Intelligent Collaborative Edge Caching via Contextual Bandits and Convex Relaxation

Guobing Zou[1], Shuyi Ye[1], Song Yang[1], Shengye Pang[1(✉)], Shengxiang Hu[2], Yanglan Gan[3(✉)], and Bofeng Zhang[4]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
{gbzou, yeshuyi1219, yangsong, pangsy}@shu.edu.cn
[2] Shanghai Ideal Information Industry (Group) Co., Ltd. China Telecom, Shanghai 201315, China
husx@chinatelecom.cn
[3] School of Computer Science and Technology, Donghua University, Shanghai 201620, China
ylgan@dhu.edu.cn
[4] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China
bfzhang@sspu.edu.cn

Abstract. Edge Data Caching (EDC) enables low-latency delivery and alleviates backhaul congestion by caching data in close proximity to end users. However, current learning-driven methods struggle to capture the multi-dimensional business value of data. Consequently, inefficient resource allocation degrades critical services and compromises the Quality of Service (QoS). Moreover, physical node failures are insufficiently addressed, risking severe backhaul delays. To address these issues, we formulate the Value and Failure-aware Collaborative Edge Data Caching (VF-CEDC) problem as a mixed-integer linear programming (MILP) model jointly optimizing business value and expected latency under physical uncertainties. To solve this, we propose an Online Learning and Convex Relaxation Algorithm (OLCRA). First, OLCRA leverages a Contextual Multi-Armed Bandit (CMAB) to adaptively infer data values. Second, it resolves the discrete caching topology via convex relaxation and probabilistic rounding, yielding near-optimal polynomial-time decisions. Extensive experiments on a real-world dataset show that OLCRA reduces average system cost by 32.3% and expected latency by 38.3% against eight competing methods.

Keywords: Edge Data Caching, Contextual Bandits, Convex Relaxation, Business Value, Failure Awareness.

1 Introduction

The proliferation of mobile applications and IoT technologies has triggered an exponential surge in wireless network traffic. To alleviate these bottlenecks, Mobile Edge Computing (MEC) decentralizes computational and storage capabilities to the network edge [1]. Within this distributed architecture, Edge Data Caching (EDC) [2, 3] plays a critical role. By proactively storing frequently requested data in close proximity to end users, EDC effectively mitigates backhaul congestion, minimizes latency, and enhances service quality. However, existing EDC studies largely overlook two key factors that can significantly impair the reliability and overall performance of caching systems in dynamic edge environments.

Although recent learning-driven caching methods have progressed beyond simple popularity models, they struggle to capture multi-dimensional data characteristics (e.g., delay sensitivity, dynamic utility) [4]. This inefficiency allows high-access but low-value data to exhaust edge storage, prematurely evicting less-frequent critical data. Furthermore, edge service continuity is threatened by uncertainties across physical nodes (e.g., hardware aging, overload-induced failures) [5] and communication links. Existing studies often prioritize placement efficiency under idealized assumptions, overlooking these fundamental risks [6, 7]. Since traditional fault-tolerance strategies (e.g., redundancy) inherently conflict with resource-constrained edge networks, unmitigated physical failures easily render data inaccessible. Ultimately, neglecting both the dynamic business value and the underlying physical risks severely compromises QoS fulfillment and the long-term reliability of collaborative caching systems.

To address these issues, we formulate the Value and Failure-aware Collaborative Edge Data Caching (VF-CEDC) problem to capture dynamic data value and physical failures. To effectively solve it, we propose an Online Learning and Convex Relaxation Algorithm (OLCRA), integrating online value estimation with mathematical optimization for robust caching decisions. Our main contributions are summarized as follows:

- We formulate the VF-CEDC problem as a mixed-integer linear programming (MILP) model. This formulation introduces a multi-dimensional business value utility model and a dual physical failure model.
- We propose the two-stage OLCRA, coupling Linear Upper Confidence Bound (LinUCB) for data utility estimation with a primal-dual interior-point method and probabilistic rounding to yield near-optimal caching strategies in polynomial time.
- Extensive experiments on a real-world dataset demonstrate that OLCRA significantly outperforms eight competing methods across multiple metrics, verifying its superior effectiveness, efficiency, and robustness.

2 Related Work

Given the limited storage and computational capacities of distributed edge servers, determining optimal data caching strategies has become a critical research focus to maximize overall system performance and user satisfaction.

