Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

A cost-efficient content distribution optimization model for fog-based content delivery networks

Prateek Yadav[1,2*] and Subrat Kar[1]

Abstract

The massive data demand requires content distribution networks (CDNs) to use evolving techniques for efficient content distribution with guaranteed quality of service (QoS). The distributed fog-based CDN model, with optimal fog node placements, is a suggested aproach by researchers to meet this demand. While many studies have focused on improving QoS by optimizing fog node placement, they have rarely considered the impact on content distribution, affected by placement, usage changes, and delivery rates. Therefore, the practical approach to fog node placement for CDN services must examine its impact on content distribution. Further, current research on fog-based CDN lacks formal methods to address key challenges: R1) strategic placement of fog nodes to process end-user requests; R2) construction of a content distribution path with guaranteed QoS; R3) cost minimization of building a fog-based CDN model. We construct this as a joint optimization problem by considering four parameters: geographical regions, open public Wi-Fi access points (OPWAPs) locations, QoS, and cost to achieve research objectives R1–R3. As a solution, we propose a dual-step framework. First, a heuristic for optimal fog node placement based on geographic regions and OPWAP locations is proposed. Second, we propose two algorithms, Greedy Performance-based Node Selection (*GPDS*) and Greedy Fog Node Selection algorithm (*GFNSA*), for selecting fog nodes, minimizing the cost of building a fog-based CDN while achieving optimal content distribution paths. The results demonstrate that the proposed methods outperform the baseline techniques and provide near-optimal solutions to the problem.

Keywords Fog computing, Content delivery networks (CDNs), Internet of things, Dual-step approach, Greedy techniques

Introduction

Currently, over two-thirds of the global population has access to the internet, with video content driving up to 80% of the data demand [1]. Content distribution networks (CDNs) help in handling such data demand and ensure quality of service (QoS) by bringing content closer to the user requesting it. The standard CDN platform operates in a two-tier system model, where a cluster of edge servers coordinates with the content provider/origin server to distribute content. However, this architecture has limitations that affect its scalability, manageability, mobility, and cost [2–8]. Scalability is limited as edge servers must be deployed in response to increasing data demand, while management becomes increasingly difficult due to an increasing number of edge servers. Mobility is restricted due to the fixed geographical location of edge servers, and the same cannot be redeployed based on demand. Finally, the architecture cost increases due to the numerical growth of edge servers.

To overcome the limitations of traditional CDN architectures, cloud-centric CDNs have been developed.

*Correspondence:
Prateek Yadav
prateek.yadav@ee.iitd.ac.in
[1] Department of Electrical Engineering, Indian Institute of Technology, Delhi, New Delhi 110016, India
[2] School of Artificial Intelligence, Bennett University, Greater Noida, Uttar Pradesh, India

These CDNs are cost-effective and easily deployable since they do not require edge servers. In cloud-centric CDNs, content replication and caching occur across various cloud locations hosted by multi-cloud service providers like Meta CDN or within the infrastructure of the same-cloud service providers, including Azure CDN, Google CDN, or Amazon's Cloud Front. Nevertheless, these data placement locations have an adverse impact on QoS [9], particularly for applications having time-sensitive requirements [10], due to the substantial geographical distance between users and cloud data center locations.

Therefore, edge and fog computing approaches are essential to maintain QoS [11, 12]. These technologies share the common goal of providing services closer to end users by offering nearby resources. Still, there is a clear distinction in their use cases and scope [13–15]. For instance, some applications like augmented reality (AR) are better suited for edge computing, whereas other technologies like CDN are better suited for fog computing [16, 17]. In this fog computing, fog nodes are the primary functional units to reduce the shortcomings of the traditional architecture [18, 19] where these nodes are used to deliver services at the edge of the network [20]. Although there are several research studies about the placement of fog nodes [21, 22], there is still a lack of research in the fog-integrated CDN domain. In this study, we explore the challenges that arise from the integration of fog computing with CDNs. Our analysis focuses on addressing the following research questions: (R1) How do we find the optimal placement location to deploy fog nodes to process the end-user content request effectively? (R2) Once the nodes are optimally deployed, how can we select them effectively to build an optimal content distribution path that ensures QoS for end users?, and (R3) How to reduce the total cost of building such a distributed fog-based CDN? We evaluate our solutions with respect to their ability to improve QoS, cost-effectiveness, and scalability.

Similarities exist between the placement of replica servers and the fog nodes when building a fog-based CDN. However, they differ in key parameters; i.e., replica servers prioritize low link usage and high network bandwidths, while fog nodes may handle locally relevant compute requests with potential minimal data transfers. Thus, crucial factors for fog nodes include (a) low network latency, (b) optimal placement, and (c) high availability. It is necessary for low-cost content distribution to ensure optimal placements of fog nodes in locations. While it is possible to *assume* that the placements of fog nodes are optimal and contents can be distributed [23–25], better solutions are possible. That is, studies on fog node placements, such as [21, 22, 26–28], presented different techniques, e.g., mixed-integer programming. However, most of the research works related to the fog-based CDN domain lack to formally address the research problems (R1)-(R3) together.

We present a cost-effective optimization model for content distribution in the fog-based CDN domain, denoted as ``Fog-CDN". In contrast to earlier works [23–25] that assumed optimal fog node locations and worked on QoS optimization, our methodology minimizes the objective function cost (overall cost), considering constraints on placement and distribution parameters. This presented Fog-CDN formal problem is a nonlinear integer problem (also *NP-hard*). Hence, we reduce its complexity by proposing a dual-step framework. In this, as a first step, we present a heuristic for the optimal placement of fog nodes. Then, we propose two greedy algorithms for content distribution based on the efficient selection of fog nodes.

We validate the model by analyzing the open public Wi-Fi access points (OPWAPs) real-world dataset, with the objective of optimizing the locations of fog nodes. The presented fog node placement algorithm clusters the OPWAPs geographically and generates the optimal service sub-regions with ideal locations for placing fog nodes closer to the end-users for Fog-CDNs. Once optimal fog node placements are observed, our proposed greedy heuristics, called Greedy Performance-based Node Selection (*GPDS*), and the Greedy Fog Node Selection Algorithm (*GFNSA*) optimize the content distribution through the efficient selection of fog nodes. Our findings show that the Fog-CDN model can significantly improve performance and reduce delays for content providers, while also minimizing the disadvantages of traditional P2P networks.

In summary, this paper builds a Fog-CDN optimization model and provides a solution to problems (R1)–(R3), that have not been addressed in earlier studies. Our contributions are the following:

1. We present a mathematical model for Fog-CDN, optimizing both placement and distribution through cost minimization of the objective function. A thorough discussion is presented for the Fog-CDN model's cost-effectiveness related to placement and distribution parameters.
2. To reduce model's complexity, we first discuss a fog-node placement algorithm by optimizing real-world OPWAP locations to show the feasibility of fog-node placement for the model. Second, we propose two greedy algorithms for efficient content distribution by selecting fog nodes (*GPDS* and *GFNSA*) for end users.
3. Finally, we show the model performance by comparing them with the baseline algorithms on various

CDN parameters for a concrete analysis. Our results suggest that the proposed approaches outperform the baseline techniques and offer near-optimal solutions for the model.

We structured the paper as follows. "Related work" section presents the related work. "System model and problem formulation" section describes the model and presents a proof of *NP-hardness* of the problem. "The proposed algorithms" section discusses the algorithmic solutions. "Performance evaluation" section presents methods and settings to compare the work and discusses the observed results. "Conclusion and future work" section concludes the paper.

Related work

Content distribution networks related research has focused primarily on traditional server placement solutions or cloud-based approaches, which may not be sufficient to address current technological advancements. Several studies have proposed solutions for edge server placement, such as Li et al. [29] discussed a server placement method for ultra-dense networks. However, the proposed model deals only with the placement of servers. Similarly, Mohan et al. [30] introduced the "Aneshak" framework for optimal server placement through an integer linear programming (ILP) model. However, they did not discuss on effective content distribution and its impact on placement. Their assessment of placement models was conducted through simulation-based methods, employing techniques like grid and random. Lahderanta et al. [28] proposed a capacitated location-allocation method for edge computing server placement. Zhang et al. [27] analyzes the edge-cloud architecture to determine the placement of edge nodes and related services. Although their study provides a generalized approach to placing services on edge nodes, it lacks details on effective content distribution and how to use it to construct such a distributed content network. Furthermore, the study lacks research on critical content distribution parameters. In recent years, considerable research attention has been focused on fog node deployments. Silva et al. [21] addressed the challenge of locating fog nodes in a network infrastructure to accommodate end-user demands efficiently, particularly in the context of user mobility and varying workload demands. Whereas, Wang et al. [22] discussed the crucial challenge of effectively deploying fog nodes in intelligent manufacturing scenarios within a fog computing platform. Recognizing the heterogeneous nature of fog nodes and the dynamic characteristics of intelligent manufacturing, a novel strategy, termed TSBP (Time–Space-Based Deployment), was proposed. Furthermore, Brogi et al. [26] investigated the complex challenge of deploying IoT applications in fog infrastructures, considering QoS and cost factors.

The solution enables the evaluation of multi-component application deployments in fog infrastructures. However, more research work is required to develop a Fog-CDN model, incorporating the problems R1–R3.

In existing research on fog-based CDN, Ghalehtaki et al. [25] formulated an optimization problem based on a bee colony algorithm to place micro-caches in the fog domain optimally. They suggested a meta-heuristic algorithm as the solution. In the same context, the authors [24] used the content-aware replication framework based on an information-centric network (ICN) to present a fog-based CDN model to improve CDN efficiency. They conclude that it is possible to achieve higher QoS by adopting an ICN approach for a CDN. However, research conducted by [31] suggests that the decrease in the delay is not significant for cache-enabled and tolerable-delay applications, as seen in scenarios like ``smart home" and ``smart factory".

Parallel to our work, the author [23] has presented a framework for developing a fog-based browser for fog-based CDN. The system provides mechanisms to handle and redirect user requests to appropriate fog nodes. However, no formal foundation has been provided to build such model. Our methods and algorithms can be readily used by fog-based CDN frameworks to make the system more comprehensive.

The summary of the comparative study is presented in Table 1.

System model and problem formulation

The following section discusses the model and various problem definitions to formulate the problem. The problem is formulated as a non-linear integer programming model, which is an *NP-hard* problem.

Model and problem definitions

We define the problem as placing fog nodes in a region and building a distribution path for the Fog-CDN model, abstractly shown in Fig. 1. It comprises a centralized cloud-based content provider (CCP) node and multiple fog nodes placed in a fog stratum.

