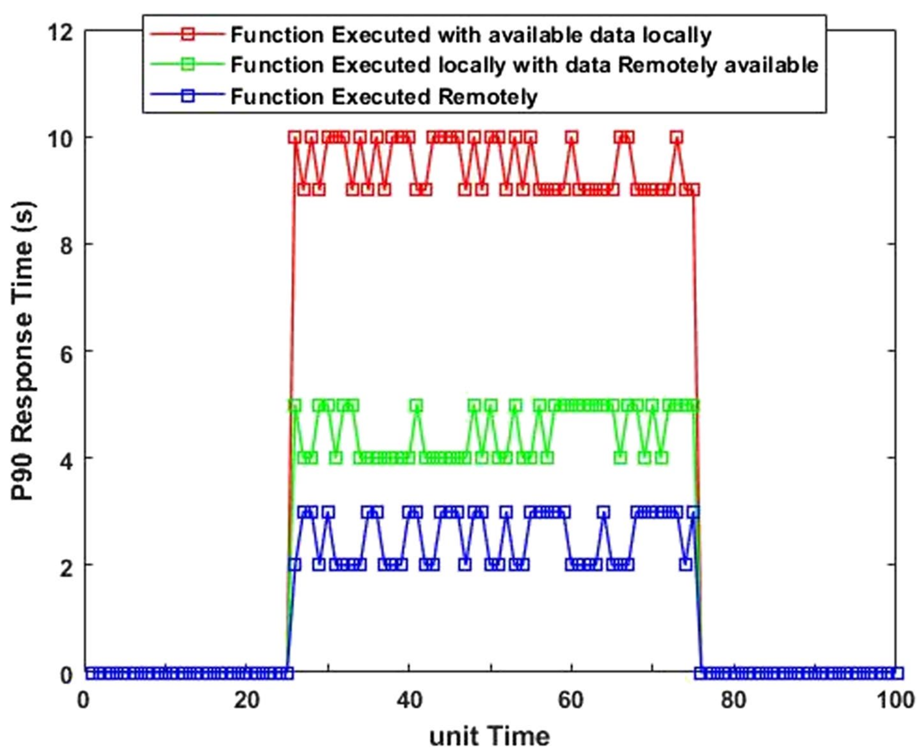

Fig. 13 Function execution analysis according to the response time

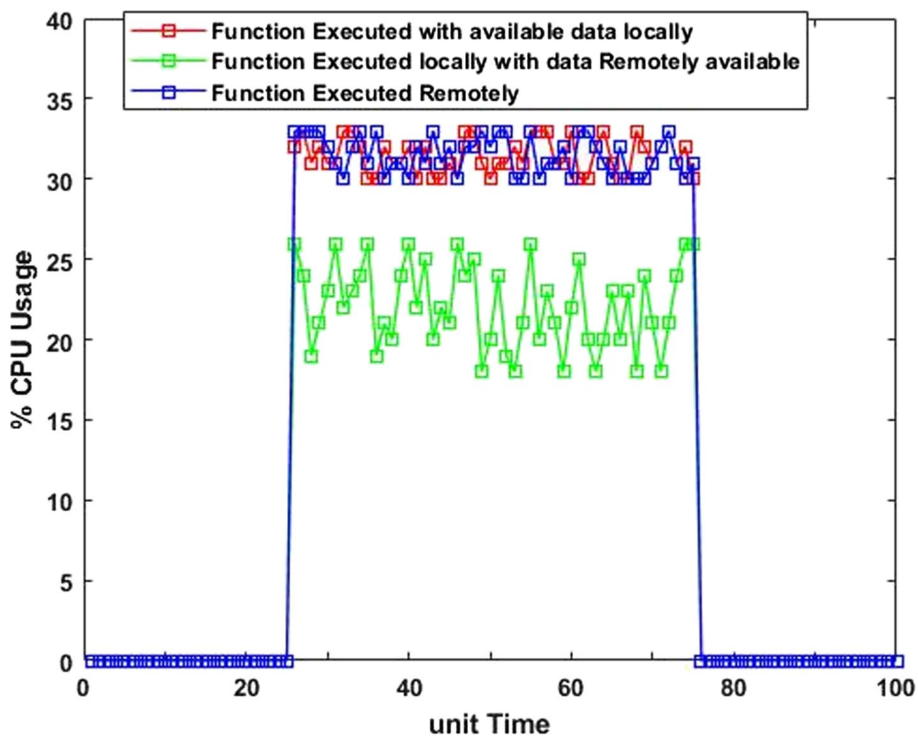

Fig. 14 CPU usage