Recent studies explore learning-based approaches to overcome traditional limitations like LRU and LFU. For instance, to adapt to dynamic request patterns, advanced methods like FDRL [8] and RoCoCache [9] integrate federated learning for distributed optimization. Although effectively tracking access patterns, these methods inherently equate data utility with popularity, overlooking multi-dimensional business value (e.g., semantic priority and latency requirements). Furthermore, recent MAB and CMAB frameworks [10, 11] address uncertain environments but predominantly focus on hit-rate maximization. In resource-constrained edge networks, this lack of value awareness inevitably causes the premature eviction of low-frequency critical data, severely impairing Quality of Service (QoS).

Meanwhile, mitigating physical failure risks is critical to maintaining high availability in MEC. Traditional strategies mainly rely on redundancy. Qiu et al. [12] improved service continuity through primary-backup placement, and Ma et al. [6] explicitly considered network and device failures in collaborative caching. Furthermore, He et al. [13] protected data distribution via erasure coding, while Luo et al. [7] incorporated failure factors alongside data popularity. However, these isolated mechanisms impose significant computational and storage overheads on resource-constrained edge servers. They also frequently treat failures as independent static probabilities, failing to capture the dynamically coupled uncertainties of heterogeneous edge environments.

To address the limitations of existing research, this paper formulates the VF-CEDC problem. By integrating multi-dimensional business value sensing with dual physical failure awareness, we construct a comprehensive optimization model. Ultimately, we propose OLCRA, an efficient online learning and convex relaxation algorithm, to reliably and intelligently formulate near-optimal collaborative caching strategies in dynamic, heterogeneous edge environments.

3 System Model

3.1 Collaborative Edge Caching

As illustrated in **Fig. 1** we consider a typical heterogeneous edge collaborative caching environment that consists of a cloud data center v_0, a set of N edge servers, and a set of F data items to be cached across the network edge.

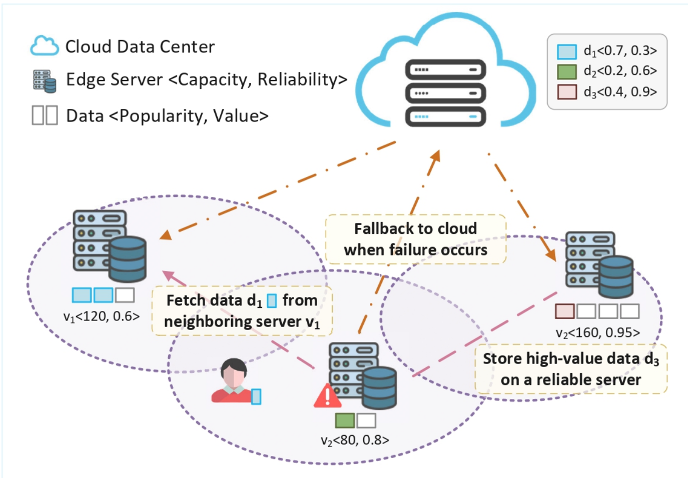


Fig. 1. An illustrative de example of value and failure-aware collaborative edge data caching in a heterogeneous edge network. In this example, a request issued at v_2 is normally served by neighboring server v_1, but fall ds back to the cloud upon failure, while the high-value data item d_3 is placed on the more reliable server v_3.

Definition 1 (Edge Server). Edge servers, denoted by $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, are connected by physical links $\mathcal{E}$ to form an undirected graph $G = (\mathcal{V}, \mathcal{E})$. Each server is characterized by a 3-tuple $v_i = \langle C_i, \beta_i, L_{i,i} \rangle$, where C_i is the storage capacity, β_i is the hardware aging rate, and $L_{i,i}$ is the local hit delay.

Definition 2 (Edge Data). Let data set $\mathcal{D} = \{d_1, \ldots, d_F\}$, where $d_k = \langle S_k, q_k, t_{ge}, \ \Gamma_{th}, \rho_k \rangle$ denotes its size, request probability, generation time, timeliness threshold, and semantic priority, acting as the online learning contextual feature $z_{k,t}$. For requests served by $\mathcal{V} \cup \{v_0\}$, the binary variable for d_k at time slot t is:

$$x_{i,k}(t) = \begin{cases} 1, & \text{if } d_k \text{ is cached on } v_i \text{ in time slot } t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Next, let $y_{i,j}^k(t)$ indicate whether the local server v_i allocates the request for data d_k to the target server v_j for retrieval in time slot t:

$$y_{i,j}^k(t) = \begin{cases} 1, & \text{if } v_i \text{ allocates the request for } d_k \text{ to } v_j \text{ in time slot } t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

3.2 Business Value Utility Model

Base Popularity. To capture sparse and fluctuating request patterns, we adopt the Mandelbrot-Zipf model [14], where the base request probability q_k of data d_k is defined as:

$$q_k = \frac{(r_k + \mu)^{-\gamma}}{\sum_{j=1}^{F} (r_j + \mu)^{-\gamma}} \tag{3}$$

where r_k is the local popularity rank, $\gamma > \ 0$ controls the decay rate, and $\mu \geq 0$ is a smoothing factor for sparse requests.

