Information-Centric Networks: A Convex Joint Cache Allocation and Router Assignment

Parisa Eslami, Pejman Ghasemzadeh, *Member, IEEE,* and Shahriar Shahabuddin, *Senior Member, IEEE*

***Abstract*—Information-Centric Networking (ICN) is an emerging communication paradigm that addresses the limitations of the current Internet architecture by shifting the focus from host-based communication to content-based delivery. A key feature of ICN is in-network caching, which reduces redundant transmissions, access latency and improves scalability. In this paper, we investigate the joint problem of cache allocation and request routing in ICN, where caching and routing decisions are inherently interdependent. We formulate the total network cost as a function of both cache placement and content delivery paths, subject to cache capacity constraints and service feasibility requirements. The resulting optimization problem is non-convex. In order to solve it, we employ inner approximation techniques based on the Difference-of-Convex (DC) programming framework combined with convex relaxation and penalty methods. This results in an iterative algorithm that jointly optimizes cache configuration and routing assignments while ensuring convergence to near-binary feasible solutions. Extensive simulations, including Monte Carlo experiments over randomized topologies, demonstrate that the proposed framework significantly reduces transportation cost, converges within a small number of iterations, and scales efficiently with network size. Comparisons with baseline strategies such as Cache Everything Everywhere (CEE) and Probabilistic Caching (Probp) show that our method achieves near-optimal performance without incurring excessive redundancy, which further validates both the practicality and efficiency of the proposed optimization approach.**

***Index Terms*—Information-Centric Networking (ICN), in-network cache allocation, inner approximation, convex optimization, difference-of-convex programming.**

I. Introduction

In recent decades, content-oriented platforms such as YouTube, Facebook, and Instagram have transformed Internet traffic from simple text exchanges to complex multimedia activities including large-scale video sharing and live streaming. While these advancements have enhanced user experiences, they have also significantly increased infrastructure costs for network operators. At the same time, this shift has exposed fundamental limitations in the current Internet architecture, particularly in terms of scalability, adaptability, and efficiency. To address these shortcomings, Information-Centric Networking (ICN) has emerged as a promising paradigm for future Internet architectures [1], [2]. A defining feature of ICN is its direct naming of content objects, decoupled from physical locations, which enables efficient and flexible content distribution [3], [4]. For example, in an ICN, a YouTube video can be retrieved from the nearest network node rather than the actual host server, which thereby reduces latency and enhancing efficiency. This naming paradigm is complemented by in-network caching, where intermediate nodes temporarily store content and serve it to subsequent users [5], [6]. In-network caching has thus become a central research focus in ICN due to its potential to reduce latency, optimize bandwidth, and improve scalability [7].

Several caching strategies have been proposed in the literature. For example, [8] categorizes caching methods across Internet-of-Things (IoT), Mobile Ad Hoc Network (MANET), and Vehicular Ad hoc NETwork (VANET) environments, linking them to content popularity, network context, and node characteristics. Hierarchical schemes leverage multi-tier memory architectures (e.g., DRAM for speed, SSD for capacity) combined with probabilistic admission and replacement policies to balance performance and mitigate SSD wear-out. [9]. In mobile ICN, context-aware caching has been proposed to cluster nodes based on mobility patterns and shared content interests, improving local availability under dynamic conditions [11]. Probabilistic caching methods, such as Centrality-Based Caching (CBC) [7] and Probcache [12], diversify cache placement by considering node centrality or embedding metadata fields, e.g., Time Since Inception, Time Since Birth, to estimate caching potential along a path [25]. More recently, socially-aware schemes such as SARAC4N [14] integrate social centrality metrics (degree, betweenness, closeness) with real-time node resource profiles, dynamically adapting cache placement to reduce redundancy and improve scalability.

Despite these advancements, existing approaches often offer limited consideration to real-world networking conditions such as dynamic traffic patterns, constrained cache capacity, and topology variability. Many rely on static or uniform cost functions that fail to adapt to diverse operational contexts. Adaptive caching frameworks, e.g., [15], have improved cache hit ratios by combining content popularity with centrality metrics, but they still assume relatively static conditions and do not fully address practical challenges of joint decision making. This paper addresses these gaps by formulating a joint cache allocation and router assignment problem that dynamically determines content placement within ICN routers.

Our approach captures the natural interdependence between caching and routing within a unified optimization framework, in contrast to methods that treat these components independently. To address the inherent non-convexity of this

P. Eslami is a Ph.D. student with the Department of Information Systems, University of Maryland, Baltimore County, USA (email: peslami1@umbc.edu).

P. Ghasemzadeh and S. Shahabuddin are Assitant Professors with the School of Electrical and Computer Engineering, Oklahoma State University, USA (emails: pejman.ghasemzadeh@okstate.edu and shahriar.shahabuddin@okstate.edu).

formulation, we apply inner approximation techniques based on Difference-of-Convex (DC) programming and convex relaxation, enabling efficient iterative solutions with provable convergence properties. Additionally, we introduce a novel DC-based coupling reformulation tailored specifically for the ICN cache–routing interaction, resulting in a jointly optimized surrogate that is not available in prior literature. The proposed framework is further validated through simulations and comparisons with existing baseline methods.