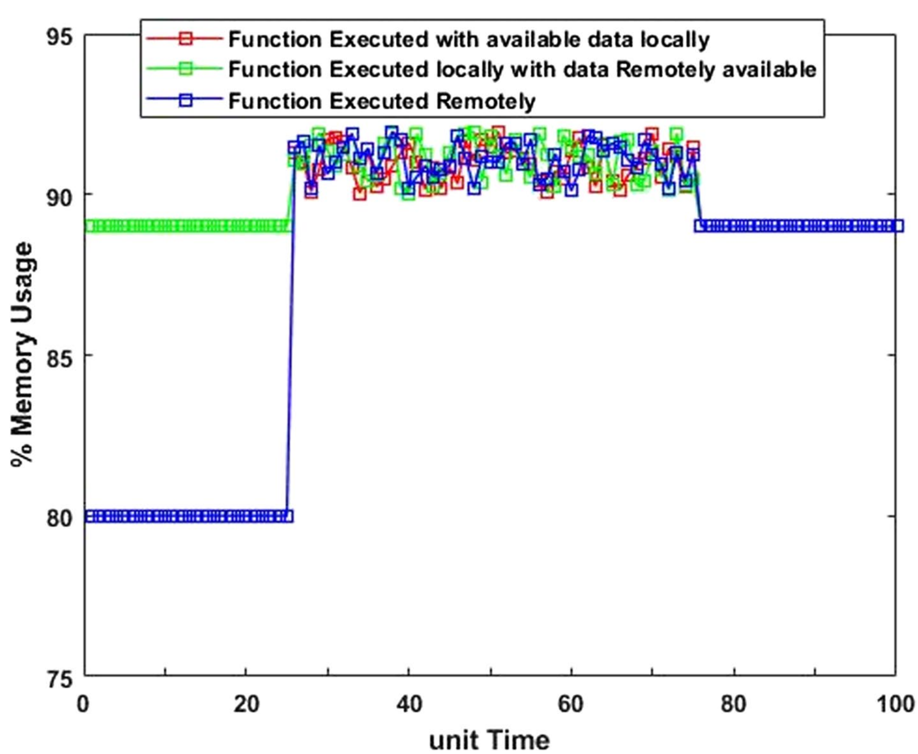

Fig. 15 Memory usage

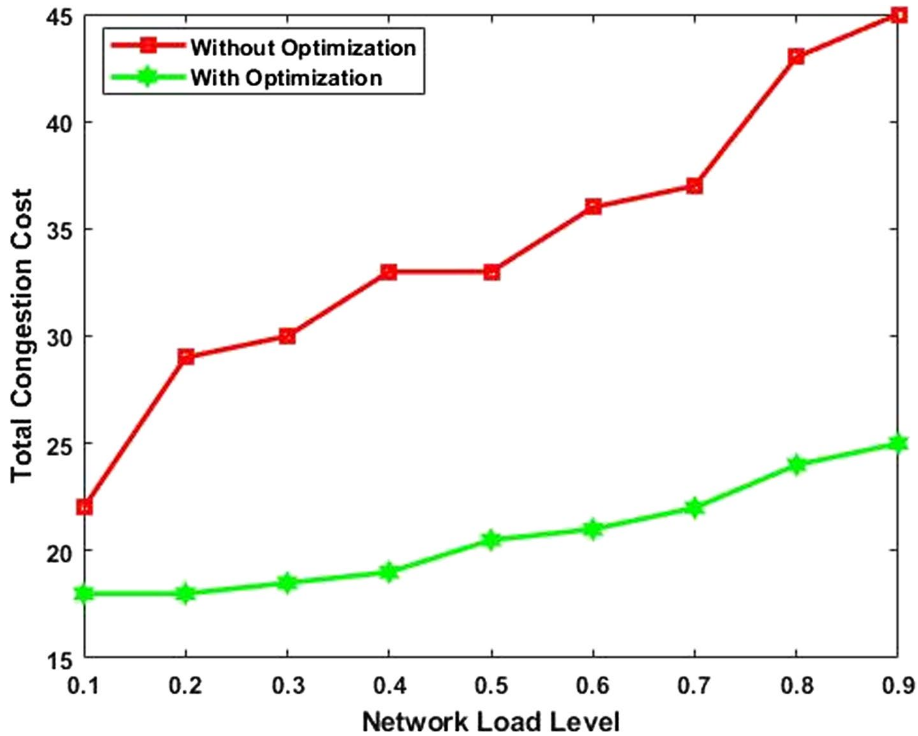

Fig. 16 Congestion cost of the HORCP model with and without optimization