Timeliness Reliability Factor. Edge data values are highly imbalanced, with few critical items dominating the timeliness value. To capture this, we introduce Age of Information (AoI) theory [15] and define the timeliness reliability factor for data d_k as:

$$T_k(t) = \left[1 + (\frac{\Delta_k(t)}{\Gamma_{th}})\right]^{-1} \tag{4}$$

where $\Delta_{k(t)} = t - t_{ge}$ is the instantaneous AoI, Γ_{th} the timeliness threshold, and $\psi >$ 1 the decay rate. It approaches 1 if $\Delta_k(t) < \Gamma_{th}$, and 0 otherwise.

Business Value-Aware Utility. To mitigate the long-tail noise in traditional popularity models, we further integrate semantic priority with AoI-based timeliness utility to define the intrinsic business value $V_k(t)$ of data d_k in slot t as:

$$V_k(t) = \frac{\rho_k}{(q_k + \mu)^{\nu}} \cdot T_k(t) \tag{5}$$

where μ ensures stability for small probabilities, ρ_k denotes semantic priority, and ν prioritizes low-frequency critical services.

3.3 Dual Physical Failure Model

Edge servers are exposed to volatile physical conditions, and their stability is affected by both hardware aging and bursty computational loads. The server outage risk $p_i^{srv}(t)$ of server v_i in time slot t is defined as:

$$p_i^{\text{srv}}(t) = \alpha\left(1 - e^{-\beta_i t}\right) + (1 - \alpha)\left[\frac{\left(\widetilde{O}_\iota(t) - \varsigma\right)^+}{1 - \varsigma}\right]^p \tag{6}$$

Here, $e^{-\beta_i t}$ models hardware aging with weight α, while the second term captures overloads. For linearity, $\widetilde{O}_\iota(t)$ is approximated at $t - 1$ with threshold ς and exponent p.

Let $l_{i,j}(t) \in \{0,1\}$ denote link availability between v_i and v_j. Under multipath fading, the failure probability $p_{i,j}^{\text{link}}(t)$ follows the Rayleigh model [16] as:

$$p_{i,j}^{\text{link}}(t) = P\left(l_{i,j}(t) = 0\right) = \int_0^{\gamma_{th}} f\left(\gamma, \overline{\gamma_{\iota,J}}(t)\right) d\gamma = 1 - \exp\left(\frac{\gamma_{th}}{\overline{\gamma_{\iota,J}}(t)}\right) \tag{7}$$

Here, γ_{th} and $\overline{\gamma_{\iota,J}}(t)$ denote the decoding threshold and average signal-to-noise ratio, strictly reducing to $p_{i,i}^{link}(t) = 0$ for local access $i = j$.

For a request from v_i to v_j, the reliability factor $\Phi_{i,j}(t)$ defines the probability of transmission uninterrupted by server or link failures:

$$\Phi_{i,j}(t) = \left(1 - p_j^{\text{srv}}(t)\right) \cdot \prod_{(u,v)\in\mathcal{P}_{j,i}} \left(1 - p_{u,v}^{\text{link}}(t)\right) \tag{8}$$

where $\mathcal{P}_{j,i} \subseteq \mathcal{E}$ denotes physical links from v_j to v_i. $\Phi_{i,j}(t)$ decays exponentially with hop count, reducing exclusively to the local physical state of v_i when $i = j$.

Let $L_{i,j}$ be the $v_i - v_j$ delay. Failures redirect requests to v_0 with penalty D_{cloud}. Given $\Phi_{i,0}(t) = 1$ and $L_{i,0} = D_{\text{cloud}}$, the expected latency $E\left[L_{i,k}(t)\right]$ to obtain d_k is:

$$E\left[L_{i,k}(t)\right] = \sum_{v_j\in\mathcal{V}\cup\{v_0\}} y_{i,j}^k(t) \cdot \left[\Phi_{i,j}(t) \cdot L_{i,j} + \left(1 - \Phi_{i,j}(t)\right) \cdot D_{\text{cloud}}\right] \tag{9}$$

Specifically, when $j = v_0$, the bracketed term naturally degenerates to D_{cloud}.