The remainder of this paper is organized as follows. Section II presents the system model and problem formulation. Section III formulates the joint cache management and router assignment optimization problem. Section IV develops convex approximations using inner approximation and analyzes their complexity and convergence. Section V evaluates the framework through simulations, including Monte Carlo experiments and baseline comparisons. Finally, Section VI concludes the paper and outlines potential future research directions. Table I summarizes the system model parameters notation.

II. System Model and Problem Formulation

We model an ICN with a random topology, represented as a graph $G = (V, E)$ where V denotes the set of network nodes and E represents the set of communication links. The nodes in V are categorized into three types: server nodes, router nodes, and user nodes. We analyze the network under two different scenarios. In the first scenario, representing a *user-generated-content* environment, every user node is capable of generating new content and injecting it directly into the network. In the second scenario, only server nodes generate content, while user nodes are restricted to sending requests and receiving the corresponding content.

Each router node is equipped with a cache memory of C_r capacity (measured in chunks). For simplicity, we assume a homogeneous network in which all routers have the same cache size. The framework naturally generalizes to heterogeneous cache sizes because the cache-capacity constraint simply becomes $\sum_m s_m x_{m,r} \leq C_r$ for each router r, allowing the optimization to operate independently with router-specific capacities without altering the structure of the DC formulation. The network hosts M content items, and each content item $m \in \{1, \ldots, M\}$ has a size denoted by s_m (in chunks). User requests are modeled through request probabilities where $p_{k,m}$ represents the probability that user k requests content m. Accordingly, the request vector for user k is defined as $req_k = \{req_k^1, req_k^2, \ldots, req_k^M\}$. Two types of decisions must be made in this system: cache allocation and router assignment. The *cache allocation strategy* determines whether a given content item should be cached at a particular router. The binary variable $x_{m,r}$ is defined such that $x_{m,r} = 1$ if content m is cached at router r, and $x_{m,r} = 0$ otherwise. The *router assignment strategy* decides which router should serve each user for every content request. The binary variable $y_{k,m,r}$ is defined such that $y_{k,m,r} = 1$ if user k is served by router r for content m, and $y_{k,m,r} = 0$ otherwise. Naturally, $y_{k,m,r}$ can only be positive if $x_{m,r} = 1$, i.e., the router must cache the requested content.

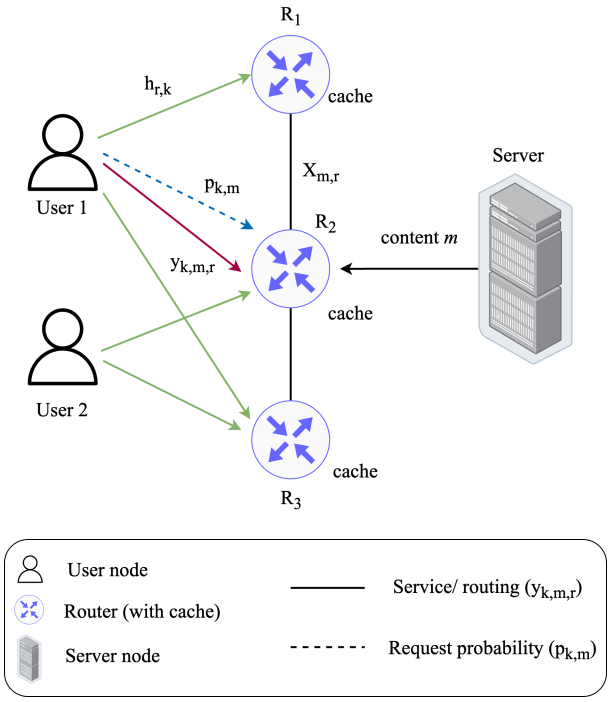

Fig. 1. ICN network model with two users, three cache-enabled routers, and one server, showing the request probability $p_{k,m}$, routing assignment $y_{k,m,r}$, cache placement $x_{m,r}$, and hop distance $h_{r,k}$

TABLE I
System Parameters

Notation	Description
$G = (V, E)$	Graph representing the ICN topology
V	Set of network nodes (server, router, and user nodes)
E	Set of communication links interconnecting nodes
C_r	Cache capacity of router r (in chunks)
M	Total number of content items in the network
s_m	Size of content item m (in chunks)
$p_{k,m}$	Probability that user k requests content m
req_k	Request vector of user k across all content items
$h_{r,k}$	Hop distance between router r and user k
$x_{m,r}$	Binary variable: 1 if content m is cached at router r, 0 otherwise
$y_{k,m,r}$	Binary variable: 1 if router r serves user k for content m, 0 otherwise

The operational cost in the network depends on three main factors: (i) the hop distance $h_{r,k}$ between a user k and the router r serving the request, (ii) the request probability $p_{k,m}$, which captures the popularity of content items and (iii) the content size s_m that affects both storage capacity and transmission cost. Based on these considerations, the network

cost function is defined as:

$$f_{x_{m,r},y_{k,m,r}} = \sum_{k\in K}\sum_{m\in M}\sum_{r\in R} h_{r,k}\cdot p_{k,m}\cdot s_m \cdot x_{m,r}\cdot y_{k,m,r}. \quad (1)$$