process under varying network load level

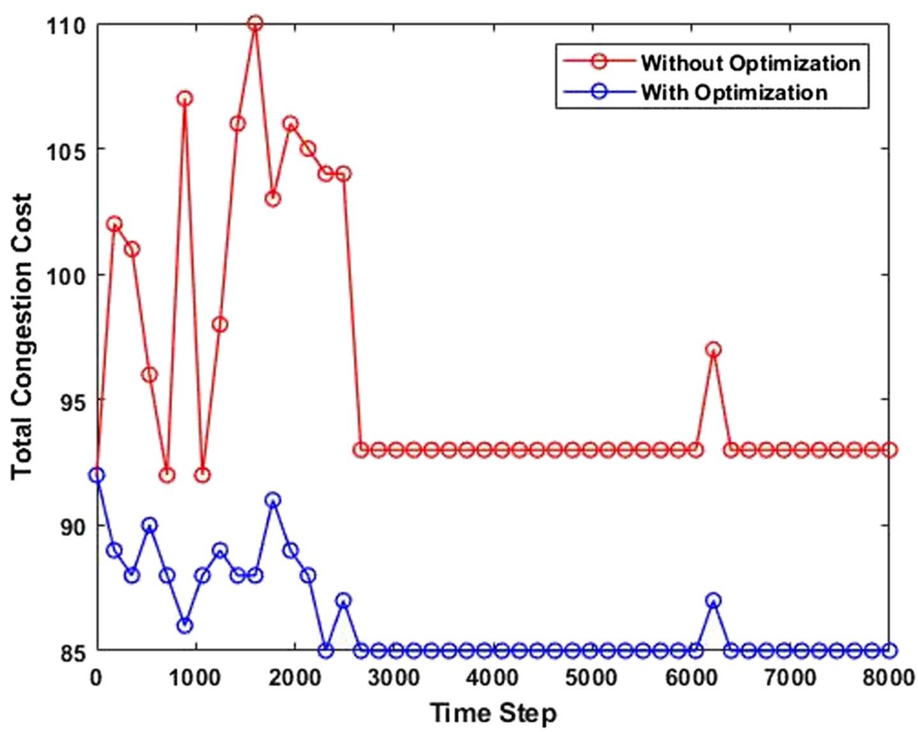

Fig. 17 Congestion cost of HORCP model with and without optimization under varying time step