3.4 VF-CEDC Problem

The VF-CEDC problem jointly optimizes caching and request allocation under edge storage constraints. To unify latency minimization and value maximization, we define the system cost as expected latency minus weighted value reward, formulated as:

$$\min_{X(t),Y(t)} \sum_{v_i\in\mathcal{V}} \sum_{d_k\in\mathcal{D}} \lambda_{i,k}(t) \cdot E\left[L_{i,k}(t)\right] - \omega \sum_{v_i\in\mathcal{V}} \sum_{d_k\in\mathcal{D}} V_k(t) \cdot x_{i,k}(t) \tag{10}$$

$$\text{s.t.} \quad \sum_{v_j\in\mathcal{V}\cup\{v_0\}} y_{i,j}^k(t) = 1, \quad \forall v_i \in \mathcal{V}, d_k \in \mathcal{D} \tag{11}$$

$$y_{i,j}^k(t) \le x_{j,k}(t), \quad \forall v_i, v_j \in \mathcal{V}, d_k \in \mathcal{D} \tag{12}$$

$$\sum_{d_k \in \mathcal{D}} x_{i,k}(t) \cdot S_k \leq C_i, \quad \forall v_i \in \mathcal{V} \tag{13}$$

$$x_{i,k}(t) \in \{0,1\}, \quad \forall v_i \in \mathcal{V}, d_k \in \mathcal{D} \tag{14}$$

$$y_{i,j}^k(t) \in \{0,1\}, \quad \forall v_i \in \mathcal{V}, v_j \in \mathcal{V} \cup \{v_0\}, d_k \in \mathcal{D} \tag{15}$$

Objective (10) minimizes the Value-Aware Cost by balancing delay penalties and caching rewards via ω. Constraint (11) assigns each request to a single server. Constraint (12) requires data caching prior to service, excluding cloud v_0. Constraint (13) enforces capacity C_i, while (14) and (15) specify binary domains.

4 Approach

To solve the VF-CEDC problem, we propose the Online Learning and Convex Relaxation Algorithm (OLCRA), integrating value inference, relaxed optimization, and discrete strategy recovery into a unified framework, as detailed in Algorithm 1.

Algorithm 1: Online Learning and Convex Relaxation Algorithm (OLCRA)

```
Input: Edge network graph G(V, E), Data set D, Request matrix Λ(t)
Output: Discrete caching strategy X̂(t) and allocating strategy Ŷ(t)
Initialize: Inverse covariance matrix A_0^{-1} ← (1/κ) I_d, bias vector b_0 ← 0_d
For each decision time slot t ∈ T do
   For each data d_k ∈ D do Estimate V_k(t) using θ_t = A_t^{-1} b_t and context z_{k,t}
   Construct B(w, τ) given inferred value V(t) and request Λ(t)
   while τ > ϵ_0 (Interior-Point Optimization) do
      Solve KKT system ∇²B Δw = −∇B for Newton step Δw
      Update w ← w + l_step Δw and decay penalty τ ← c τ
   Parse converged vector for continuous solutions X*(t), Y*(t)
   For each edge server v_i ∈ V do
      Execute probabilistic rounding: x̂_{i,k}(t) = 𝕀{ξ ≤ x*_{i,k}(t)} for ξ ~ U(0,1)
      while Σ S_k x̂_{i,k}(t) > C_i do Evict d_k with min(V_k(t)/S_k) to meet capacity
   end For
   For each server v_i and requested data d_k do
      Identify target set C_{i,k} = {v_j ∈ V | x̂_{j,k}(t) = 1} ∪ {v_0}
      Select j* = arg min_{v_j ∈ C_{i,k}} [Φ_{i,j}(t) L_{i,j} + (1 − Φ_{i,j}(t)) D_cloud]
      Set ŷ^k_{i,j*}(t) = 1
   end For
   Deploy {X̂(t), Ŷ(t)}, observe rewards r_{k,t}, and update AoI for each data
   For each requested d_k do Update A_t^{-1} via Sherman-Morrison and update b
   end For
end For
Return X̂(t), Ŷ(t)
```

4.1 Online Value Learning via Contextual Multi-Armed Bandits

To infer the unknown intrinsic value $V_k(t)$, we formulate a Contextual Multi-Armed Bandit (CMAB) model solved via LinUCB [17]. We assume the realized value follows a linear mapping:

$$V_k(t) = \theta_*^\top z_{k,t} + \eta_{k,t} \tag{16}$$

where $z_{k,t} \in \mathbb{R}^{d}$ is the context vector, θ is the unknown weight, and $\eta_{k,t}$ denotes $\mathcal{F}_{t-1}$-adapted R-sub-Gaussian noise.