The optimization problem can then be formulated as:

$$\min_{x_{m,r},y_{k,m,r}} f_{x_{m,r},y_{k,m,r}} \quad (2a)$$

subject to:

$$\text{[C1]: } \sum_{r\in R} x_{m,r} \geq 1, \quad \forall m \quad (2b)$$

$$\text{[C2]: } \sum_{r\in R} x_{m,r}\cdot y_{k,m,r} \geq 1, \quad \forall m,k \quad (2c)$$

$$\text{[C3]: } \sum_{m\in M} x_{m,r}\cdot s_m \leq C_r, \quad \forall r \quad (2d)$$

$$\text{[C4]: } x_{m,r} \in \{0,1\}, \quad \forall m,r \quad (2e)$$

$$\text{[C5]: } y_{k,m,r} \in \{0,1\}, \quad \forall m,k,r \quad (2f)$$

Constraint [C1] ensures that each content item is cached at least once in the network. Constraint [C2] guarantees that every user can access each content item. Constraint [C3] enforces the cache capacity constraint at each router. Constraints [C4] and [C5] define the binary nature of the decision variables.

III. Problem Reformulation with Inner Approximation

Before presenting the mathematical details, we provide a brief intuition behind the proposed reformulation. The core difficulty of the problem arises from the multiplicative coupling between cache placement $x_{m,r}$ and routing assignment $y_{k,m,r}$, which makes the feasible region non-convex. Our approach first represents this coupling using a difference-of-convex (DC) function as illustrated in Fig. 2, then constructs a sequence of convex inner approximations that gradually tighten the surrogate problem. This allows the algorithm to maintain feasibility while iteratively refining cache and routing decisions. The optimization problem formulated in the previous section is inherently non-convex. To make the problem tractable, we apply inner approximation methods such as DC to obtain a convex surrogate formulation.

A. Difference-of-Convex (DC) Reformulation

DC programming addresses nonconvex problems whose objectives/constraints can be written as the *difference of two convex functions*. In our case, every bilinear term $x_{m,r}\cdot y_{k,m,r}$ is rewritten via the identity:

$$\begin{aligned} x_{m,r}\cdot y_{k,m,r} &= \tfrac{1}{4}(x_{m,r}+y_{k,m,r})^2 - \tfrac{1}{4}(x_{m,r}-y_{k,m,r})^2 \\ &= f_1(x_{m,r},y_{k,m,r}) - f_2(x_{m,r},y_{k,m,r}), \end{aligned} \quad (3)$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are convex quadratics. Applying the convex–concave procedure, we majorize the concave part $-f_2$ by its first-order (affine) upper bound at the current iterate $(\hat{x}_{m,r},\hat{y}_{k,m,r})$. Equivalently, we linearize f_2 as:

$$\begin{aligned} (x_{m,r}-y_{k,m,r})^2 \approx\ & (\hat{x}_{m,r}-\hat{y}_{k,m,r})^2 \\ & + 2(\hat{x}_{m,r}-\hat{y}_{k,m,r})(x_{m,r}-\hat{x}_{m,r}) \\ & - 2(\hat{x}_{m,r}-\hat{y}_{k,m,r})(y_{k,m,r}-\hat{y}_{k,m,r}). \end{aligned} \quad (4)$$

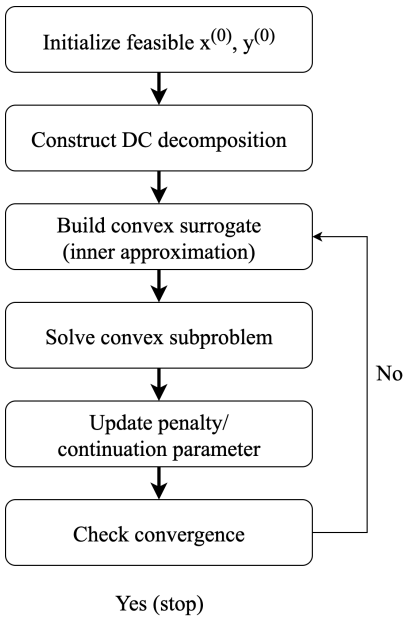

Fig. 2. Flowchart of the proposed DC-based joint cache allocation and router assignment algorithm, illustrating initialization, surrogate construction, convex subproblem solving, and iterative penalty updates.

This means that $-f_2$ is replaced by its affine upper bound and the resulting surrogate objective and constraints are convex. We apply this replacement both in the objective (equation (2a)) and in Constraint [C2] wherever the product $x_{m,r}\cdot y_{k,m,r}$ appears, yielding a tractable convex subproblem at each iteration. Although deriving a closed-form dual is challenging because of the bilinear terms and combinatorial feasibility constraints, a partial Lagrangian relaxation provides useful structural insights. By relaxing the coupling constraint $x_{m,r}\cdot y_{k,m,r}\geq 1$, the associated dual variables can be interpreted as "service penalties" indicating the marginal cost of content unavailability. Similarly, relaxing the cache capacity constraints quantifies the shadow cost of storage scarcity. These dual interpretations justify our use of penalty-augmented DC programming, as the penalty parameters act analogously to dynamic dual multipliers. This perspective helps validate the surrogate problem and explains its ability to approximate the original non-convex formulation while preserving key structural relationships.