The fog stratum is composed of various geographic regions, each containing several locations for fog node placement, which coordinate with multiple edge network access points (i.e., OPWAPs). The nodes are interconnected by upload and download links, each with varying link capacity and cost, from the user to the cloud node through the OPWAPs and fog nodes. We define the

Table 1 Key features and the contributions of various studies related to fog-based CDN model

Papers	Major Contribution	Criteria for fog-based CDN analysis						
		Metrics	FO	CDA	CoA	IoT Model	RW Data	GeoA
Li et al. [29]	Discussed a framework to place the edge servers optimally in UDN.	Cost	✓	✗	✓	✓	✗	✗
Mohan et al. [30]	Suggested an optimal edge server placement method called 'Anveshak' to place edge servers.	User satisfaction	✓	✗	✗	✓	✓	✗
Zhang et al. [27]	Addressed the interdependence of edge server deployment and service placement in mobile-edge computing (MEC).	Profit	✓	✗	✗	✓	✓	Partial
Lahderanta et al. [28]	Proposed a method for edge computing server placement using capacitated location allocation.	Load balancing	✓	✗	✗	✓	✓	Partial
Silva et al. [21]	Presented a model for locating the fog node by proposing a mixed-integer programming model.	Multi-objective	✓	✗	✗	✓	✓	✗
Wang et al. [22]	Discussed the crucial challenge of effectively deploying fog nodes in intelligent manufacturing scenarios within a fog computing platform.	QoS (Response Time)	✓	✗	✗	✓	✗	Partial
Brogi et al.[26]	Discussed the cost of deploying IoT applications for fog infrastructure.	Cost	✗	✗	✗	✓	✗	✗
Ghalehtaki et al. [25]	Presented a model to place micro-caches optimally in fog-based CDN using bee-colony approach.	Micro-caches location	✓	✗	✗	✓	✗	✗
Alghmadi et al. [24]	Suggested a fog-based CDN model based on ICN and content-aware replication approach.	Delay	✗	✗	✗	✓	✗	✗
Ibrahim et al. [23]	Suggested a fog-based browser for the fog-based CDN.	Load-time of web pages	✗	✗	✗	✓	✗	✗
This work	**Presented a dual-step framework for the Fog-CDN model based on the optimal fog node deployments and building a distribution path.**	**Cost and QoS**	✓	✓	✓	✓	✓	✓

FO Formal Optimization, *CDA* Content Distribution Analysis, *CoA* Cost Analysis, *RW Data* Real-World Data, *GeoA* Geographical Analysis

problem as follows: for every fog node n, there exists a binary decision variable $\mathrm{D}_{(\mathrm{r,p,n})}$ such that

$$\mathrm{D}_{(\mathrm{r,p,n})} = \begin{cases} 1 & \text{if a fog node n is placed at locally p in region r,} \\ 0 & \text{otherwise.} \end{cases}$$

Here, $r \in \mathcal{R}$ represents the set of regions denoted as $\mathcal{R} = \{1, 2, 3, \ldots, n_r\}$, $p \in \mathcal{P}$ represents the set of potential locations of fog node placement, defined as $\mathcal{P} = \{1, 2, 3, \ldots, n_p\}$ and $n \in \mathcal{N}$ represents the set of fog nodes, defined as $\mathcal{N} = \{1, 2, 3, \ldots, n_f\}$, respectively. We define the placement cost of each fog node as F_n^{Cost}. Further, to reduce the problem complexity, we assume that the set of potential locations $\mathcal{P}$ represents the locations of OPWAPs, which are already placed. In the presented Fog-CDN model, the term *potential* refers to the availability of candidate locations for fog node placement, such as existing OPWAPs. The selection of optimal fog node placements is based on the evaluation of these potential locations and the determination of the most suitable that meets the optimization objectives [21, 22, 26]. To define the location of each fog node within a region, we use the notation G_p^r to represent the set of fog nodes placed at locations p and region r, where p and r belong to sets $\mathcal{P}$ and $\mathcal{R}$, respectively. For instance, G_p^1 denotes the set of nodes placed at different locations in region 1, which can be expressed as $\{G_1^1, G_2^1, G_3^1, \ldots, G_p^1\}$. We consider Z_0 as the cloud-based content provider (CCP) that delivers content to multiple fog nodes. $\mathcal{U}$ is the set of users, defined as $\mathcal{U} = \{1, 2, \ldots, u\}$, where u represents the total number of users in all regions. Further, let $\mathcal{M}$ denote the set of all nodes in the model, defined as: $\mathcal{M} = \mathcal{N} \bigcup \mathcal{U} \bigcup \{Z_0\}$. Finally, the parameter X_r represents the minimum quantity of fog nodes required for effective content distribution within a region. The calculation of X_r depends on the objectives, network capacity, and desired level of service, which are determined by experts experience [22].

We denote the set of links as $\mathcal{L}$, which represent all the distribution paths between nodes and users. In this, L_d denotes the set of links that are utilized to distribute requested contents, where $L_d \subseteq \mathcal{L}$. Also, we introduce a binary decision variable E_l for every link $l \in L_d$, where $E_l = 1$ denotes the selection of link l. Further, d_l denotes the delay incurred by every link $l \in L_d$. C_l represent the content replication cost for every link $l \in L_d$. d_{u_l} presents the sum of all the delays observed by a

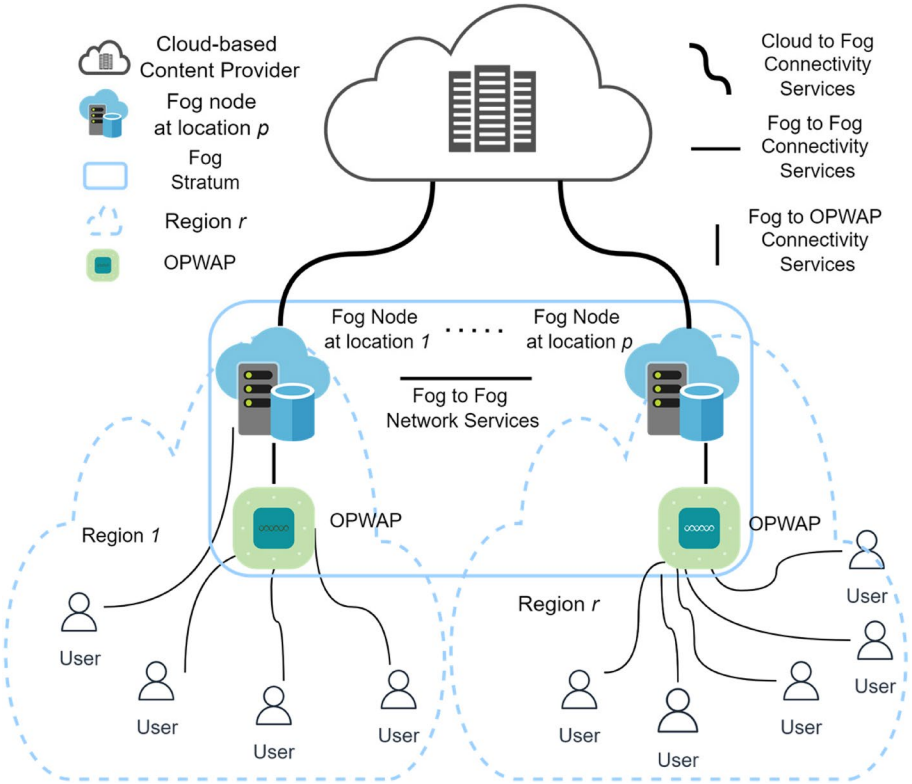


Fig. 1 Abstract model of fog CDN

user u ($u \in \mathcal{U}$ and $l \in L_d$) while receiving the requests from nodes through links. It serves as the overall average delay. This delay, d_{u_l}, is crucial to provide necessary QoS to end-users. At the last, we define various delay requirements of data-driven services through set.

$\mathcal{S} = \{S_1, S_2, \ldots, S_s\}$, which denote the overall service delays. We assume[1] that the delay between the OPWAPs and the fog node is negligible compared to the other two delays, (a) between the user and the fog node, and (b) between the fog node and the CCP node. The content distribution cost is composed of two constituents: (a) *cost* associated with the link's transmissions, i.e., uploading, downloading, and content replication cost, and (b) the storage cost associated with the repository to store contents.

Calculating cost related to upload and download is a straightforward process. However, calculating the cost of replication requires careful analysis, since the nodes must evaluate the cost of replication for each request. Adopting the method from [35], we define the overall content replication cost C_l as follows: whenever content is requested by a user, it is fetched and stored on various fog node sites before being delivered to the user. Each time, the replication cost is calculated by the nodes to assess the related cost. We consider three types of costs to represent the replication cost C_l: upload (ingress), download (egress), and storage costs. The replication cost between the CCP node i to fog node j is denoted by C_l^{ij}, where $i, j \in \mathcal{M}$, and $l \in L_d$, is defined as:

$$C_l^{ij} = \left(st_j + up_j \cdot f + do_i \cdot f\right) \cdot W, \tag{1}$$

where st_j denotes the cost per GB of storage, depend on the size of the replica stored on the fog node,[2] up_j signifies the cost per GB for upload traffic levied by the fog node, do_i is the unit download cost levied by a node i (CCP or fog), W represents the size of content replica, and f denotes the frequency of content updates. The cost of replication from fog nodes to users is expressed as: The

[1] This assumption is based on the understanding that distance (geographical or hop distance) significantly impacts delay [32–34] and the overall distance between the user and the fog node is greater than the distance between OPWAP and the fog node.

[2] Usually, storage costs are considered as an incremental function of length–time, and it is chargeable per unit time (where unit would be in the range of months). The cost of storage is pre-calculated for each problem instance.

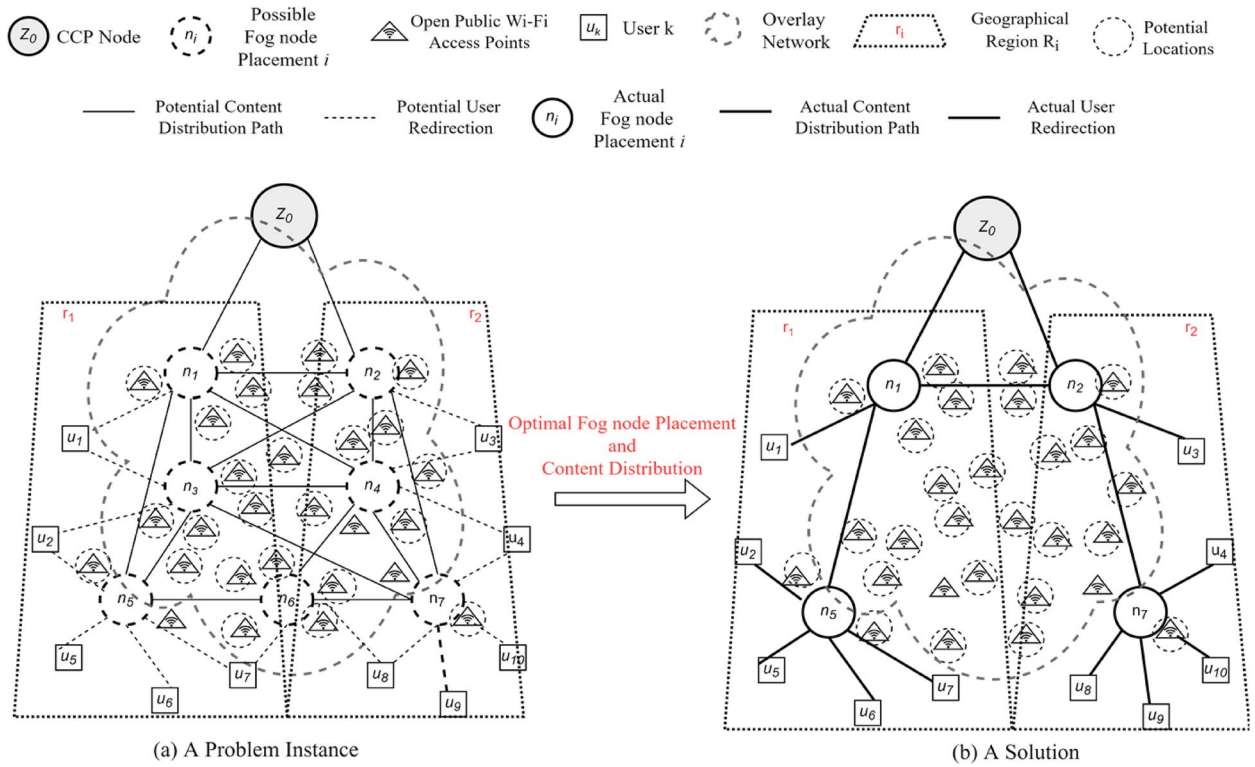


Fig. 2 An example of problem scenario

replication cost from the fog nodes to the user can be defined as:

$$C_l^{jk} = w_k.do_j, \tag{2}$$

where end-user $u_k \in \mathcal{U}$ requested w_k bytes.