Applying the Sherman-Morrison formula to update the inverse covariance matrix A_t^{-1} limits the ridge regression complexity to $O(d^2)$ per request. To balance exploitation and exploration, the estimated dynamic value score $\hat{V}_k(t)$ is evaluated as:

$$\hat{V}_k(t) = \theta_t^\top z_{k,t} + \alpha_t |z_{k,t}|_{A_t^{-1}} \tag{17}$$

where α_t controls exploration. This estimated score $\hat{V}_k(t)$ then serves as the deterministic parameter $V_k(t)$ for the subsequent optimization phase.

4.2 Convex Relaxation and Interior-Point Method

Given deterministic $V_k(t)$, the original discrete topology is computationally intractable due to binary variables $X(t)$ and $Y(t)$. We relax these variables into the continuous domain [0,1]. By defining the expected link delay as:

$$\overline{L_{i,j}}(t) = \Phi i,j(t) L_{i,j} + \left(1 - \Phi_{i,j}(t)\right) D_{cloud} \tag{18}$$

the objective and constraints become strictly linear, yielding a convex formulation.

We adopt a primal-dual interior-point method with a logarithmic barrier $\tau > 0$ to solve the relaxed problem. During each Newton iteration, the search direction Δw for the augmented variable $w = [X, Y]^\top$ is determined by:

$$\nabla^2 \mathcal{B}(w, \tau) \Delta w = -\nabla \mathcal{B}(w, \tau) \tag{19}$$

As $\tau \to 0$, $\mathcal{B}$ converges to the continuous optimum $\left(X^*(t), Y^*(t)\right)$, which is discretized via probabilistic rounding. Upon capacity overflow, servers evict items with minimal value density $V_k(t)/S_k$ and reroute affected requests.

4.3 Computational Complexity Analysis

To validate theoretical feasibility, we analyze OLCRA regarding computational complexity and optimality guarantees.

Computational Complexity. OLCRA operates strictly in polynomial time. For $N = |\mathcal{V}|$ servers, $F = |\mathcal{D}|$ data items, and feature dimension d, the learning phase uses the Sherman-Morrison formula to reduce matrix inversion cost to $O(d^2)$ per request. In the optimization phase, relaxing the $\mathcal{NP}$-hard MILP problem, which has a worst-case complexity of $O\left(2^{N^2 F}\right)$, yields a tractable convex formulation. The primal-dual interior-point method converges in $O\left(\sqrt{N^2F}\log(1/\epsilon_0)\right)$ iterations, efficiently solving the sparse KKT system. This polynomial-time execution avoids exponential computational explosion, corroborating the ultra-low CPU time observed in our experiments.

Performance and Optimality. OLCRA's solution quality is theoretically guaranteed by learning and optimization bounds. First, the LinUCB estimator achieves a sublinear expected regret of $\tilde{O}\left(d\sqrt{T}\right)$ over T slots, ensuring rapid convergence to the true dynamic data value. Second, the strictly convex barrier function $\mathcal{B}(w,\tau)$ guarantees convergence to the continuous global optimum as $\tau \to 0$. Finally, unbiased probabilistic rounding ensures the discrete caching strategy tightly approximates this optimum under physical uncertainties. By decoupling the discrete topology into lightweight learning and polynomial-time convex relaxation, OLCRA successfully balances real-time efficiency with near-optimal performance.

In conclusion, OLCRA comprehensively solves the VF-CEDC problem. By decoupling the discrete topology into lightweight online learning and polynomial-time convex relaxation, it balances real-time efficiency with near-optimal caching performance, validating our framework.

5 Experiments

5.1 Experimental Setup

To evaluate OLCRA, we conduct extensive simulations on the real-world Shanghai Telecom dataset [18], denoted as SHH Telecom. System parameters are dynamically generated to emulate edge servers, heterogeneous data requests, and physical conditions, as detailed in **Table 1**. For scalability evaluation, the number of data categories |D| is set to 500, 800, and 1000.