B. Relaxation

Next, we relax the binary constraints [C4] and [C5] by allowing the variables to be placed in the continuous interval $[0,1]$. However, to maintain practicality and promote near-discrete solutions that are compatible with real-world implementations [23], [24], we introduce two auxiliary constraints to replace original [C4] and [C5] that can preserve convexity while promoting binary-like behavior:

$$[c_4]: \begin{cases} x_{m,r}-x_{m,r}^2 \leq 0, & \forall m,r, \\ 0\leq x_{m,r}\leq 1. \end{cases} \quad (5)$$

$$[c_5]: \begin{cases} y_{k,m,r} - y_{k,m,r}^2 \leq 0, & \forall k, m, r, \\ 0 \leq y_{k,m,r} \leq 1. \end{cases} \tag{6}$$

These quadratic constraints are also approximated using DC programming and incorporated into the objective function via a penalty method, with penalty parameters λ_1 and λ_2. The penalty parameters λ_1 and λ_2 regulate the trade-off between solution feasibility and approximation accuracy. In practice, they are initialized with moderate positive values and can be gradually increased during successive iterations to enforce near-binary values of $x_{m,r}$ and $y_{k,m,r}$. This continuation strategy allows the algorithm to avoid premature convergence to poor local minima while ensuring that the final solutions remain close to binary, consistent with the original problem formulation. The penalized objective is then formulated as:

$$\begin{aligned} f(x_{m,r}, y_{k,m,r}) = & \sum_{k\in K}\sum_{r\in R}\sum_{m\in M} h_{r,k}\cdot p_{k,m}\cdot s_m \cdot \\ & \Bigg[\frac{1}{4}(x_{m,r}+y_{k,m,r})^2 - \frac{1}{4}(\hat{x}_{m,r}-\hat{y}_{k,m,r})^2 \\ & - \frac{1}{2}(\hat{x}_{m,r}-\hat{y}_{k,m,r})(x_{m,r}-\hat{x}_{m,r}) \\ & + \frac{1}{2}(\hat{x}_{m,r}-\hat{y}_{k,m,r})(y_{k,m,r}-\hat{y}_{k,m,r}) \\ & + \lambda_1 \sum_{r\in R}\sum_{m\in M}(x_{m,r} - \hat{x}_{m,r}^2 - 2\hat{x}_{m,r}(x_{m,r}-\hat{x}_{m,r})) \\ & + \lambda_2 \sum_{k\in K}\sum_{r\in R}\sum_{m\in M} \\ & (y_{k,m,r} - \hat{y}_{k,m,r}^2 - 2\hat{y}_{k,m,r}(y_{k,m,r}-\hat{y}_{k,m,r}))\Bigg] \end{aligned} \tag{7}$$

C. Final Reformulated Optimization Problem

The convex approximation of the original problem can now be defined as follows:

$$\min_{\{X_{m,r}\},\{Y_{k,m,r}\}} f(x_{m,r}, y_{k,m,r}) \tag{8a}$$

subject to:

$$[c_1]: \sum_{r\in R} x_{m,r} \geq 1, \quad \forall m \tag{8b}$$

$$\begin{aligned} [c_2]: 1 + \sum_{r\in R}\Bigg[& \frac{1}{4}(x_{m,r}-y_{k,m,r})^2 - \frac{1}{4}(\hat{x}_{m,r}+\hat{y}_{k,m,r})^2 \\ & - \frac{1}{2}(\hat{x}_{m,r}+\hat{y}_{k,m,r})(x_{m,r}-\hat{x}_{m,r}) \\ & - \frac{1}{2}(\hat{x}_{m,r}+\hat{y}_{k,m,r})(y_{k,m,r}-\hat{y}_{k,m,r})\Bigg] \leq 0, \quad \forall k, \forall m \end{aligned} \tag{8c}$$

$$[c_3]: \sum_{m\in M} x_{m,r}\cdot s_m \leq C_r, \quad \forall r \tag{8d}$$

$$[c_4]: 0 \leq x_{m,r} \leq 1, \quad \forall m, \forall r \tag{8e}$$

$$[c_5]: 0 \leq y_{k,m,r} \leq 1, \quad \forall m, \forall r, \forall k. \tag{8f}$$

Constraint [C2] requires that for every user–content pair, at least one router storing the requested item must be assigned to serve the user. This condition introduces the bilinear term $x_{m,r}\cdot y_{k,m,r}$ and is therefore non-convex. To convexify it, we express the bilinear product as a difference of convex quadratic functions and linearize the concave part using a first-order approximation at the current iterate. This yields the surrogate formulation in equation (8). This reformulation ensures that the logical requirement, every user–content pair must be served by at least one feasible router, is preserved in the convex approximation.