Figure 2 illustrates the optimization problem of Fog-CDN. In Fig. 2a, solid lines indicate feasible content delivery paths between Z_0 and fog nodes n_1 to n_7 placed across two regions, r_1 and r_2. The dashed circle indicates potential fog node placements, while each triangle represents an OPWAP placed in a specific region. Within this scenario, we presume that every connected user u_k from a site has a path. However, only certain paths fulfill the QoS requirement for each user request, as shown through dashed line. Figure 2b illustrates an instance solution of the problem, where n_1 is designated as the fog node site to handle requests from u_1 and distribute content from Z_0 to n_2, n_5, and n_7. n_5 is selected as the fog node site to serve requests from u_2, u_5, u_6, and u_7, and so on. The total cost of the solution incorporates placement and distributions cost, as described before.

Problem objective

The goal of the problem is to minimize the cost of building a Fog-CDN for end users while ensuring the QoS requirements for providing data services. This total cost includes both placement and content distribution cost. We formulate the total cost (A) of building the Fog-CDN as follows:

$$\text{total cost}(A) = \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}}\sum_{n\in\mathcal{N}} D_{(r,p,n)} \cdot F_n^{Cost} + \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}}\sum_{n\in\mathcal{M}}\sum_{l\in L_d} D_{(r,p,n)} \cdot E_l \cdot C_l \tag{3}$$

In the above formulation, the first part signifies the placement cost of fog nodes, and the second part signifies the cost of building a content distribution path. Thus, the overall minimization problem is defined as follows:

$$\min(\text{total cost}(A)); \tag{4}$$

subject to:

$$E_l = \begin{cases} 1 \text{ if } d_{u_l} \le S_s, \quad \forall u \in \mathcal{U},\ \forall l \in L_d, \text{ and } \forall S_s \in \mathcal{S} \\ 0 \text{ otherwise.} \end{cases} \tag{5}$$

$$\sum_{n\in\mathcal{N}} D_{(r,p,n)} \ge X_r, \ \forall r \in \mathcal{R}, \ \textit{and} \ \forall p \in \mathcal{P} \tag{6}$$

Table 2 Table of problem notations

Problem notations	Definition
$\mathcal{R}$	set of regions
$\mathcal{P}$	set of all potential locations for fog node placements
$\mathcal{N}$	set of fog nodes
$\mathcal{M}$	set of total nodes, which includes a set of users $\mathcal{U}$, set of fog node $\mathcal{N}$, and cloud content provider node $\boldsymbol{Z}_0$
$\boldsymbol{Z}_0$	a cloud content provider node
$\boldsymbol{D}_{(r,p,n)}$	a binary node decision variable defined over the set of regions, the set of locations, and the set of fog nodes
$\boldsymbol{G}_{\boldsymbol{p}}^{\boldsymbol{r}}$	represent the set of fog nodes placed at locations $\boldsymbol{p}$ in region $\boldsymbol{r}$
$\mathcal{U}$	a set of users
$\mathcal{L}$	a set that defines all the connected links between users and nodes
$\boldsymbol{L}_{\boldsymbol{d}}$	a set that defines content distribution paths, where $\boldsymbol{L}_{\boldsymbol{d}} \subseteq \mathcal{L}$
$\boldsymbol{E}_{\boldsymbol{l}}$	a binary decision variable defined over each link $\boldsymbol{l} \in \boldsymbol{L}_{\boldsymbol{d}}$
$\boldsymbol{F}_{\boldsymbol{n}}^{\boldsymbol{Cost}}$	fog node placement cost,where $\boldsymbol{n} \in \mathcal{N}$
$\boldsymbol{C}_{\boldsymbol{l}}$	denotes the associated replicating content cost for every link $\boldsymbol{l} \in \boldsymbol{L}_{\boldsymbol{d}}$
$\boldsymbol{W}$	denotes the size of content replica
$\boldsymbol{w}_{\boldsymbol{k}}$	denotes the end-user's $\boldsymbol{u}_{\boldsymbol{k}}$ content request in bytes
$\boldsymbol{st}_{\boldsymbol{j}}$	denotes the cost of storing content at node $\boldsymbol{j}$
$\boldsymbol{up}_{\boldsymbol{j}}$	denotes the cost of uploading content at node $\boldsymbol{j}$
$\boldsymbol{do}_{\boldsymbol{j}}$	denotes the cost of downloading content at node $\boldsymbol{j}$
$\boldsymbol{d}_{\boldsymbol{l}}$	denotes the incurred delay for every link $\boldsymbol{l} \in \boldsymbol{L}_{\boldsymbol{d}}$
$\boldsymbol{d}_{\boldsymbol{u}_{\boldsymbol{l}}}$	the sum of all the delays observed by a user $\boldsymbol{u}$ ($\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{l} \in \boldsymbol{L}_{\boldsymbol{d}}$) while receiving the requests from nodes through links, represented as overall average delay
$\mathcal{S}$	a set that defines the overall delay requirement of various services
$\boldsymbol{S}_{\boldsymbol{s}}$	denotes the delay requirement of a service $\boldsymbol{S}_{\boldsymbol{s}} \in \mathcal{S}$
$\boldsymbol{X}_{\boldsymbol{r}}$	denotes the minimum quantity of fog nodes required for effective content distribution within a region $\boldsymbol{r}$, where $\boldsymbol{r} \in \boldsymbol{R}$

$$D_{(r,p,n)} \in \{0,1\}, \quad \forall r \in \mathcal{R},\ \forall p \in \mathcal{P},\ and\ \forall n \in \mathcal{N} \tag{7}$$

$$E_l \in \{0,1\},\ \forall l \in L_d \tag{8}$$

Constraint (5) guarantees the QoS (overall average delay) for end-users by selecting only those content distribution links that satisfy the CDN service delay requirements. Constraint (6) guarantees fog nodes cover each region. The binary nature of the decision variable for nodes are guaranteed by constraint (7), and the same is achieved for links through constraint (8). The problem defined above is a joint problem of placing fog nodes and building a distribution path, an instance of uncapacitated facility location problem [36]. We have implemented and solved the above formulations in the CPLEX solver [37]. In general, such problems are hard to solve, which we prove in the following subsection. To address the complexity of such a problem, we need a heuristic or meta-heuristic algorithm to solve it. In Table 2, we summarize the notation used in this paper.

NP-Hardness proof

In the field of network design, placing servers and building paths for content distribution has been widely acknowledged as a problem that is hard to solve. This research presents evidence that the Fog-CDN problem is an *NP-hard* problem. For this, we reduce problem to the quadratic assignment problem (QAP) [38, 39]. We follow the mapping given as follows.

As defined previously, $\mathcal{R}$ and $\mathcal{P}$ denote the set of regions and fog node placement locations. We define $\mathcal{T}$, as $\mathcal{T} = \mathcal{P} \times \mathcal{R}$ so that for any $(i, r) \in \mathcal{T}$, we have $i \in \mathcal{P}$ and $r \in \mathcal{R}$, which denotes the mapping of locations in $\mathcal{P}$ and regions in $\mathcal{R}$. Each decision on fog node placement is binary and indicates the placement of nodes in a specific region. b_{ir} presents the QAP facility deployment cost, denoting cost of placing fog node at i in region r. To integrate the QAP distance matrix d_{ij} and flow matrix f_{ij} as the problem's delay and distribution cost matrix, respectively, we define it as follows. Consider l_{ij} as the set of links in QAP, where every link in the set signifies the connection between fog nodes i to j. f_{ij} is the content distribution cost associated with a link between fog node i to j. Further, d_{ij} denotes the incurred link delay between fog node i to j.

Using the given definitions, it becomes apparent that the Fog-CDN problem's placement and distribution cost can be represented as the deployment cost and flow

matrices of the QAP, respectively. The problem's QAP formulation is as follows.

$$
\begin{aligned}
&min \sum_{i,r\in\mathcal{P}} b_{ir}x_{ir} + \sum_{i,j\in\mathcal{N}} \sum_{r\in\mathcal{R}} f_{ij}\cdot x_{ir}\cdot x_{jr} \\
&\text{Subject to: } \sum_{r\in R} x_{ir} = 1, \forall i \in \mathcal{N} \\
&\sum_{i\in N} x_{ir} = 1, \forall r \in \mathcal{R} \\
&x_{ir} \in \{0,1\}, \forall i, r \in \mathcal{P} \\
&d_{ij} \leq S_s, \forall (i,j) \in \mathcal{L},\ \forall S_s \in \mathcal{S}
\end{aligned} \tag{9}
$$

The objective of the Fog-CDN problem is to reduce the overall cost associated with the deployment of fog nodes and the distribution of content. Decision variable x_{ir} indicates the fog node at location i in region r. The initial two constraints guarantee the coverage of each region and the assignment of each node location to a specific region. The subsequent constraint mandates the binary nature of x_{ir}. The final constraint imposes restrictions on the delay for effective content distribution.

QAP is widely recognized as *NP-hard*, and the Fog-CDN problem reduction to QAP suggests that it is also an *NP-hard* problem. Hence, it is an *NP-hard* problem.

The proposed algorithms

This section discusses the dual-step framework to reduce the complexity of the problem. First, we describe the process of finding optimal placement locations in a region for fog nodes that is closer to OPWAPs and end users. Second, once the optimal location of the fog nodes is determined, we propose a set of greedy algorithms for fog node selection such that efficient content distribution can be achieved for the end users while maintaining QoS.