Table 1. Experimental Parameter Configurations

Parameter	Value Range
Number of edge servers	[10, 20]
Storage capacity threshold C_i	[60, 150] GB
Physical size of data replica S_k	[500, 2000] MB
Local hit transmission delay $L_{i,i}$	[5, 10] ms
Edge collaborative delay $L_{i,j}$	[15, 30] ms
Cloud backhaul penalty delay $D_{\{cloud\}}$	200 ms
Contextual feature dimension d, Semantic priority ρ_k	8, U(1,10)
Zipf distribution skewness γ, Timeliness threshold Γ_{th}	[0.8, 1.2], [10, 50] s
Request arrival rate $\lambda_{i,k}(t)$	[10, 200] times/slot
Value preference weight ω, Utility parameters (μ, ν, ψ)	0.5, (10^{-3}, 1.2, 2.0)
Server survival probability ($\alpha = 0.6, p = 2, \varsigma = 0.7$)	[0.85, 0.99]
Link survival probability ($\gamma_{th} = 10$dB, shortest path)	[0.90, 0.99]
Ridge regression regularization κ	0.1

5.2 Baselines and Evaluation Metrics

To demonstrate the superiority of OLCRA, we compare it against four baselines, four state-of-the-art (SOTA) methods, and two ablation variants:

- ***Random***: Randomly generates edge data caching and request allocation strategies.
- ***LFU*** and ***LRU***: Traditional baselines relying on historical frequency and temporal recency, which lack considerations for cloud-edge synergy and dynamic value.
- ***Greedy***: Greedily allocates cache space to minimize expected delay, neglecting value differences and physical failure risks.
- ***FDRL*** [8]: Employs federated reinforcement learning to adaptively fit continuous caching strategies for time-varying requests.
- ***DCCC*** [19]: Achieves load balancing via multi-level proxies and consistent hashing.
- ***RoCoCache*** [9]: Utilizes discrete variational autoencoders and robust federated deep learning for dynamic edge cache allocation.
- ***uEDC-L*** [7]: Adopts uncertainty sets for server failures and linear decision rules to provide robust approximate caching solutions.
- ***OLCRA-NV*** and ***OLCRA-NF***: Ablation variants omitting the LinUCB-based dynamic value inference and the physical failure-awareness modules, respectively.

The performance is evaluated using four core metrics: Value-Aware Cost (VAC) defined as the objective function in Eq. (10) (reported in 10^4 scale), Average Expected Delay (AED), Cache Hit Ratio (CHR), and CPU Time (CT) representing the computation delay per time slot.

5.3 Performance Comparison

Table 2 summarizes the performance across four metrics. The global best and second-best results are highlighted in dark and light gray, respectively.

Table 2. Performance Comparison on Shanghai Telecom Dataset

Method	SHH Telecom @500				SHH Telecom @800				SHH Telecom @1000			
	VAC	AED	CHR	CT	VAC	AED	CHR	CT	VAC	AED	CHR	CT
Random	5.845	145.6	0.326	4.553	14.52	162.3	0.283	5.271	25.68	175.8	0.245	6.035
LFU	4.231	95.43	0.553	8.235	11.56	112.5	0.514	9.536	21.25	128.4	0.479	11.45
LRU	4.352	98.25	0.537	6.857	11.84	115.6	0.495	8.419	21.86	131.2	0.454	9.726
Greedy	3.658	85.16	0.651	12.63	10.25	98.48	0.587	15.74	18.56	115.3	0.528	18.82
FDRL	3.125	68.37	0.763	245.1	9.234	82.57	0.711	286.6	16.85	95.61	0.651	325.7
DCCC	3.054	65.25	0.785	115.8	8.956	79.43	0.739	142.3	16.52	92.15	0.687	168.2
RoCoCache	2.842	61.59	0.823	315.7	8.571	75.61	0.785	368.9	15.85	88.47	0.731	412.8
uEDC-L	2.915	58.43	0.814	45.85	8.532	71.29	0.764	58.62	15.68	85.30	0.716	75.92
OLCRA-NV	3.452	55.45	0.840	18.59	9.854	70.83	0.790	22.52	17.65	84.79	0.740	26.68
OLCRA-NF	2.713	68.50	0.774	24.98	8.427	88.20	0.729	29.93	15.23	105.4	0.684	34.83
OLCRA	2.384	48.50	0.910	25.62	7.125	62.40	0.862	30.77	13.55	74.50	0.820	35.91

Compared to heuristic methods, OLCRA consistently achieves the lowest VAC and AED across all scales. Overall, it reduces VAC and AED by an average of 32.3% and 38.3% relative to all evaluated baselines. Specifically, in the SHH Telecom@500 scenario, it reduces both metrics by up to 59.2% and 66.7%. While traditional baselines neglect physical failures and remain vulnerable to link disconnections, OLCRA strategically caches valuable data on reliable servers, maintaining a CHR between 0.82 and 0.91. Although Random and LRU exhibit minimal CT, ignoring data value and server status induces heavy cloud backhaul penalties, yielding the worst VAC.