D. Convergence Analysis

To prove the convergence of the proposed iterative approximation algorithm for any fixed ξ, we show that $g(X_k, \xi)$ forms a non-decreasing sequence with respect to the iteration index k. The following inequalities will then hold [16]:

$$g(x_{m,r}^{j+1}, y_{k,m,r}^{j+1})_\lambda \leq \hat{g}(x_{m,r}^{j+1}, x_{m,r}^{j}, y_{k,m,r}^{j+1}, y_{k,m,r}^{j})_\lambda \tag{9}$$

$$g(x_{m,r}^{j+1}, y_{k,m,r}^{j+1})_\lambda \leq \hat{g}(x_{m,r}^{j}, x_{m,r}^{j}, y_{k,m,r}^{j}, y_{k,m,r}^{j})_\lambda \tag{10}$$

$$g(x_{m,r}^{j+1}, y_{k,m,r}^{j+1})_\lambda = g(x_{m,r}^{j}, y_{k,m,r}^{j})_\lambda \tag{11}$$

The intuition behind inequalities (9)–(11) is that the linearized surrogate function serves as a tight convex upper bound of the original non-convex function at each iteration. By construction, the approximation is exact at the current point and overestimates elsewhere, which guarantees that the sequence of objective values forms a non-decreasing (monotonic) sequence. As a result, the iterative algorithm generates progressively improved feasible solutions and converges to a stationary point of the original non-convex problem, as established in [16].

E. Computational Complexity

The computational complexity of solving the reformulated conic optimization problem via interior-point methods is polynomial with respect to the problem dimension [16]. We compute the arithmetic and Newton complexities as follows:

$$\begin{aligned} \text{Complexity}(p,\varepsilon) = \mathcal{O}(1)\cdot\sqrt{m+1}\cdot n\Big(n^2 + m + \sum_{i=1}^{m} k_i^2\Big) \\ \cdot \log\left(\frac{\text{size}(p) + \|\text{Data}(p)\|_1 + \varepsilon^2}{\varepsilon}\right) \end{aligned} \tag{12}$$

$$\begin{aligned} \text{Compl}^{\text{Nwt}}(p,\varepsilon) = \mathcal{O}(1)\cdot\sqrt{m+1} \\ \cdot \log\left(\frac{\text{size}(p) + \|\text{Data}(p)\|_1 + \varepsilon^2}{\varepsilon}\right) \end{aligned} \tag{13}$$

In equations (12) and (13), n denotes the number of decision variables, i.e., the total number of cache allocation and router assignment variables, m represents the number of constraints in the reformulated problem, and k_i indicates the dimension of the i^{th} second-order cone constraint introduced in the convex reformulation. With these definitions, the complexity expressions quantify the arithmetic and Newton iterations required by interior-point methods to solve the surrogate convex problem within an ε-optimal solution tolerance.

IV. Simulation and Numerical Results

A. Simulation Setup

We evaluate the proposed framework through numerical simulations based on the system model presented in section II with parameters in Table II. It should be noted that content sizes are normalized to one chunk in the simulation setup. This normalization is adopted for clarity, allowing us to isolate the effect of routing–placement coupling without introducing variability from heterogeneous object sizes. However, the proposed optimization framework directly supports arbitrary content sizes s_m, which would make constraint [C3] a heterogeneous knapsack-type constraint rather than a simple counting rule. Incorporating realistic size variability may further influence cache placement decisions, and exploring this direction constitutes an important extension of the current study. We compare our solution with several existing caching strategies in terms of transportation cost, overall network performance, and computational complexity. We consider a scenario in which multiple content items are available, and the request rate for each item by a user node is determined by its popularity at that node. To model content popularity, we employ the Zipf distribution, which has been widely validated for characterizing file access patterns on the Internet [17]–[19]. Specifically, we define a request generation function at each user node that samples content based on the Zipf distribution. The popularity skew is controlled by the exponent z_{pop}, where higher values indicate more concentrated interest in a smaller set of contents.

TABLE II
Simulation Parameters

Parameter	Value
Number of nodes ($\|V\|$)	50
Number of server nodes	5
Number of router nodes	30
Number of user nodes	15
Topology model	Erdős–Rényi random graph
Monte Carlo runs	100
Number of content items (M)	100
Content size (s_m)	1 chunk (normalized)
Cache capacity (C_r)	10 chunks per router
Request distribution	Zipf law
Zipf exponent (z_{pop})	$1 \leq z_{\text{pop}} \leq 10$
Performance metric	Transportation cost (hop-weighted traffic)
Baselines	CEE, Probp ($p = 0.5$)

B. Impact of Content Popularity

In our first experiment, we examine the impact of the Zipf exponent z_{pop} on the transportation cost in the network. The value of z_{pop} is varied from 1 to 5, and the number of DC iterations for inner approximation is fixed at 10. It should be noted that the algorithm typically converges within just six iterations. As shown in Fig. 3, higher content popularity among users leads to reduced transportation cost since the same content is more likely to be served from nearby caches.

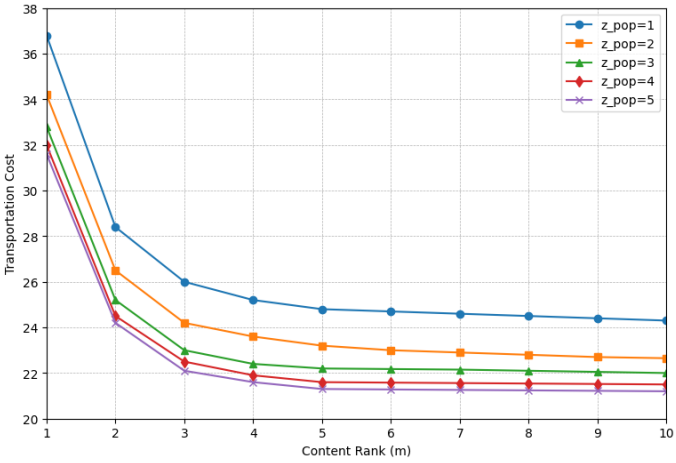

Fig. 3. Popularity distribution under Zipf's law for different exponents z_{pop}. The x-axis represents content rank (1 = most popular), and higher z_{pop} values correspond to more skewed distributions.