An optimal fog node placement algorithm

To address the complexity of deployments in a region, the first algorithm adopts a machine learning technique, *k-means* algorithm. It helps in determining optimal clusters in a region by analyzing potential fog node locations $\mathcal{P}$ (i.e., OPWAP locations), along with the *Voronoi* principle. The *Voronoi* principle calculates the nearest neighbors so that the optimal location is closer to its corresponding OPWAP locations (in an optimal sub-region). Lines 3–7 of Algorithm 1 present the above process. This way, we reduce the problem complexity as well as achieve the optimal locations for fog node placements while minimizing the distance between the OPWAPs and the fog nodes. In general, we provide seeds for the *Voronoi* algorithm to build it. However, we initiate the generators for *Voronoi* method by leveraging the *k-means* technique. The advantage of utilizing the suggested approach is that it does not entail the selection of generator locations that are near the sub-region of a region. Instead, it effectively determines the optimal locations employing the *k-means* technique. Furthermore, we employed the *k-means* clustering technique in our evaluation because it was deemed suitable for the distribution of OPWAPs. However, the applicability of various clustering algorithms may differ due to the data distributions in different situations. The presented heuristic works as follows:

Algorithm 1 PFOLR: an algorithm for Placing Fog-node in Optimal Locations of a Region

***Inputs**: A region $R_0 \subseteq \mathcal{R}$, and set of potential OPWAP locations $P_0 \subseteq \mathcal{P}$*
***Output:** A set of optimal fog node locations for the region*

1: $V = \{(R_0, P_0)\}$
2:. ***for all** r in |V | do*
3: Call *CGroup*(R_0, P_0, k ,r)
4: Cluster (R_0, P_0) into k groups
5: $Ccenters \leftarrow k$
6: *[Gregion,Gseed]* =CPARTITION(R_0,*Ccenters*)
7: Optimal placement locations ← *Gseed*
8: ***end for***

1. Step 1 presents the initialization of the algorithm by inputting region R_0 and OPWAP locations P_0, which denotes the potential fog node placement locations.
2. Steps 2–5, first, we initialize the variable r, as $r = 1$. It provides the initial centroid randomization for the *k-mean* algorithm. Next, we call the CGROUP method, which denotes the implementation of the *k-mean*. To accurately measure the cluster from the set of locations, the methods consider the inputs R_0, P_0, pre-fixed cluster centers k, and r. It helps in accurately generating cluster centers for the given set of locations. We store all cluster centers in the *Ccenters*, which serves as input for the CPARTITION method.
3. In Step 6, the CPARTITION method (based on *Voronoi* algorithm) is used to calculate the nearest neighbors by generating sub-regions with their seed points. The CPARTITION method utilizes the generated cluster centers (*Ccenters*) and region R_0 to construct optimal sub-regions (stored in *Gregion*) and select the optimal places of the fog node depending on *Ccenters*, which are then stored in *Gseed* as seed points. This step ensures that the observed fog node locations are optimal while in proximity to the OPWAPs and the generated sub-region.
4. Finally, in Step 7, the seed points generated by the *Voronoi* algorithm is used as the fog node placement locations.

In terms of time complexity, the CGROUP method (*k-means*), which operates in $\mathcal{O}(P_0kr)$ time, demonstrates higher efficiency compared to alternative computations. The CPARTITION method, used to construct Voronoi algorithm, operates in $\mathcal{O}\left(P_0\log_2 P_0\right)$ time. To sort maximum $2P_0 - 5$ Voronoi vertices requires $\mathcal{O}\left(P_0\log_2 P_0\right)$ time. Accordingly, the proposed overall heuristic complexity for the R_0 regions is $\mathcal{O}\left(R_0P_0kr + R_0P_0\log_2 P_0\right)$. This complexity has deeper implications, especially concerning the factors R_0 and P_0. Thus, it needs further discussion, presented below.

1. *Case* $P_0 \ll kr + \log P_0$: In situations where the number of potential OPWAP locations (P_0) is significantly smaller than the product of clusters (kr) and the logarithm of P_0, our heuristic exhibits high computational efficiency. This scenario often aligns with real-world deployments, making our approach particularly effective when the number of regions (R_0) is substantial.
2. *Inverse Case*: $P_0 \gg kr + \log P_0$: Conversely, when the number of potential OPWAP locations (P_0) is larger, our heuristic maintains its efficiency due to the linear relationship with R_0. This adaptability is crucial for scalability, ensuring the practical applicability of our algorithm across various scales of OPWAP distributions within a given region. The presented heuristic's time complexity of $\mathcal{O}\left(R_0P_0kr + R_0P_0\log_2 P_0\right)$ reflects a balance between computational efficiency and accuracy in fog node placement.

Fog node selection algorithms

Greedy performance-based node selection

We describe our first proposed algorithm for fog node selection as greedy performance-based node selection (GPDS), based on approximation algorithms related to set-covering problems [40]. Our focus is on the user rather than the evaluation of fog node sites. The algorithm assigns users to sites with the shortest link delay and selects the closest fog-node site if necessary. This is done to avoid repeating the different site selections for a user. Note that F_j^{Cost} is added to the cost only when the new fog node is selected. The above procedures are shown in Algorithm 2 from lines 1–9. In this, the first step initializes the set of fog nodes in a region G_j, where a user u_j can be assigned, and the second step sorts the distances in ascending order. From steps 3–9, for every user, the algorithm calculates the cost for each fog node in a region, selects the one with the minimum cost, assigns the user u_j to the fog node G_{j*} with the minimum cost, and ensures that if fog node is not already chosen, it should get selected.

In summary, the algorithm initializes fog nodes, sorts distances based on the number of fog nodes, assigns users to fog nodes with minimum cost, and ensures fog node selection, all of which are executed in sequence for each user in the region.

Algorithm 2 GPDS: Greedy Performance-based Node Selection

$G_j \subseteq G_j^r$, set of fog nodes in a region r, where user u_j can be assigned.
Sort d_{ul_j} in ascending sequence of $|G_j|$.
for all u_j do
    $j^* \leftarrow \underset{j \in \{j | n_j \in G_j\}}{argmin} F_j^{Cost} + w_j C_{l_j}$
    Assign user u_j to G_{j*}
    if G_{j*} is not selected **then**
        select G_{j*}
    end if
end for

Greedy fog node selection algorithm

The second algorithm we propose for fog node selection is called GFNSA (Greedy fog node selection algorithm). This is because we incrementally decide to select a fog node site in a region with the minimum placement cost and maximum service utility. We allocate all possible users to this site. To understand the performance of the fog node site, we have described the site service utility as the ratio of the total requested volume (requested in bytes) to the cost of serving those requests. Possible fog node site users are those within the region range specified by the delay parameter for the CDN services of this site but not yet assigned. We first select the fog node and then search for the next best fog node until all users are assigned to a fog node site.

The above procedures are shown in Algorithm 3 from lines 1–9. In the first two steps, we initialize the algorithm by defining the set of users and the current set of users that can be assigned to fog nodes in a region. From lines 3–9, we calculate the total requested volume W_j, select the fog node that maximizes the expression (Step 5–Service utility), assign all the users to the selected fog node, and update the user set. We repeat the above steps until all the users are allocated. The above both greedy algorithms have a factor approximation ratio of $\mathcal{O}(\log n)$ for the minimum set cover problem, where n is the size of the universe of elements to be covered [41]. Consequently, algorithms' overall complexity is $\mathcal{O}(\log \mathcal{M})$.

Algorithm 3 GFNSA: Greedy Fog Node Selection Algorithm

U is the set of users that needs to be allocated.
Set U_j represents the current user set, can be assigned to $G_j \in G_j^r$ in a region r.
while $U \neq 0$ *do*
    $W_j \leftarrow \sum_k w_k, \quad u_k \in U_j$
    $j^* \leftarrow \underset{j \in \{j | G_j \text{ is available}\}}{argmax} \frac{W_j}{W_j C_{l_j} + F_j^{Cost}}$
    Assign all the users in U_{j*} *to* G_{j*}
    Select G_{j*}
    $U \leftarrow U \backslash U_j$
end while

Discussion on the proposed fog node selection algorithms

The proposed algorithms are designed to work offline, which means they provide effective solutions based on the given input. The redirection process of user requests to appropriate fog nodes can be achieved using existing CDN technologies, which are an integral part of CDN. Existing techniques namely URL rewriting, transparently intercepted user's requests, DNS-based request redirection, or URL rewriting can be used for this purpose [42]. We assume that the proposed fog node selection algorithms interact with the origin server system (CCP node) to facilitate the redirection of user requests to appropriate fog nodes.

Performance evaluation

The following section demonstrates the method used to assess the model. First, referring to the experimental evaluation in [27, 28, 35, 43], we provide two methods for comparative studies. Next, we present the simulation parameters used in the model. Finally, we discuss the results and analyze their strengths and limitations.

Compared methods

1. *Greedy Request-based Approach (GRA):* The algorithm adopted from [35, 43], operates by selecting the lowest cost fog node site to serve the user's initial request, which depends on the total cost to serve that request. Subsequent requests are then directed to the assigned fog node site. This approach is similar to the*GPDS* algorithm but differs in the user evaluation order, i.e., users are evaluated based on the order of their initial request arrivals. Additionally, considerations for content replication and user redirection only involve the volume of each user's first request.

Algorithm 4 GRA: Greedy Request-based Approach

```
for all request q (ascending arrival order) do
    q is requested by u_j, which is of size w_j
    if u_j is served by n_j then
        Assign q to n_j
    else
        G_j ⊆ G'_j ; set of nodes where u_j may be allocated.
        j* ← argmin_{j∈{j|n_j∈G_j}} F_j^Cost + w_j C_lj
        Assign q to n_j^*
    end if
    if G_j* is not choosen then
        choose G_j*
    end if
end for
```

2. *Randomized Request-based Approach (RRA):* In this, we randomly select a fog node for each incoming user request, but without evaluating the cost for each potential fog node site. Instead, a random fog node site would be selected for each request, with subsequent demands by the same user redirec towards the allocated fog node.

Simulation setup

For the simulation, we have considered a hierarchical network model shown in Fig. 2. This includes a CCP node that serves content to various fog nodes deployed at various locations in various regions. Finally, end-users are connected to the corresponding region and fog nodes through various OPWAPs.

Table 3 Examples of the fog node

Type of fog nodes	Router-based computing node	Micro-server
Storage (TB)	1 — 8	12 — 14
Computing power	1 — 2 GHz	2 — 3 GHz
Cost ($)	250 — 450	700 — 1800

Table 4 Few examples of typical prices charged by cloud CDN providers

Cost type	Azure	Amazon	Google
Egress data transfer ($/GB)	0.081	0.085	0.080
Ingress data transfer ($/GB)	0.01	0.01	0.01
Storage ($/GB)	0.15	0.23	0.20

Table 5 Simulation parameters summary

Parameters	Default	Range
Delay (ms)	50	10 – 500
Number of regions	20	5 – 20
Fog nodes per region	40	10 – 100
Replica size	200 MB	2 MB – 10 GB
Update frequency (in %)	20	10 – 20

To define the various costs of the fog nodes, we use the cost ranges defined in Table 3, randomly select a cost, and attach it to the corresponding fog node. Although we have selected some examples of fog nodes for simulation, it can be further categorized to deploy various varieties of fog nodes', e.g., mini or micro data center [44]. Finally, we map a cloud content provider, regions, fog nodes, and users in a single metric field for evaluation.

We have estimated the cost of content distribution (transmisson and storage cost) referring to the usual values charged by different CDN-providers like *Azure, Amazon, and Google*, summarized in Table 4. We randomly select the costs defined in the table and assign the cost to the corresponding links according to the definition detailed in "System model and problem formulation" section. Furthermore, link delays are uniformly distributed and randomly assigned to the links according to the values defined in Table 5.

We have considered a range of content replica sizes, that is, 2 MB to 10 GB with 200 MB size as the default replica size [35]. The reason behind choosing the replica size range is that much of the traffic is generated by video content, and these video content, specifically streaming content, are divided into small chunks, which can be as low as the size of 2 MB [45]. A high-quality video replica can be as large as 10 GB. A wide range of replica sizes allows the model to be accurately measured.

