POI-Based Edge Service Deployment With Topology-Aware Optimization

Guobing Zou[1], Mengjia Yang[1], Song Yang[1], Shengye Pang[1*], Sen Niu[2], Yanglan Gan[3*], Bofeng Zhang[2]
[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
[2] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China
[3] School of Computer Science and Technology, Donghua University, Shanghai, China
{gbzou, mjyang, yangsong, pangsy}@shu.edu.cn, {senniu,bfzhang}@sspu.edu.cn, ylgan@dhu.edu.cn

2025 IEEE International Conference on Web Services (ICWS) | 979-8-3315-5563-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICWS67624.2025.00095

***Abstract*—Edge service deployment has attracted significant attention in recent years, aiming to optimize service placement on edge servers while satisfying diverse requirements. However, existing approaches often overlook the influence of geographical contexts on service demands, where user needs vary significantly across regions with distinct characteristics. They also fail to account for differences between direct responses and multi-hop forwarding in edge network topology, leading to unsatisfactory edge service deployment strategies. To this end, we formulate the Points of Interest-Based Edge Service Deployment (POI-ESD) problem with topology-aware optimization, integrating POI attributes and spatial distributions while incorporating edge network topology to enhance service placement. By proving the $\mathcal{NP}$-hardness of POI-ESD problem, we propose a novel graph-encoded genetic algorithm, MTGA, to efficiently generate high-quality deployment strategies. It ensures strategic placement of edge services in regions that best match user demands, improving the service utilization and satisfiability for edge users. Extensive experiments on a real-world dataset combining Shanghai Telecom and Baidu Maps POI data demonstrate that MTGA significantly outperforms existing competing approaches, achieving superior performance of edge service deployment.**

***Index Terms*—Service Computing, Edge Service Deployment, Points of Interest, Edge Topology, Graph-encoded Genetic Algorithm**

I. Introduction

With the unprecedented growth of Internet of Things (IoT) devices and the rapid advancement of 5G/B5G technologies, the digital landscape has witnessed an explosive surge in data consumption and computation demands [1]. The proliferation of latency-sensitive and computation-intensive applications [2], such as autonomous driving, virtual reality (VR) [3], and interactive gaming, has posed significant challenges to traditional cloud computing paradigms [4]. In the conventional cloud-centric architecture, all service requests must be transmitted to and processed at remote data centers through long-distance multi-hop transmissions, leading to high service latency and network congestion [5]. This centralized approach has become seriously inadequate for supporting the increasing demands of real-time processing and interactive applications that require instant response and reliable service quality [6].

Edge computing extends cloud capabilities to the network edge, emerging as a promising distributed paradigm [7]. In this paradigm, edge servers are built at cellular base stations to provide computational and storage resources close to end users [8]. These edge servers are tactically placed across different regions, with overlapping coverage to ensure seamless service provision. Users submit service requests to nearby edge servers via radio access networks for local processing [9], rather than forwarding them to remote cloud centers. By eliminating long-distance data transmissions on the core network, it significantly reduces response latency [10]. Moreover, this localized processing mechanism substantially decreases backhaul network traffic, effectively alleviating network congestion and enhancing overall system efficiency.

With the increasing diversity of edge services and resource demands, the Edge Service Deployment (ESD) problem has become crucial for ensuring efficient and low-latency service provisioning [11]. Existing research has approached this challenge from multiple perspectives. He et al. [12] constructs integer linear programming model considering resource shareability and develops maximum flow-based approximation algorithms to maximize edge-served users. Liu et al. [5] leverages multi-agent deep reinforcement learning for collaborative service placement optimization in latency-sensitive scenarios. Bi et al. [13] develops Lagrangian dual decomposition algorithms to jointly optimize service quality and operational costs through iterative linear relaxation. OuYang et al. [14] proposes contextual multi-armed bandit learning for adaptive deployment based on user preferences, while Jia et al. [15] designs priority-based strategies with improved Hungarian algorithms for joint data-service placement optimization. Fan et al. [16] further explores the collaborative optimization of placement, scheduling and ·resource allocation.

Although existing researches have explored edge service deployment from various perspectives, two key aspects require further exploration. First, geographical contexts and their impact on service demands are often overlooked. In real world, different geographic regions have distinct functional attributes [17], [18] and POIs shaping diverse user demands. However, current investigations often simulate user distribution via probabilistic models [19], [20], leading to mismatches between deployed services and potentially actual user demands. Second, while edge network topology has been considered for deployment optimization [21], most studies neglect the

* Corresponding authors.