Fig. 4 demonstrates the convergence behavior of the surrogate and original objective functions approximated by $g(X_k, \xi)$ for a number of DC iterations. The results show that both the surrogate function and the true objective decrease monotonically and converge to nearly identical values. Most of the improvement occurs within the first four to six iterations, after which the functions stabilize. This confirms both the rapid convergence of the algorithm and the accuracy of the convex surrogate in approximating the original non-convex formulation.

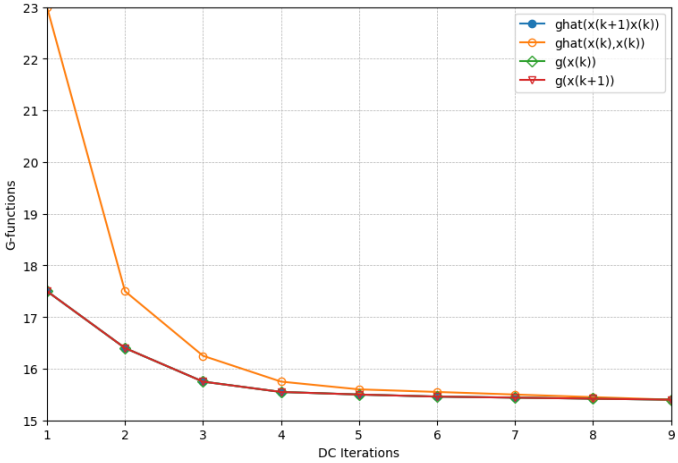

Fig. 4. Convergence of the proposed DC-based algorithm, showing the decrease in objective value across iterations for a representative network instance.

C. Monte Carlo Simulations

To evaluate the robustness of our proposed cache allocation and router assignment across various network topologies, we perform Monte Carlo simulations. In each iteration, the topology is randomly regenerated and the Zipf exponent is

swept over its full range. This approach captures the effect of topology variance on performance. As shown in Fig. 5, the reduction in transport cost with increasing content popularity is consistently observed across multiple randomized network topologies, which demonstrates the robustness of the proposed method.

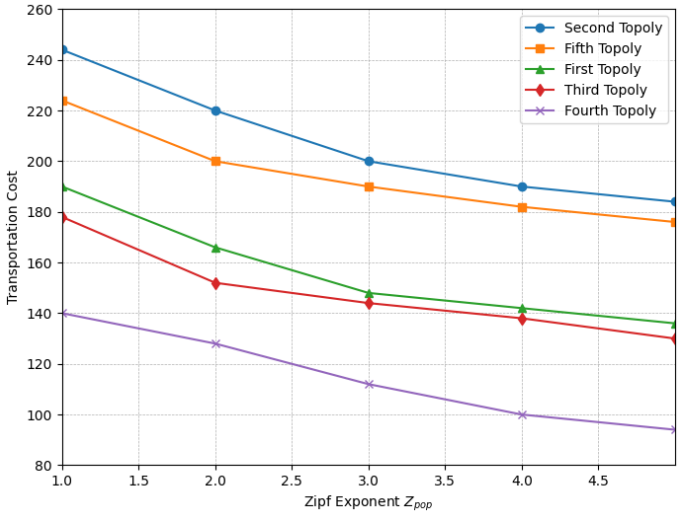

Fig. 5. Transportation cost as a function of the Zipf exponent z_{pop} under a fixed network topology. Increasing z_{pop} emphasizes highly popular items, leading to improved caching efficiency and reduced overall transportation cost.

D. Comparison with Baseline Methods

We further compare the proposed method with two widely used baseline strategies: Cache Everything Everywhere (CEE) and Probabilistic Caching (Probp) [20]–[22]. CEE caches every content item at every router. While this approach maximizes redundancy and incurs high storage usage, it achieves the lowest transportation cost under static traffic patterns, making it an effective lower bound. In contrast, Probp caches content probabilistically. In our experiments, we set the cache probability to $p = 0.5$, meaning half of the requested contents are cached across the network. As p approaches 1, Probp's behavior begins to resemble that of CEE. As illustrated in Fig. 6, the proposed method consistently achieves near-optimal performance while avoiding excessive redundancy and storage waste associated with CEE. It should be noted that Figures 3 and 6 use different numerical scales because they correspond to different experimental conditions. Fig. 3 varies the Zipf exponent under a fixed topology with lower aggregate request volume, whereas Fig. 6 averages transportation cost across multiple randomized topologies and higher traffic load. As a result, absolute cost values differ in scale even though the metric definition is identical.