The representation of delay between cloud and end-users, spanning various regions and fog nodes, involves utilizing the overall average delay of the connecting links, denoted

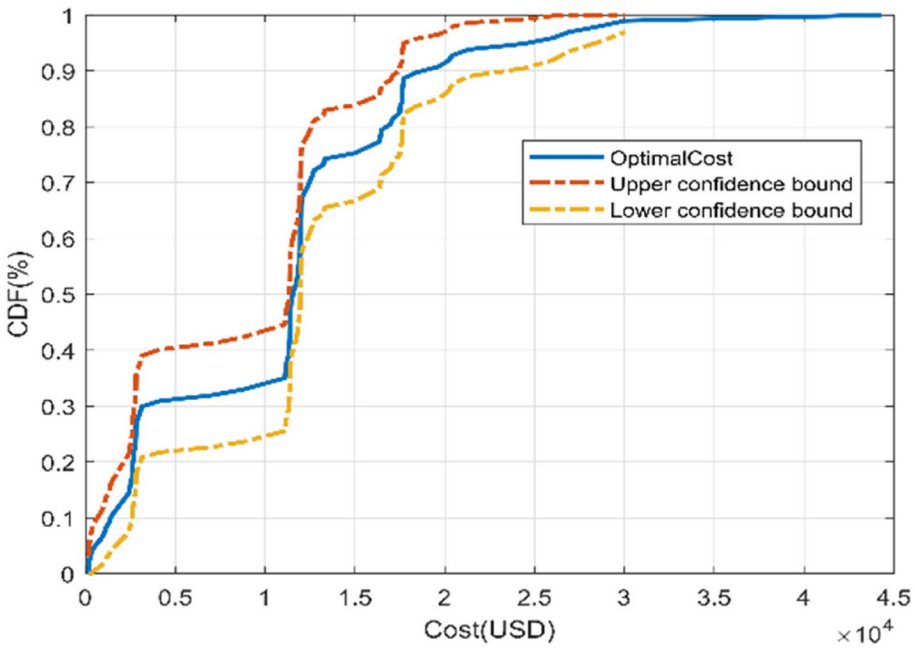


Fig. 3 CDF of the optimal cost in all the tests

as "Delay." We chose a default setting of 50 ms (ms) with a range of delays between 10–500 ms. The chosen default aligns with the maximum service delay prerequisites for diverse services. Delays within the range of 10 to 50 ms are deemed suitable for real-time applications, while delays up to 500 ms are permissible for non-real-time services such as file downloads. These delay values are essential to provide guaranteed QoS for end-users and support for various services, as discussed in "Introduction" section and "System model and problem formulation" section of the paper.

In our simulation, the region represents an area where various fog nodes can be placed. According to the CDN provider requirements, these regions can be tuned to facilitate multiple services to end users. For example, a CDN provider can choose a rectangular area of service, such as an area of $17 \times 25\ km^2$, to provide services [46]. For the fog node placement locations, we have analyzed data set [47] to show the geo-distribution of OPWAPs over a rectangle region. This data set contains various OPWAP locations in a region with their unique information, such as geographical coordinates, boroughs, and unique SSIDs. An inherent limitation of the datasets is their dynamic nature, implying that the incorporation of recently identified OPWAP locations is not assured.

Thus, to show our model and algorithm's performance, we fixed the maximum regions to twenty, each having a maximum number of hundred fog nodes placed. This suffices to provide various understandings of the model. These fog node sites have various storage ranges, costs, and sufficient computing capabilities, which is helpful in accurately evaluating the model. We have set the update frequency for contents to be 20% of storage size. The literature indicates that this can range from 10%–20% for the optimal performance of the content distribution [48, 49]. Simulations were carried out on a system utilizing an Intel Core i5-11300H CPU and 10 GB of memory. The implementation of all proposed methodologies was achieved through the utilization of *CPLEX API*, *MATLAB 2020b* API, and *Python* (*ver.* 3.10). We summarize the values in Tables 3, 4 and 5.

Evaluation of the model

As defined in the previous subsection, we add all values in the same metric field for evaluation. We use parameters defined in Table 5 to generate various settings. For each of the settings, we generate ten settings by choosing ten random values and fixing the rest four parameters to their default values. These settings correspond to the $10 \times 5 = 50$ settings for the evaluation, and then we map these settings to our metric field by generating various combinations of a CCP node, regions, fog nodes, and users. We randomly generate two combinations for each setting; thus, twenty evaluations for each set correspond to $50 \times 20 = 1000$ settings. On the basis of these evaluations, the model can be calculated accurately. After generating these data files, we used the CPLEX optimizer [37] to provide the various results for the models. These results are plotted against the total cost of the Fog-CDN model.

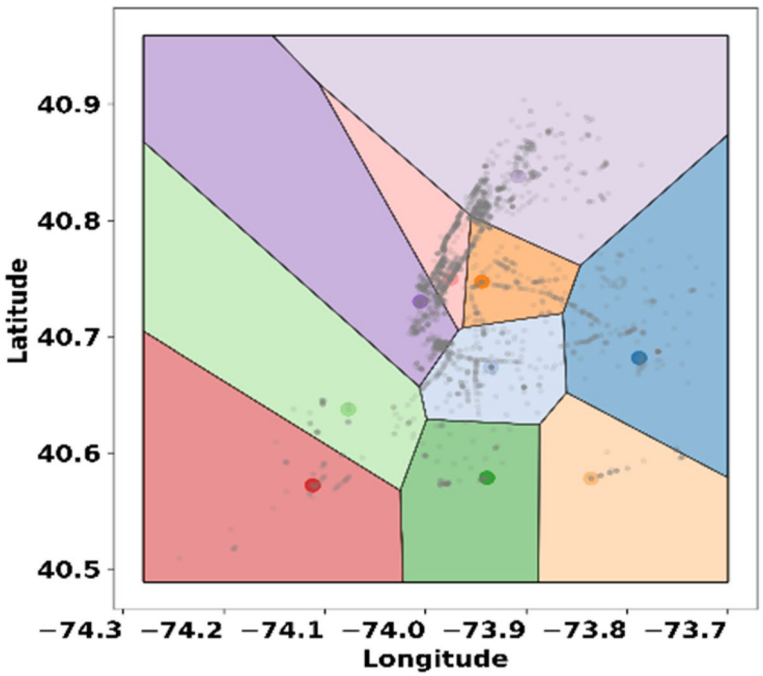
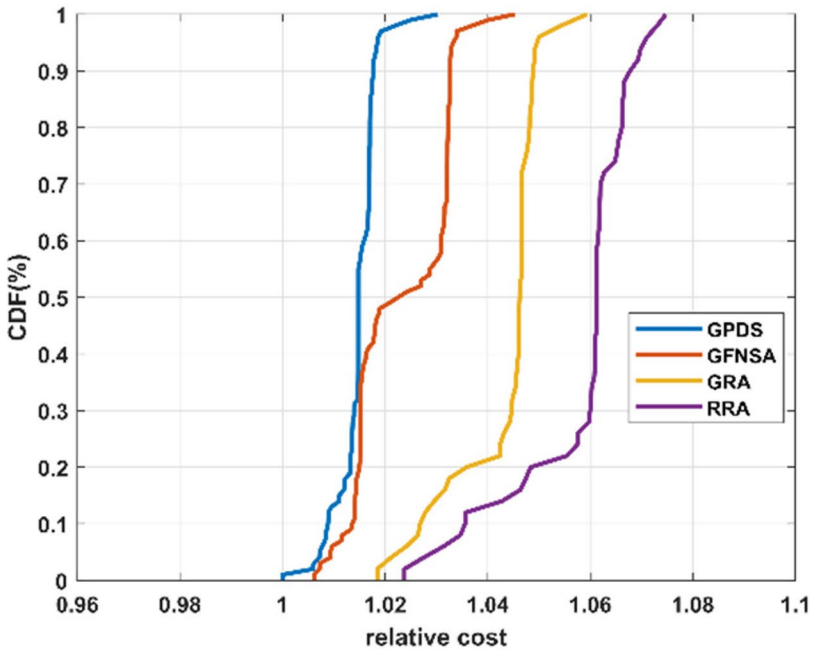
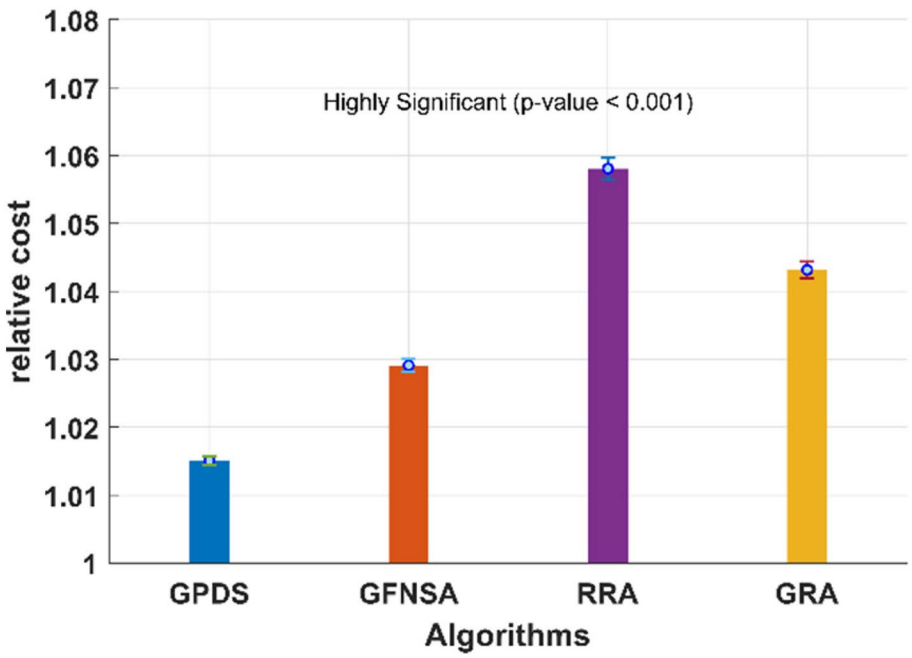


Fig. 4 The effect of Algorithm 1 PFOLR on a region for optimal fog node placements in a sample of geo-distributed OPWAPs

In Fig. 3, we have plotted the cost against all the test results. It shows that almost 90% of the test costs are less than 20,000 USD, which is much less than placing an aggregate data center cost for the same purpose [50]. Thus, content service providers can use our method to build a Fog-CDN structure that can offer various services to end-users while minimizing the total cost. These



(a)



(b)

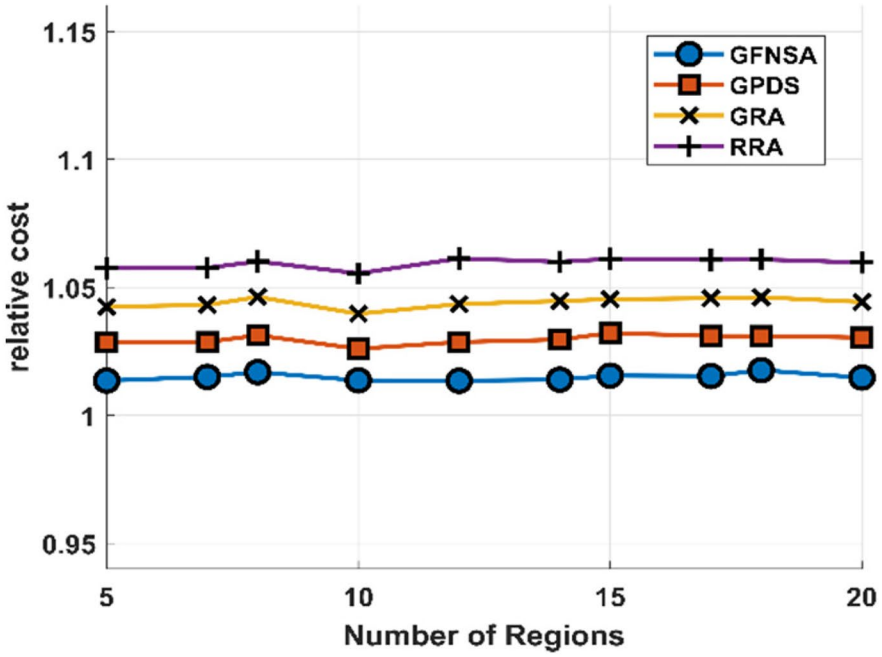
Fig. 5 Performance statistics for fog node selection algorithms

results show the maximum placement of two thousand fog nodes in 20 regions.