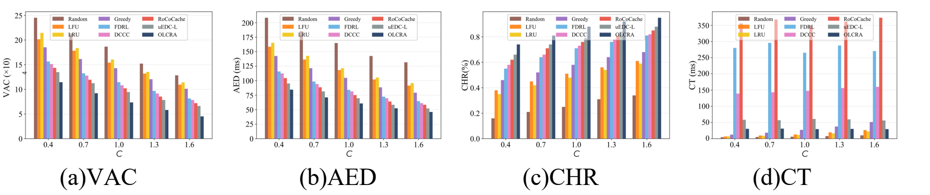
Compared to SOTA learning-based methods (e.g., FDRL, DCCC, RoCoCache, and uEDC-L), OLCRA consistently achieves superior VAC and AED with minimal computational overhead. While DNN-based approaches such as FDRL and RoCoCache incur substantial inference delays of 325.7 ms and 412.8 ms respectively in the SHH Telecom@1000 scenario, OLCRA completes the optimization in only 35.91 ms. This represents a remarkable 91.3% reduction in CT relative to RoCoCache, demonstrating the exceptional efficiency of OLCRA in delivering highly robust and real-time caching services for large-scale edge environments.

Ablation results further verify the effectiveness of OLCRA. OLCRA-NV significantly increases VAC because low-value popular data may occupy limited storage, while OLCRA-NF leads to higher AED because critical data may be placed on unreliable edge nodes and thus trigger frequent cloud backhaul. These results demonstrate the importance of jointly considering data value perception and physical failure awareness in edge caching optimization.

5.4 Performance Impact of Parameters

We further investigate the impact of key parameters on the system performance under the SHH Telecom@800 scenario.

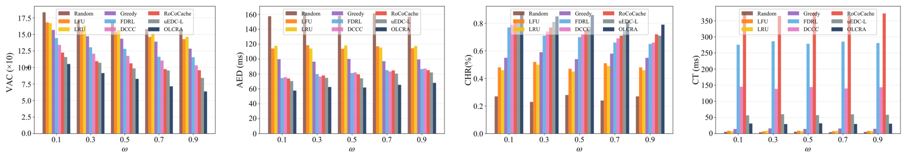
Impact of Edge Server Capacity (C). As shown in Fig. 2, increasing capacity C inflates the CT of heuristics like Greedy due to expanded replica searches. Meanwhile, DRL methods like FDRL and RoCoCache suffer severe CT fluctuations from inherent sampling randomness. Remarkably, the CT of OLCRA remains highly stable at around 30 ms. This stability arises because the dimensionality of the Hessian matrix in OLCRA's interior-point method is strictly bounded by $|\mathcal{V}|$ and $|\mathcal{D}|$, which is perfectly decoupled from the physical capacity C. This mechanism ensures OLCRA's robust real-time scheduling capability.



(a)VAC (b)AED (c)CHR (d)CT

Fig. 2. Performance comparison under different capacity limits (C).

Impact of Value Preference Factor (ω). Fig. 3(a) illustrates that a higher ω prioritizes business value over transmission latency. While traditional heuristics remain static, OLCRA intentionally evicts high-frequency low-value data to accommodate rare yet critical high-value replicas. As shown in Fig. 3(b) and (c), this value-first strategy consequently sacrifices overall CHR and increases AED. Nevertheless, through failure-aware multi-hop collaboration, OLCRA mitigates extreme delays and maintains the lowest VAC across all ω settings to achieve an optimal trade-off between latency and business utility in complex edge environments.



(a)VAC (b)AED (c)CHR (d)CT

Fig. 3. Performance comparison under different value preference weights (ω).

6 Conclusion

This paper proposes an intelligent collaborative edge data caching approach to address the complexities of highly dynamic and heterogeneous MEC environments. To efficiently solve the VF-CEDC problem without prior knowledge of data utility, we develop an innovative two-stage algorithm, OLCRA, which seamlessly integrates contextual bandits with convex optimization. Specifically, OLCRA leverages the LinUCB method to adaptively infer multi-dimensional business values through continuous environment interactions. Furthermore, it utilizes convex relaxation coupled with a primal-dual interior-point method to formulate robust, low-latency caching and routing decisions under dual physical failure uncertainties. Extensive experiments on a real-world dataset validate the superiority of this intelligent framework over eight competing methods, demonstrating its exceptional efficiency, robustness, and potential for reliable service provisioning in future edge networks.