E. Computational Complexity

Finally, we analyze the computational complexity of the proposed algorithm. Fig. 7 shows the arithmetic complexity as a function of the number of users, while it illustrates the

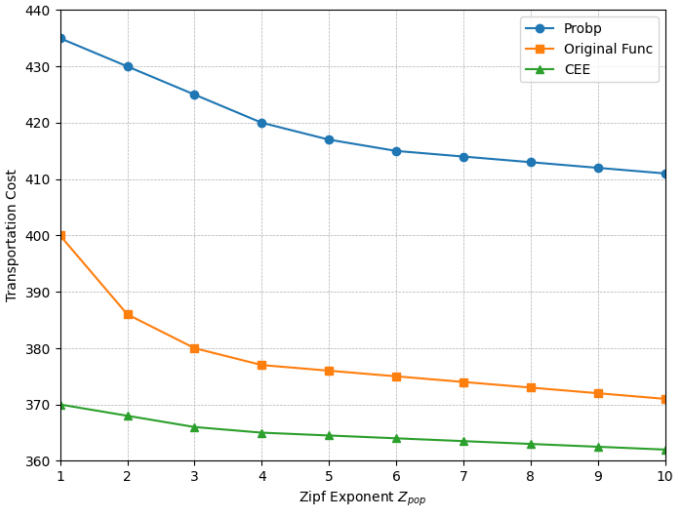

Fig. 6. Comparison of transportation cost across different caching strategies as the Zipf exponent z_{pop} varies. The proposed DC-based method consistently outperforms baseline approaches across all popularity skew levels.

Newton complexity. The results confirm that the arithmetic complexity grows polynomially with network size, whereas Newton complexity increases nearly linearly, demonstrating the scalability of the interior-point method used in solving the convex reformulation.

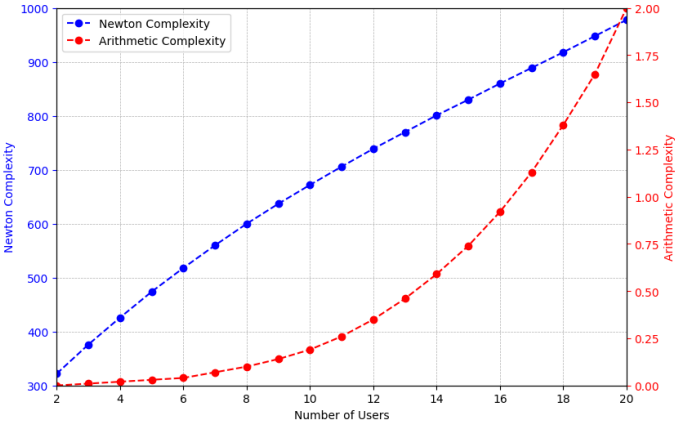

Fig. 7. Computational complexity of the proposed method as a function of the number of users. The left axis reports arithmetic complexity per iteration, while the right axis shows the corresponding Newton-step complexity.

F. Validation of Surrogate Approximation

Lastly, we validate the accuracy of the DC approximation by comparing the objective function values of the surrogate and the original problem for all Monte Carlo runs. As shown in Fig. 8, both formulations converge closely after a few iterations, confirming that the surrogate optimization faithfully captures the behavior of the original non-convex problem.

V. Conclusion and Future Work

This paper presented a unified optimization framework for joint cache allocation and router assignment in Information-

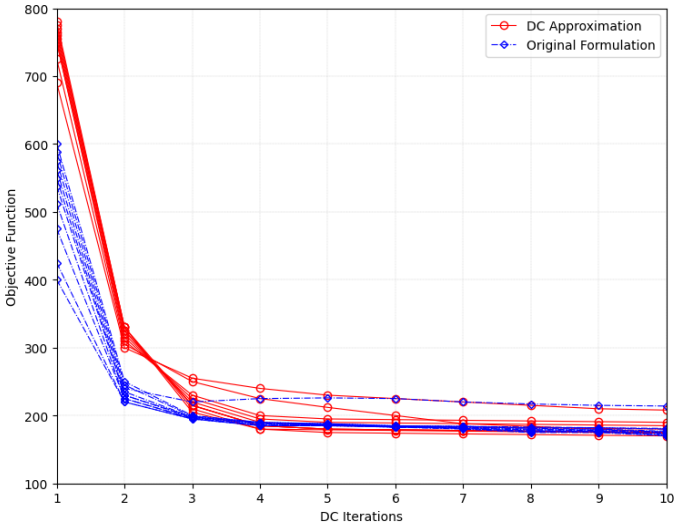

Fig. 8. Comparison between the original non-convex objective and the convex surrogate objective during the iterative optimization process, illustrating tight approximation and stable convergence behavior.

Centric Networks (ICNs). Unlike models that treat these decisions separately, our formulation captures their coupling through a non-convex optimization problem, which we solve using inner approximation, DC programming, and convex relaxation to obtain a tractable iterative algorithm. Extensive simulations across randomized topologies show that the proposed method consistently reduces transportation cost, converges rapidly, and scales efficiently. Comparisons with baseline caching strategies confirm that our approach achieves near-optimal performance without the redundancy inherent in naive schemes. While we adopt a static Zipf popularity model, the framework naturally supports temporal updates via periodic convex surrogate recalibration or online DC iterations.Beyond technical gains, the method provides a foundation for economically motivated in-network caching strategies, where optimized placement and routing jointly reduce operational cost and improve QoS.

Future work includes incorporating heterogeneous routers, mobility-driven dynamics, and time-varying popularity $p_{k,m}(t)$, as well as exploring heterogeneous cache capacities and online learning for popularity estimation to further align the model with real-world ICN deployments.