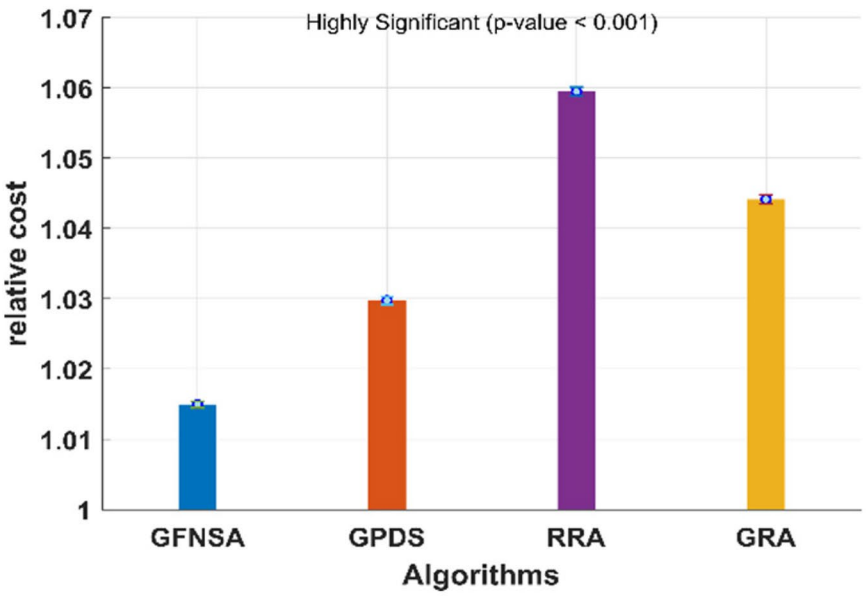
Performance evaluation of the algorithms

We demonstrate the performance of fog node placement and fog node selection algorithms to solve the problem. In the first step, we need to determine the fog node placement locations to build Fog-CDN model.

The proposed Algorithm 1 *PFLOR* results, shown in Fig. 4, which represents a region with various optimal *Voronoi* sub-regions, each with an ideal placement location for the fog node. Each colored dot represents the optimized fog node placement generated by the



(a)



(b)

Fig. 6 Algorithms simulations results for parameter number of regions on the FogCDN model. **a** Relative cost achieved by the algorithms in various settings. **b** Performance error bars

algorithm, closer to a sub-region OPWAPs locations (represented as grayed-out points). The proposed fog node placement Algorithm 1, *PFLOR*, generates ten optimal Voronoi sub-regions according to the dataset's OPWAP locations.

For evaluations of greedy-based fog node selection algorithms, we report the relative cost, defined as the ratio of total cost over optimal cost to evaluate our algorithms. In Fig. 5, we have shown the performance statistics of all algorithms by reporting the CDF of all the relative costs achieved by the algorithms for the model. Figure 5a demonstrates that the overall performance of the proposed Algorithm 2 GPDS and Algorithm 3 GFNSA compared with the algorithms GRA and RRA. Compared to baseline techniques, both proposed algorithms work well and frequently achieve the relative cost of 1.0327 and 1.0452, respectively. To show the significance of the results, we have presented the error bar plot over the mean relative costs achieved by the algorithms shown in the form of bar plots in Fig. 5b. Additionally, Fig. 5b shows the mean and standard error of the relative costs for the algorithms, and the performance significance outcome by performing the t-test between the two proposed algorithms (GPDS and GFNSA) to determine if both algorithms are significantly different.

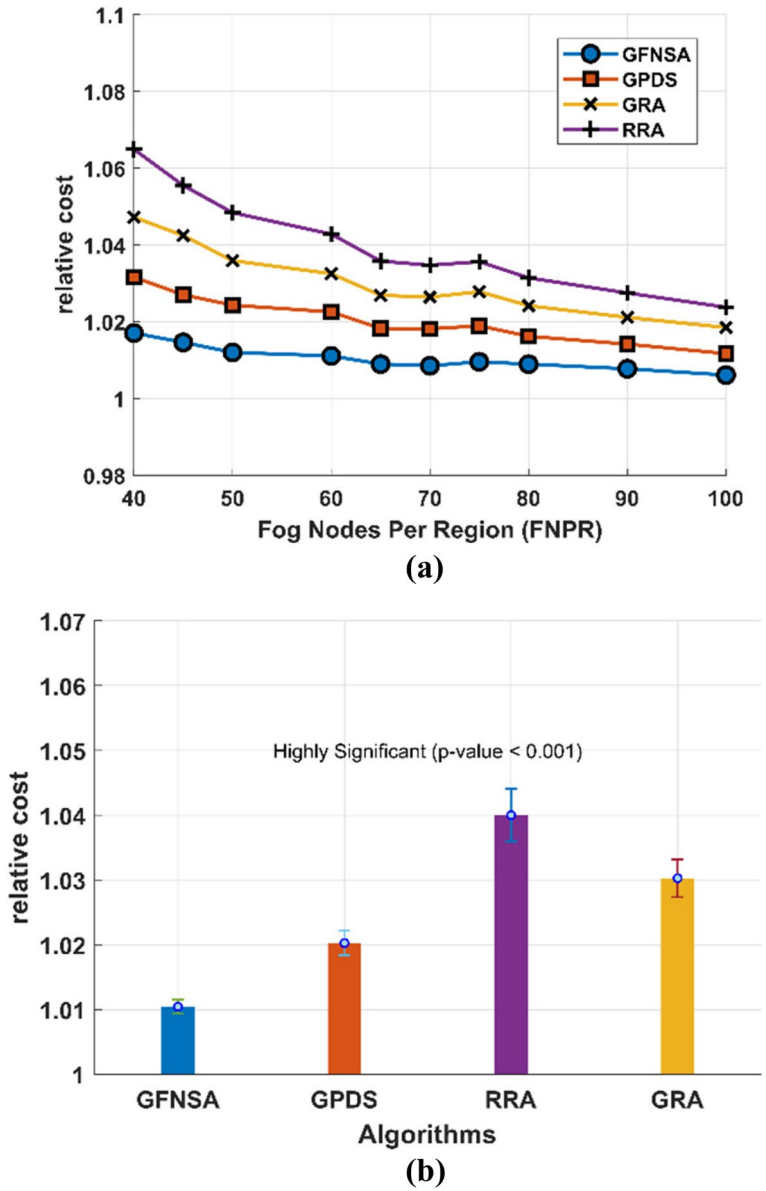


Fig. 7 Algorithms simulations results for parameter FNPR on the Fog-CDN model. **a** Relative cost achieved by the algorithms in various settings. **b** Performance error bars

The mean of all relative costs for GPDS is 1.0151, and the mean relative costs for GFNSA is 1.0291. The t-test on the two sets of data resulted in a p-value of less than 0.001. This indicates that the difference between the mean relative costs of the two proposed algorithms is statistically significant. Thus, based on the observed results, we can say that both proposed algorithms are statistically different and perform well compared to existing methods. Furthermore, we suggest Algorithm GPDS as the preferred algorithm for fog node selection due to its lower mean value. Therefore, we recommend using Algorithm GPDS to achieve a lower cost for the model.

Next, we evaluate the performance of the algorithms according to the simulation parameters defined in the model. To better comprehend the results presented in Figs. 7, 8, 9 and 10, first, we must examine the total cost structure.

Let $F_{\mathcal{N}}^{OPT}W$ denote the optimal fog node selection cost and $\Delta F_{\mathcal{N}}^{OPT}W$ represents the additional fog node selection cost incurred by the algorithms. Similarly, let $\sum_k C_k^{OPT} w_k$ denote the optimal user access cost and

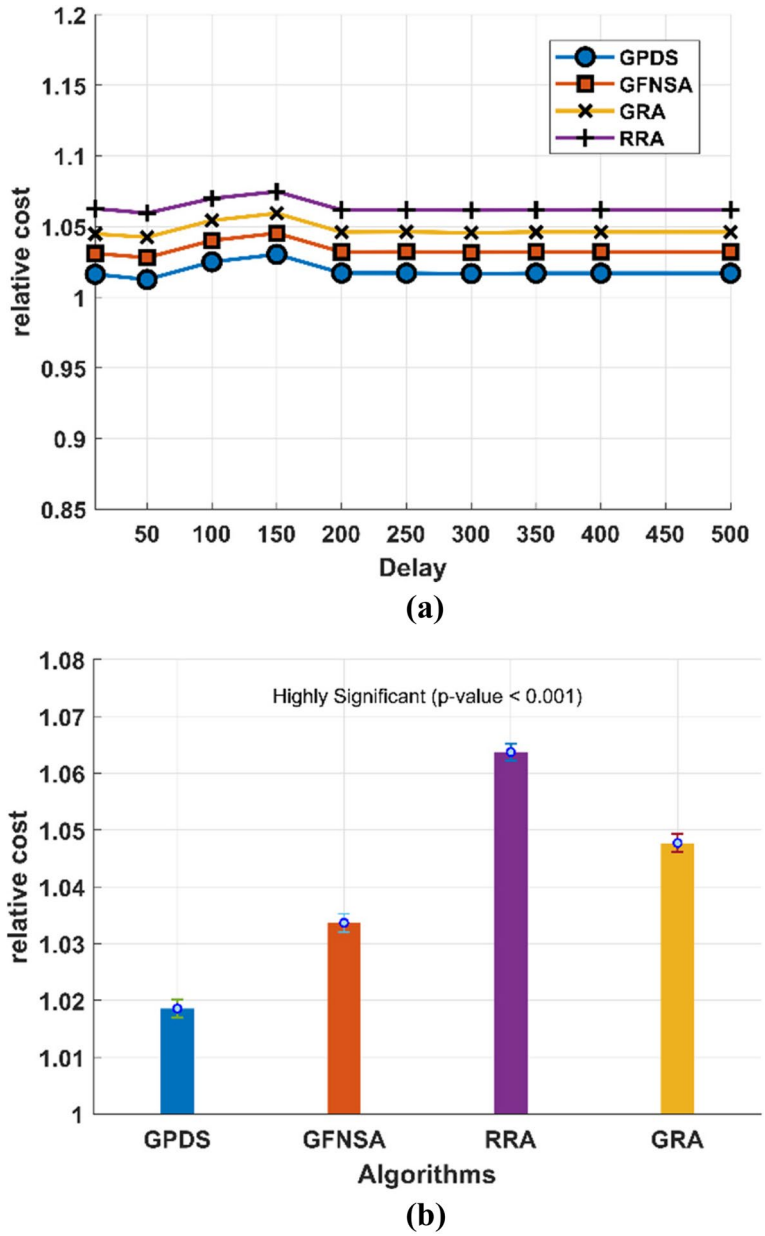


Fig. 8 Algorithms simulations results for parameter delay on the Fog-CDN model. **a** Relative cost observed by algorithms in various settings. **b** Performance error bars