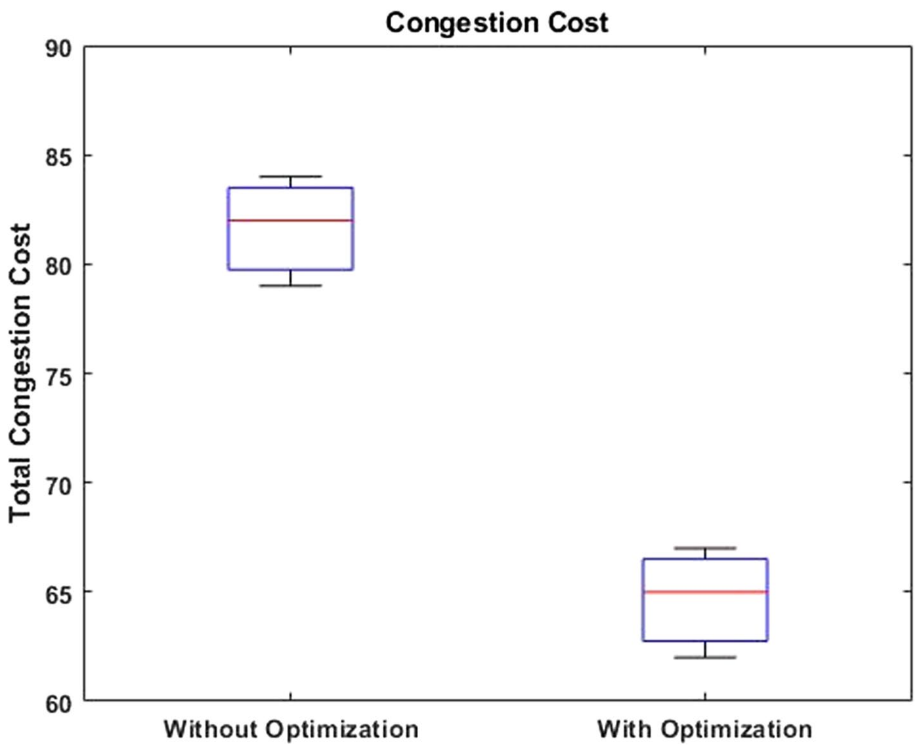

Fig. 18 Overall congestion cost analysis of HORCP model with and without optimization techniques

main advantages include greater effectiveness, lower costs, effective resource use, and minimal delay. During evaluation, the performance and results of the proposed method are validated and compared using different parameters such as congestion cost, response time, CPU usage, memory usage, hit ratio, and byte hit ratio. The obtained results reveal that the proposed HORCP model outperforms the existing content placement algorithms with improved performance outcomes.

This work can be enhanced in future by implementing a new machine learning and deep learning model for assuring the security of CDNs.

Authors' contributions

The author M.Sasikumar Contributed and put effort on paper to Organize the Paper. Developed the theoretical formalism, performed the analytic calculations and performed the numerical simulations. J.Jesu Jayarin designed the model and the computational framework and analyzed the data Also technically contributed and made English Corrections and grammar checking. Also involved and helped to derive the mathematical equation. The author F.Sangeetha Francelin Vinnarasi involved in the Background study of the Paper and helped the mathematical derivations. Also involved and provided a factual review and helped edit the manuscript. All authors discussed the results and commented on the manuscript. All the authors involved for Completing the Review Comments.

Funding

No funding was received to assist with the preparation of this manuscript.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current Study.

Declarations

Competing interests

The authors declare no competing interests.

Received: 15 May 2023 Accepted: 17 September 2023
Published online: 03 October 2023

References

1. Xiong, G., Wang, S., Yan, G., & Li, J. (2023). Reinforcement learning for dynamic dimensioning of cloud caches: A restless bandit approach. IEEE/ACM Transactions on Networking, 1–15. https://doi.org/10.1109/TNET.2023.3235480
2. Cui T, Yang R, Fang C, Yu S (2023) Deep reinforcement learning-based resource allocation for content distribution in IoT-edge-cloud computing environments. Symmetry 15(1):217. https://doi.org/10.3390/sym15010217
3. Liu J, Yao W, Wang C, Yang Q (2023) Provisioning network slice for mobile content delivery in uncertain MEC environment. Comput Netw 224:109629. https://doi.org/10.1016/j.comnet.2023.109629
4. Andjamba, T. S., & Zodi, G.-A. L. (2023). A load balancing protocol for improve dvideo on demand in SDN-based clouds. In17th International Conference on Ubiquitous Information Management and Communication (IMCOM), 2023 (pp. 1–6). https://doi.org/10.1109/IMCOM56909.2023.10035591