REFERENCES

[1] Zhang, Zhe, Chung-Horng Lung, Xin Wei, Mingkai Chen, Subhajit Chatterjee, and Zhicai Zhang. "In-network caching for ICN-based IoT (ICN-IoT): A comprehensive survey." IEEE Internet of Things Journal 10, no. 16 (2023): 14595-14620.

[2] Qaiser, Firdous, Khoula Said Al Harthy, Mudassar Hussain, Jaroslav Frnda, Rashid Amin, Rahma Gantassi, and Muhammad D. Zakaria. "Classifications and Analysis of Caching Strategies in Information-Centric Networking for Modern Communication Systems." Engineering Reports 7, no. 2 (2025): e70005.

[3] Jacobson, Van, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. "Networking named content." In Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp. 1-12. 2009.

[4] Networking, Named Data. "Named data networking." Named Data Networking 20 (2012).

[5] Xylomenos, George, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. "A survey of information-centric networking research." IEEE communications surveys & tutorials 16, no. 2 (2013): 1024-1049.

[6] Naeem, Muhammad Ali, Ikram Ud Din, Yahui Meng, Ahmad Almogren, and Joel JPC Rodrigues. "Centrality-Based On-Path Caching Strategies in NDN-Based Internet of Things: A Survey." IEEE Communications Surveys & Tutorials (2024).

[7] Zhang, Wancai, and Rui Han. "An Efficient Multipath-Based Caching Strategy for Information-Centric Networks." Electronics 14, no. 3 (2025): 439.

[8] Doan Van, Dong, and Qingsong Ai. "In-network caching in information-centric networks for different applications: A survey." Cogent Engineering 10, no. 1 (2023): 2210000.

[9] Chao, Yichao, and Rui Han. "A Hierarchical Cache Architecture-Oriented Cache Management Scheme for Information-Centric Networking." Future Internet 17, no. 1 (2025).

[10] Banerjee, Subharthi, Pejman Ghasemzadeh, Michael Hempel, and Hamid Sharif. "Topography relaxation in determining unsafe state intersections for uncertain cps." IEEE Sensors Letters 4, no. 4 (2020): 1-4.

[11] Leira, Luís, Miguel Luís, and Susana Sargento. "Context-based caching in mobile information-centric networks." Computer Communications 193 (2022): 214-223.

[12] Chaudhary, Pankaj, and Neminath Hubballi. "PeNCache: Popularity based cooperative caching in named data networks." Computer Networks 257 (2025): 110995.

[13] Hu, Zhaoming, Chao Fang, Zhuwei Wang, Jining Chen, Shu-Ming Tseng, and Mianxiong Dong. "Joint Content Caching and Request Routing for User-Centric Many-Objective Metaverse Services." IEEE Transactions on Network Science and Engineering (2025).

[14] Khan, Amir Raza, Umar Shoaib, and Hannan Bin Liaqat. "SARAC4N: Socially and Resource-Aware Caching in Clustered Content-Centric Networks." Future Internet 17, no. 8 (2025): 341.

[15] Koide, Masaki, Naoyuki Matsumoto, and Tomofumi Matsuzawa. "Caching method for information-centric ad hoc networks based on content popularity and node centrality." Electronics 13, no. 12 (2024): 2416.

[16] Shapiro, Alexander, and Arkadi Nemirovski. "On complexity of stochastic programming problems." In Continuous optimization: Current trends and modern applications, pp. 111-146. Boston, MA: Springer US, 2005.

[17] Padmanabhan, Venkata N., and Lili Qiu. "The content and access dynamics of a busy web site: Findings and implications." In Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 111-123. 2000.

[18] Newman, Mark EJ. "Power laws, Pareto distributions and Zipf's law." Contemporary physics 46, no. 5 (2005): 323-351.

[19] Adamic, Lada A., and Bernardo A. Huberman. "Power-law distribution of the world wide web." science 287, no. 5461 (2000): 2115-2115.

[20] Li, Zhe, and Gwendal Simon. "Time-shifted tv in content centric networks: The case for cooperative in-network caching." In 2011 IEEE international conference on communications (ICC), pp. 1-6. IEEE, 2011.

[21] Laoutaris, Nikolaos, Hao Che, and Ioannis Stavrakakis. "The LCD interconnection of LRU caches and its analysis." Performance Evaluation 63, no. 7 (2006): 609-634.

[22] Rajahalme, Jarno, Mikko Särelä, Pekka Nikander, and Sasu Tarkoma. "Incentive-compatible caching and peering in data-oriented networks." In Proceedings of the 2008 ACM CoNEXT Conference, pp. 1-6. 2008.

[23] Pham Dinh, Tao, and Hoai An Le Thi. "Recent advances in DC programming and DCA." Transactions on computational intelligence XIII (2014): 1-37.

[24] Guenin, Bertrand, Jochen Könemann, and Levent Tuncel. A gentle introduction to optimization. Cambridge University Press, 2014.

[25] Eslami, Parisa, Mohammad Hossein Amerimehr, and Seyed Pooya Shariatpanahi. "A new framework for mobile edge caching by proposing flexible user in heterogeneous cellular networks." IEEE Access 8 (2020): 188938-188950.