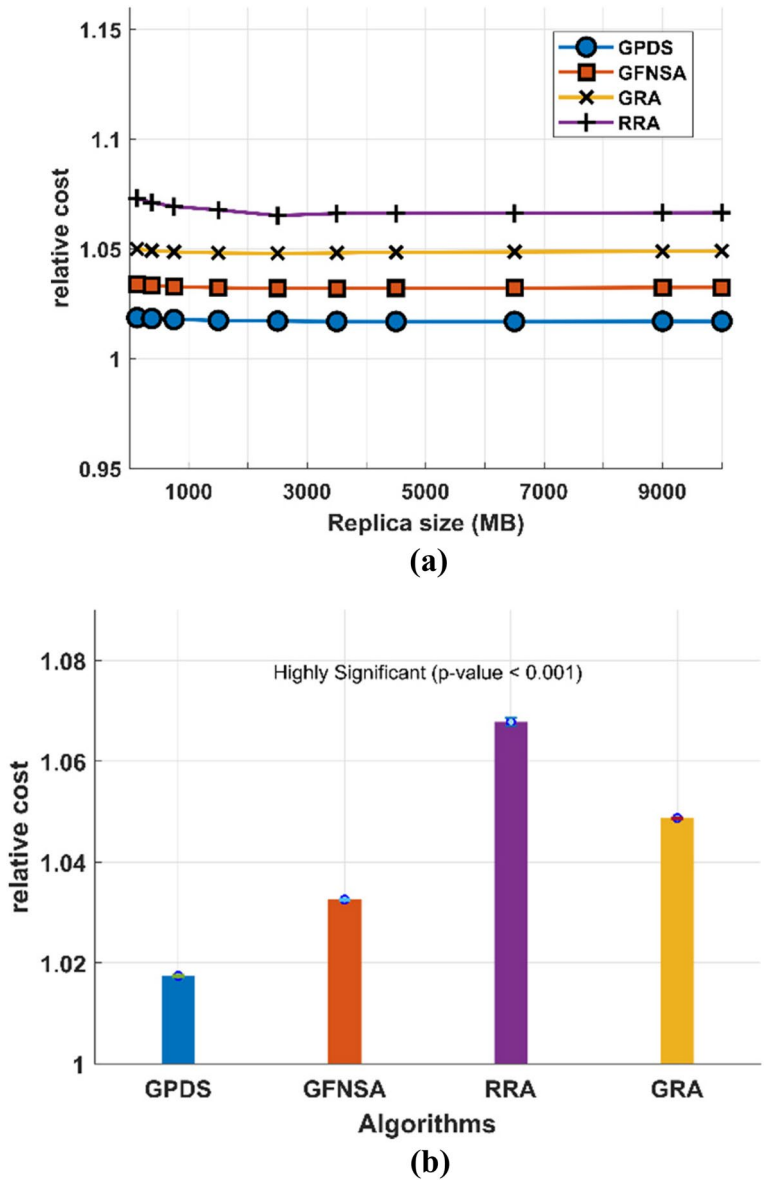


Fig. 9 Algorithms simulations results for parameter replica size on the Fog-CDN model. **a** Relative cost observed by algorithms in various settings. **b** Performance error bars

$\sum_k \Delta C_k w_k$ represent the additional cost of user access. Therefore, the relative cost shown in the figures is:

$$relative\ cost = \frac{(F_{\mathcal{N}}^{OPT} W + \Delta F_{\mathcal{N}}^{OPT} W) + \sum_k (C_k^{OPT} + \Delta C_k) w_k}{F_{\mathcal{N}}^{OPT} W + \sum_k C_k^{OPT} w_k}$$

In one extreme case, when $W \gg \sum_k w_k$,

$$relative\ cost \approx \frac{F_{\mathcal{N}}^{OPT} + \Delta F_{\mathcal{N}}^{OPT}}{F_{\mathcal{N}}^{OPT}} = 1 + \frac{\Delta F_{\mathcal{N}}^{OPT}}{F_{\mathcal{N}}^{OPT}}$$

We can observe only the additional cost for selecting a fog node site. Alternatively, when $W \ll \sum_k w_k$,

$$relative\ cost \approx \frac{\sum_k (C_k^{OPT} + \Delta C_k) w_k}{\sum_k C_k^{OPT} w_k} = 1 + \frac{\sum_k \Delta C_k w_k}{\sum_k C_k^{OPT} w_k}$$

We can only observe the additional cost of user access. In the rest of the cases, the relative sizes of $\sum_k w_k$ and W determine which part of the non-optimality (for either user access or the fog node site selection) is dominating the total relative cost.

We show all observed results related to algorithms GPDS, GFNSA, GRA, and RRA in Figs. 6, 7, 8, 9 and 10. In Figs. 6 and 7, we varied the parameter number of regions and the number of fog nodes placed in each region and observed that as both increase, Algorithm 3

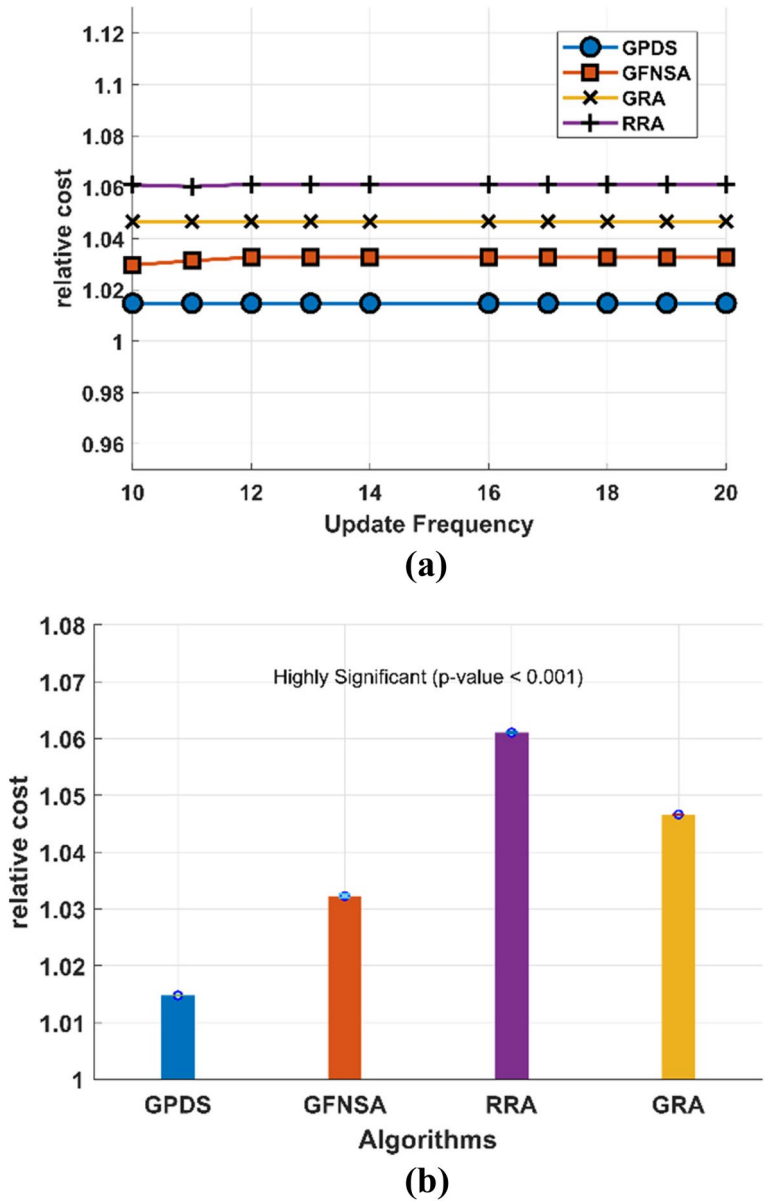


Fig. 10 Algorithms simulations results for parameter update frequency on the Fog-CDN model. **a** Relative cost observed by algorithms in various settings. **b** Performance error bars

GFNSA outperforms Algorithm 2 GPDS. This is because Algorithm GFNSA selects the optimal fog node at the cost of $F_{\mathcal{N}}^{OPT}W$, while Algorithm GPDS incurs an additional cost to select fog nodes, $\Delta F_{\mathcal{N}}^{OPT}W$. Moreover, both algorithms perform at near-optimal levels with increasing FNPR. Figures 6b and 7b present the means and error bars for the statistical performance of all algorithms. From the observed results, we can conclude that Algorithm 3 GFNSA performs better than Algorithm 2 GPDS and outperforms the otherbaseline techniques.

In Fig. 8, we have varied the delay parameter for various delay settings to present the algorithms' performance, as described in "Simulation setup" *subsection*. We found that Algorithm 2 GPDS is able to achieve a lower relative cost while delivering optimal QoS (delay) when compared to others. Figures 9 and 10 shows the performance of parameters content replica size and update frequency compared to *relative cost*. We found that with increasing parameter values of replica size and update frequency, the proposed algorithms

provide near-optimal results. Although compared to Algorithm 3 GFNSA, Algorithm 2 GPDS performs slightly better. Compared to the proposed algorithms, Algorithm 4 GRA processes requests in the order they arrive and assigns users to a site based solely on their first request, unlike GPDS and GFNSA, which aggregate all requests from each user into one. This assignment order results in a higher relative cost due to more sites being opened and users being assigned to sites with higher download costs compared to the other two heuristics. The reason for the poor performance of Algorithm 5 RRA compared to all algorithms is due to the random allocation of the fog nodes site.

In summary, Algorithm 2 GPDS and Algorithm 3 GFNSA provide near-optimal solutions for the various problem instances defined for the model and outperform the baseline techniques. We have presented various results on the simulation parameters to show the cost-effectiveness and scalability of our algorithms, and with the observed results we conclude that both algorithms achieve the objectives (R1)–(R3). In this, Algorithm 2 GPDS performs slightly better compared to Algorithm 3 GFNSA in three of the parameters and Algorithm 3 GFNSA in the other two parameters. Therefore, we suggest Algorithm 2 GPDS as the preferred algorithm for fog node selection.

Conclusion and future work

This article presents a mathematical approach to the challenge of accommodating the growing demand for data and diverse applications by utilizing developing technologies, such as fog-based infrastructure, to assist CDNs. Our research demonstrates how strategically placed fog nodes in a region can be used to build a Fog-CDN model that can collaborate with edge network hotspots and serve requests to users within a few hundred meters. The Fog-CDN model optimizes the placement of fog nodes and the replication path for content distribution, minimizing total costs and ensuring cost-efficient content delivery. The model ensures QoS to end users and can support different data-driven services based on CDN service delay requirements. In addition, we have presented the two heuristics to solve the model, and the results show that these algorithms provide near-optimal solutions to the problem on various network parameters. Beyond methodological complexities, our contributions demonstrate the practical feasibility, and have real-world implications. The study's impact lies in providing efficient solutions to contemporary content delivery challenges using emerging technology (fog-based solutions).

Acknowledging the dynamic nature of fog-based CDN environments, in the future, we will include integrating mobility and dynamicity considerations into the model to further enhance end-user services. Mobility in the system is associated with the movement of users or nodes across different regions. This might impact the optimal placement of fog nodes and the efficiency of content distribution. Recognizing the variability in delays due to factors like congestion, network failures, and path changes, we aim to develop mechanisms that explicitly address these variations. Furthermore, our ongoing work will focus on seamless SLA integration to manage the evolving demands of real-world scenarios.

Acknowledgements
The authors acknowledge the Bharti School of Telecommunication Technology and Management (BSTTM) at IIT Delhi and Bennett University, Greater Noida, India for the provision of facilities crucial to the research.