Acknowledgments. This work was supported by National Natural Science Foundation of China (No. 62272290, 62572114).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials **19**(4), 2322–2358 (2017)
2. Li, H., Sun, M., Xia, F., Xu, X., Bilal, M.: A survey of edge caching: Key issues and challenges. Tsinghua Science and Technology **29**(3), 818–842 (2024)

3. Xia, X., Chen, F., He, Q., Cui, G., Lai, P., Abdelrazek, M., Grundy, J., Jin, H.: Graph-Based Optimal Data Caching in Edge Computing. In: International Conference on Service-Oriented Computing (ICSOC). pp. 477–493 (2019)
4. Chu, W., Zhang, X., Jia, X., Lui, J.C., Wang, Z.: Online optimal service caching for multi-access edge computing: A constrained multi-armed bandit optimization approach. Computer Networks **246**, 110395 (2024)
5. Liu, L., Kang, L., Li, X., Zhou, Z.: Service reliability based on fault prediction and container migration in edge computing. Applied Sciences **13**(23), 12865 (2023)
6. Ma, Y., Zhao, H., Guo, K., Xia, Y., Wang, X., Niu, X., Zhu, D., Dong, Y.: A fault-tolerant mobility-aware caching method in edge computing. Computer Modeling in Engineering & Sciences **140**(1), 907–927 (2024)
7. Luo, R., Zhang, Z., He, Q., Xu, M., Chen, F., Dai, X., Wu, S., Jin, H.: Cost-Effective Edge Data Caching With Failure Tolerance and Popularity Awareness. IEEE Transactions on Mobile Computing **24**(6), 5357–5369 (2025)
8. Sun, C., Li, X., Wen, J., Wang, X., Han, Z., Leung, V.C.M.: Federated deep reinforcement learning for recommendation-enabled edge caching in mobile edge-cloud computing networks. IEEE Journal on Selected Areas in Communications **41**(3), 690–705 (2023)
9. Chen, Z., Liang, J., Yu, Z., Cheng, H., Min, G., Li, J.: Resilient collaborative caching for multi-edge systems with robust federated deep learning. IEEE Transactions on Networking **33**(2), 654–669 (2025)
10. Sun, Q., Wu, S., Xv, Y., Chen, J., Sun, K.: Optimization of multi-cell edge caching strategies based on trial and error multi-armed bandit algorithm. In: IEEE International Conference on Communications, Information System and Computer Engineering (CISCE). pp. 216–220 (2025)
11. Wang, L., Wang, Y., Yu, Z., Xiong, F., Ma, L., Zhou, H., Guo, B.: Similarity caching in dynamic cooperative edge networks: An adversarial bandit approach. IEEE Transactions on Mobile Computing **24**(4), 2769–2782 (2025)
12. Qiu, Y., Liang, J., Leung, V.C.M., Wu, X., Deng, X.: Online reliability-enhanced virtual network services provisioning in fault-prone mobile edge cloud. IEEE Transactions on Wireless Communications **21**(9), 7299–7313 (2022)
13. He, Q., Zhang, G., Wang, J., Luo, R., Dai, X., Hu, Y., Chen, F., Jin, H., Yang, Y.: EdgeHydra: Fault-tolerant edge data distribution based on erasure coding. IEEE Transactions on Parallel and Distributed Systems **36**(1), 29–42 (2025)
14. Mandelbrot, B.: An informational theory of the statistical structure of language. In: Communication Theory, pp. 486–502 (1953)
15. Kaul, S., Yates, R., Gruteser, M.: Real-time status: How often should one update? In: 2012 Proceedings IEEE INFOCOM. pp. 2731–2735 (2012)
16. Lee, M.C., Molisch, A.F.: Optimal delay-outage analysis for noise-limited wireless networks with caching, computing, and communications. IEEE Transactions on Wireless Communications **22**(2), 1417–1431 (2023)
17. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World Wide Web. pp. 661–670 (2010)
18. Wang, S., Guo, Y., Zhang, N., Yang, P., Zhou, A., Shen, X.: Delay-Aware Microservice Coordination in Mobile Edge Computing: A Reinforcement Learning Approach. IEEE Transactions on Mobile Computing **20**(3), 939–951 (2021)
19. Wang, T., Deng, Y., Mao, J., Chen, M., Liu, G., Di, J., Li, K.: Towards intelligent adaptive edge caching using deep reinforcement learning. IEEE Transactions on Mobile Computing **23**(10), 9289–9303 (2024)