5. Banu SS, Balasundaram SR (2021) Cost optimization for dynamic content delivery in cloud-based content delivery network. J Inform Technol Res 14(4):18–32. https://doi.org/10.4018/JITR.2021100102
6. Lee CKM, Ng CK, Chung SY, Keung KL (2023) Cloud-based Cyber-Physical Logistics System with Nested MAX-MIN Ant Algorithm for e-commerce logistics. Expert Syst Appl 211:118643. https://doi.org/10.1016/j.eswa.2022.118643
7. Zhang, J., & Yeh, E. (2023). 'Congestion-aware routing and content placement in elastic cache networks,' arXiv preprint arXiv:2303.01648.
8. Abbasi M, Khosravi MR, Ramezani A (2023) Intelligent resource management at the network edge using content delivery networks. Enterprise Information Systems 17(5):2037159. https://doi.org/10.1080/17517575.2022.2037159
9. Jayakumar, S., Sheelvanthmaths, P., & Akki, C. B. (2022). Technical analysis of content placement algorithms for content delivery network in cloud. International Journal of Electrical and Computer Engineering, 12(1), 489. https://doi.org/10.11591/ijece.v12i1.pp489-496
10. Xing R, Su Z, Xu Q, Zhang N, Luan TH (2022) Secure content delivery for connected and autonomous trucks: A coalition formation game approach. IEEE Trans Intell Transp Syst 23(11):20522–20537. https://doi.org/10.1109/TITS.2022.3184973
11. Goswami V, Panda G (2022) Multimedia content delivery services in the cloud with partial sleep and abandonment. J Supercomput 78(15):17178–17201. https://doi.org/10.1007/s11227-022-04532-1
12. Koch, J., & Hao, W. (2022). Practical applications of edge computing to accelerate cloud hosted webcontent. In IEEE world AIIo Congress, 2022, (256–263). https://doi.org/10.1109/AIIoT54504.2022.9817313
13. Morel, A. E., Calyam, P., Qu, C., Gafurov, D., Wang, C., Thareja, K., Mandal, A., Lyons, E., Zink, M., Papadimitriou, G., & Deelman, E. (2023). Network services management using programmable data planes for visual cloud computing. In International Conference on Computing, Networking and Communications (ICNC), 2023 (pp. 130–136). https://doi.org/10.1109/ICNC57223.2023.10074183
14. Karim, F. A., Mohd Aman, A. H. Mohd., Hassan, R., & Nisar, K. (2022). A survey on information-centric networking with cloud Internet of things and artificial intelligence. Wireless Communications and Mobile Computing, 2022, 1–11https://doi.org/10.1155/2022/7818712
15. Manzoor, A., Ahsan Kazmi, S. M., & Hong, C. S. (2022). Efficient content delivery to vehicles passing through a wireless-enabled traffic signal system. In International Conference on Information Networking (ICOIN), 2022 (pp. 460–463). https://doi.org/10.1109/ICOIN53446.2022.9687241
16. Alberro L, Castro A, Grampin E (2022) Experimentation environments for data center routing protocols: A comprehensive review. Future Internet 14(1):29. https://doi.org/10.3390/fi14010029
17. Akbari MR, Barati H, Barati A (2022) An efficient gray system theory-based routing protocol for energy consumption management in the Internet of Things using fog and cloud computing. Computing 104(6):1307–1335. https://doi.org/10.1007/s00607-021-01048-z
18. Zhang J, Li S, Wang C (2022) A secure dynamic content delivery scheme in named data networking. Security Communication Netw 2022:1–15. https://doi.org/10.1155/2022/6304927
19. Lahande PV, Kaveri PR (2022) Reinforcement learning applications for performance improvement in cloud computing—A systematic review. Sustainable Advanced Comput 2021:91–112