Authors' contributions
Both authors have equally contributed to the work.

Funding
No funding has been received for this work.

Availability of data and materials
Dataset is available on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable. No human or animal subjects are involved in this study.

Competing interests
The authors declare no competing interests.

Received: 8 September 2023 Accepted: 10 August 2024
Published online: 17 September 2024

References

1. Cisco Annual Internet Report 2018–2023. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11–741490.pdf. Accessed 04 Apr 2022
2. Content delivery reference architecture. https://www.akamai.com/resources/reference-architecture/content-delivery-reference-architecture. Accessed 03 Mar 2022
3. Song Y, Wo T, Yang R, Shen Q, Xu J (2021) Joint optimization of cache placement and request routing in unreliable networks. J Parallel Distrib Comput 157:168–178. https://doi.org/10.1016/j.jpdc.2021.06.006
4. Wen Y, Chen Y, Shao ML, Guo JL, Liu J (2020) An efficient content distribution network architecture using heterogeneous channels. IEEE Access 8:210988–211006. https://doi.org/10.1109/ACCESS.2020.3037164
5. Elazhary H (2019) Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. J Netw Comput Appl 128:105–140
6. Nygren E, Sitaraman RK, Sun J (2010) The Akamai network: a platform for highperformance internet applications. SIGOPS Oper Syst Rev 44(3):2–19
7. Kilanioti I, Fern´andez-Montes A, Fern´andez-Cerero D, Karageorgos A, Mettouris C, Nejkovic V, Albanis N, Bashroush R, Papadopoulos GA (2019) Towards efficient and scalable data-intensive content delivery: state-of-the-art, issues and challenges. In: Kol odziej J, Gonz´alez-V´elez H (eds) High-Perform. Model. Simul. Big Data Appl. LCN CS. Springer, Cham, pp 88–137
8. Wang Z, Huang J, Rose S (2018) Evolution and challenges of DNS-based CDNs. Digit Commun Netw 4(4):235–243. https://doi.org/10.1016/j.dcan.2017.07.005

9. Stocker V, Smaragdakis G, Lehr W, Bauer S (2017) The growing complexity of content delivery networks: challenges and implications for the Internet ecosystem. Telecommun Policy 41(10):1003–1016. https://doi.org/10.1016/j.telpol.2017.02.004
10. Zheng Z, Zhao C, Zhang J (2021) Time-sensitive overlay routing via segment routing with failure correction. IEEE (ICC Workshops). pp 1–6. https://doi.org/10.1109/ICCWorkshops50388.2021.9473766
11. Alharbi HA, Aldossary M (2021) Energy-efficient edge-fog-cloud architecture for iotbased smart agriculture environment. IEEE Access 9:110480–110492. https://doi.org/10.1109/ACCESS.2021.3101397
12. Yu W, Liang F, He X, Hatcher WG, Lu C, Lin J, Yang X (2018) A survey on the edge computing for the internet of things. IEEE Access 6:6900–6919. https://doi.org/10.1109/ACCESS.2017.2778504
13. Abdali T-AN, Hassan R, Aman AHM, Nguyen QN (2021) Fog computing advancement: concept, architecture, applications, advantages, and open issues. IEEE Access 9:75961–75980. https://doi.org/10.1109/ACCESS.2021.3081770
14. Li B, Shi Y, Yuan Y (2022) Suitability-based Edge Server Placement Strategy in 5G Ultra-dense Networks. Proc. IEEE CSCWD. pp 1108–1113. https://doi.org/10.1109/CSCWD54268.2022.9776038
15. Yadav P, Kar S (2024) Efficient content distribution in fog-based CDN: a joint optimization algorithm for fog-node placement and content delivery. IEEE Internet Things J 11(9):16578–16590. https://doi.org/10.1109/JIOT.2024.3355468
16. Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA (2018) A comprehensive survey on fog computing: state-of-the-art and research challenges. IEEE Commun Surv Tut 20(1):416–464. https://doi.org/10.1109/COMST.2017.2771153
17. Firouzi F, Jiang S, Chakrabarty K, Farahani B, Daneshmand M, Song J, Mankodiya K (2023) Fusion of IoT, AI, edge–fog–cloud, and blockchain: challenges, solutions, and a case study in healthcare and medicine. IEEE Internet Things J 10(5):3686–3705. https://doi.org/10.1109/JIOT.2022.3191881
18. Mansouri Y, Babar MA (2021) A review of edge computing: features and resource virtualization. J Parallel Distrib Comput 150:155–183. https://doi.org/10.1016/j.jpdc.2020.12.015
19. Aleisa MA, Abuhussein A, Sheldon FT (2020) Access control in fog computing: challenges and research agenda. IEEE Access 8:83986–83999. https://doi.org/10.1109/ACCESS.2020.2992460
20. Sarkar S, Chatterjee S, Misra S (2018) Assessment of the suitability of fog computing in the context of internet of things. IEEE Trans Cloud Comput 6(1):46–59. https://doi.org/10.1109/TCC.2015.2485206
21. Silva RA, Fonseca NL (2019) On the location of fog nodes in fog-cloud infrastructures. Sensors 19(11):2445. https://doi.org/10.3390/s19112445
22. Wang J, Li D, Hu Y (2021) Fog nodes deployment based on space-time characteristics in smart factory. IEEE Trans Industr Inform 17(5):3534–3543. https://doi.org/10.1109/TII.2020.2999310
23. Ibrahim AH, Fayed ZT, Faheem HM (2021) Fog-based CDN framework for minimizing latency of web services using fog-based HTTP browser. Future Internet 13(12):320–335. https://doi.org/10.3390/fi13120320
24. Alghamdi F, Mahfoudh S, Barnawi A (2019) A novel fog computing based architecture to improve the performance in content delivery networks. Wireless Commun Mobile Comput 2019:78–84. https://doi.org/10.1155/2019/7864094
25. Ghalehtaki RA, Kianpisheh S, Glitho R (2019) A bee colony-based algorithm for micro-cache placement close to end users in fog-based content delivery networks. Proc. IEEE CCNC, Las Vegas, NV, USA, pp 1–4. https://doi.org/10.1109/CCNC.2019.8651773
26. Brogi A, Forti S, Ibrahim A (2018) Deploying fog applications: how much does it cost, by the way? Proc. 8th Int. Conf. Cloud Comput. Serv. Sci., Madeira. pp 68–77. https://doi.org/10.5220/0006676100680077
27. Zhang X, Li Z, Lai C, Zhang J (2022) Joint edge server placement and service placement in mobile-edge computing. IEEE Internet Things J 9(13):11261–11274. https://doi.org/10.1109/JIOT.2021.3125957
28. Lähderanta T, Leppänen T, Ruha L, Lovén L, Harjula E, Ylianttila M, Riekki J, Sillanpää MJ (2021) Edge computing server placement with capacitated location allocation. J Parallel Distrib Comput 153:130–149. https://doi.org/10.1016/j.jpdc.2021.03.007
29. Li B, Hou P, Wu H, Hou F (2021) Optimal edge server deployment and allocation strategy in 5G ultra-dense networking environments. Pervasive Mobile Comput 72:101312–101317. https://doi.org/10.1016/j.pmcj.2020.101312
30. Mohan N, Zavodovski A, Zhou P, Kangasharju J (2018) Anveshak: placing edge servers in the wild. Proc. ACM Mobile Edge Commun, Budapest, pp 7–12. https://doi.org/10.1145/3229556.3229560
31. Drijver FB (2018) Assessment of benefits and drawbacks of ICN for IoT applications. PhD thesis, Delft Univ Technol, Netherlands
32. Padmanabhan VN, Subramanian L (2001) An investigation of geographic mapping techniques for internet hosts. ACM Conf. Appl., Technol., Architecture, Protocols Comput. Commun. SIGCOMM '01. ACM, New York, NY, USA, pp 173–185. https://doi.org/10.1145/383059.383073
33. Pi Y, Jamin S, Danzig P, Qian F (2020) Latency imbalance among internet loadbalanced paths: a cloud-centric view. Proc ACM Meas Anal Comput Syst 4(2):1–29. https://doi.org/10.1145/3392150
34. Xiang C, Wang X, Chen Q, Xue M, Gao Z, Zhu H, Chen C, Fan Q (2019) No-jump-into-latency in China's internet! toward last-mile hop count based ip geolocalization. Int Symp Qual Serv IWQoS '19. ACM, New York, NY, USA
35. Chen F, Guo K, Lin J, La Porta T (2012) Intra-cloud lightning: Building CDNs in the cloud. Proc. IEEE INFOCOM. IEEE, Orlando, FL, USA, pp 433–441. https://doi.org/10.1109/INFCOM.2012.6195782
36. Conn AR, Cornuejols G (1990) A projection method for the uncapacitated facility location problem. Math Program 46(1):273–298. https://doi.org/10.1007/BF01585746
37. ILOGCPLEXOptimizationStudio-Ove view. https://www.ibm.com/products/ilog-cplex-optimization-studio. Accessed 04 Oct 2021
38. Burkard RE, Cela E, Pardalos PM, Pitsoulis LS (1998) The Quadratic Assignment Problem. In: Du DZ, Pardalos PM (eds.) Springer, Boston, MA. pp. 1713–1809. https://doi.org/10.1007/978-1-4613-0303-9-27
39. Garey MR, Johnson DS (1990) Computers and intractability; a guide to the theory of NP-completeness. W. H. Freeman & Co., USA
40. Chvatal V (1979) A greedy heuristic for the set-covering problem. Math Oper Res 4(3):233–235. https://doi.org/10.1287/moor.4.3.233
41. Williamson DP, Shmoys DB (2011) The design of approximation algorithms, 1st edn. Cambridge Univ Press. https://doi.org/10.1017/CBO9780511921735
42. Held G (2010) A practical guide to content delivery networks, 2nd edn. CRC Press, Boca Raton. https://doi.org/10.1201/EBK1439835883
43. Chen F (2012) Resource allocation in information-centric networks. PhD thesis, Pennsylvania State University, USA
44. Micro-datacenter. https://download.schneider-electric.com/files?pDocRef=SPDMDCSpecsS-SeriesEN. Accessed 08 Oct 2021
45. Pantos R, May W (2017) HTTP Live Streaming. Request for Comments RFC 8216, IETF. https://doi.org/10.17487/RFC8216
46. Ma G, Wang Z, Zhang M, Ye J, Chen M, Zhu W (2017) Understanding performance of edge content caching for mobile video streaming. IEEE J Sel Areas Commun 35(5):1076–1089. https://doi.org/10.1109/JSAC.2017.2680958
47. NYC Open Data. https://data.cityofnewyork.us/browse?category=Environment&page=2. Accessed 24 Feb 2022
48. Carbunar B, Potharaju R, Pearce M, Vasudevan V, Needham M (2013) A framework for network aware caching for video on demand systems. ACM Trans Multimedia Comput Commun Appl 9(4):1–22
49. Yadav P, Kar S (2020) Evaluating the impact of region based content popularity of videos on the cost of cdn deployment. 26th Nat Conf Commun (NCC). pp 1–6. https://doi.org/10.1109/NCC48643.2020.9056021
50. Bouten N, Famaey J, Mijumbi R, Naudts B, Serrat J, Latré S, De Turck F (2015) Towards NFV-based multimedia delivery. In: 2015 IFIP/IEEE Int Symp Integr Netw Manage. (IM). pp 738–741

Publisher's Note

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com