20. Zolfaghari B, Srivastava G, Roy S, Nemati HR, Afghah F, Koshiba T, Razi A, Bibak K, Mitra P, Rai BK (2021) Content delivery networks: State of the art, trends, and future roadmap. ACM Comput Surv 53(2):1–34. https://doi.org/10.1145/3380613
21. Asheralieva A, Niyato D (2019) Game theory and Lyapunov optimization for cloud-based content delivery networks with device-to-device and UAV-enabled caching. IEEE Trans Veh Technol 68(10):10094–10110. https://doi.org/10.1109/TVT.2019.2934027
22. Sun P, AlJeri N, Boukerche A (2020) DACON: A novel traffic prediction and data-highway-assisted content delivery protocol for intelligent vehicular networks. IEEE Trans Sustain Comput 5(4):501–513. https://doi.org/10.1109/TSUSC.2020.2971628
23. Chen M, Wang L, Chen J, Wei X, Lei L (2019) A computing and content delivery network in the smart city: Scenario, framework, and analysis. IEEE Network 33(2):89–95. https://doi.org/10.1109/MNET.2019.1800253
24. Islam N, Haseeb K, Rehman A, Alam T, Jeon G (2023) An adaptive and secure routes migration model for the sustainable cloud of things. Clust Comput 26(2):1631–1642. https://doi.org/10.1007/s10586-022-03677-1
25. Sadeghi A, Wang G, Giannakis GB (2019) Deep reinforcement learning for adaptive caching in hierarchical content delivery networks. IEEE Trans Cogn Commun Netw 5(4):1024–1033. https://doi.org/10.1109/TCCN.2019.2936193
26. Sinky H, Khalfi B, Hamdaoui B, Rayes A (2019) Adaptive edge-centric cloud content placement for responsive smart cities. IEEE Netw 33(3):177–183. https://doi.org/10.1109/MNET.2019.1800137
27. Liu Y, Lu D, Zhang G, Tian J, Xu W (2019) Q-learning based content placement method for dynamic cloud content delivery networks. IEEE Access 7:66384–66394. https://doi.org/10.1109/ACCESS.2019.2917564
28. Alghamdi F, Mahfoudh S, Barnawi A (2019) A novel fog computing based architecture to improve the performance in content delivery networks. Wirel Commun Mob Comput 2019:1–13. https://doi.org/10.1155/2019/7864094
29. Zhao, J., Liang, P., Liufu, W., & Fan, Z. (2020). Recent developments in content delivery network: A survey. In Parallel architectures, algorithms and programming, Revised Selected Papers 10: 10th International Symposium, PAAP 2019, Guangzhou, China, December 12–14, 2019 (pp. 98–106).
30. Hasan K, Jeong S-H (2019) Efficient caching for data-driven IoT applications and fast content delivery with low latency in ICN. Appl Sci 9(22):4730. https://doi.org/10.3390/app9224730
31. Pervej, M. F., Jin, R., Lin, S.-C., & Dai, H. (2022). 'Efficient Content Delivery in User-Centric and Cache-Enabled Vehicular Edge Networks with Deadline-Constrained Heterogeneous Demands,' arXiv preprint arXiv:2202.07792.
32. Mishra, S., Kumar, M., Singh, N., & Dwivedi, S. (2022). A survey on AWS cloud computing security challenges and solutions. In6th International Conference on Intelligent Computing and Control. Systems (ICICCS), 2022 (pp. 614–617). https://doi.org/10.1109/ICICCS53718.2022.9788254
33. Shoaib U, Arshad MJ, Khattak HA, Ezat Ullah M, Almogren A, Ali S (2022) Fast dataaccess through nearestlocation-basedreplicaplacement. Sci Program 2022:1–13. https://doi.org/10.1155/2022/2496269
34. Ghaznavi M, Jalalpour E, Salahuddin MA, Boutaba R, Migault D, Preda S (2021) Content delivery network security: A survey. IEEE Commun Surveys Tutorials 23(4):2166–2190. https://doi.org/10.1109/COMST.2021.3093492
35. Esfandiari S, Rezvani MH (2021) An optimized content delivery approach based on demand–supply theory in disruption-tolerant networks. Telecommun Syst 76(2):265–289. https://doi.org/10.1007/s11235-020-00711-8

Publisher's Note