2836-3868/25/$31.00 ©2025 IEEE
DOI 10.1109/ICWS67624.2025.00095

differential influences of multi-hop forwarding among edge servers. Increased hops introduce longer distances, causing higher response delays and degraded user experiences due to more required network communication costs.

To tackle the two issues mentioned above, this paper introduces POIs as geographical contexts, considering POI-based matching benefit and topology-aware decay benefit to define the POI-based edge service deployment problem. Based on the formulated POI-ESD problem, we propose a novel graph-encoded Genetic Algorithm (MTGA) to efficiently derive high-quality edge service deployment strategy. Extensive experiments are carried out on real-world dataset to demonstrate the edge service deployment performance of our approach.

The primary contributions of this paper are as follows:

- We formulate the POI-ESD problem to enhance the functional alignment between edge servers and deployed services by analyzing the attributes and distribution of covered POIs as well as the impacts of edge network topology to optimize both POI-based matching benefit and topology-aware decay benefit.
- We transform the POI-ESD problem to a multi-objective optimization one with $\mathcal{NP}$-hardness theoretical proving. To better solve a POI-ESD problem, we propose an improved Genetic Algorithm, MTGA, which adopts a graph-based representation to encode deployment strategies and integrates POI relevance and edge topological connectivity as heuristic guidance in genetic operations.
- Extensive experiments on a real-world dataset combining Shanghai Telecom and Baidu Maps POI data validate the effectiveness of the POI-ESD problem modeling in edge computing and demonstrate the superior performance of MTGA compared to baseline and state-of-the-art competing approaches.

The remainder of this paper is structured as follows. Section II presents a motivating example and analyzes the challenges of edge service deployment. Section III formulates and models the POI-ESD problem. Section IV proposes a graph-encoded genetic algorithm MTGA. In Section V, comparative experiments are conducted to demonstrate the performance of the proposed approach. Section VI reviews the related works. Finally, we conclude the paper and point out future work in Section VII.

II. Motivating Example

Figure 1 illustrates a typical edge service deployment scenario in an urban edge computing environment. This scenario consists of multiple edge servers $\{v_0, v_1, \ldots, v_7\}$ distributed across different regions, each with a limited coverage represented by dashed circles. These servers are interconnected by high-speed links, shown as blue lines, enabling them to forward service requests to other edge servers where the required services are deployed. Points of different colors represent various types of POIs, such as *Residences*, *Shopping Malls*, *Offices*, *Attractions*, and *Hospitals*, distributed across different areas of the city.

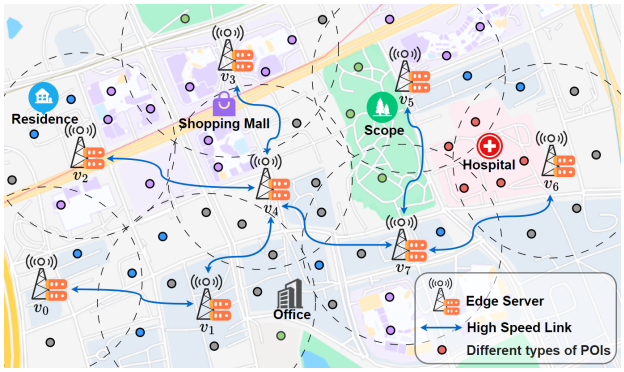


Fig. 1. A motivating example of the POI-ESD problem, illustrating multiple edge servers with limited coverage areas (dashed circles), interconnected by high-speed links (blue lines), and Points of Interest (points in different colors).

Different edge servers have distinct geographical contexts due to their locations, specifically reflected in the number and types of POIs within their coverage areas. For instance, server v_3 predominantly covers *Shopping Mall* POIs, where users are primarily concerned with shopping and entertainment, resulting in a higher likelihood of requests for payment services or location-based recommendations. In contrast, although server v_1 covers relatively more POIs, most are of the *Office* type, making payment services on it unlikely to be frequently invoked and resulting in low resource utilization. Therefore, aligning service deployment with the functional attributes and demand patterns of the covered POIs is more important than simply considering the extent of the coverage area.

Due to resource constraints and limited deployment budgets, services can only be deployed on a subset of edge servers. However, the edge network topology enables servers to forward requests to others where the required services are deployed. If budget constraints prevent the deployment of payment service on v_2 or v_3, placing it on v_4 can serve users near v_2 and v_3 via one-hop forwarding, and reach v_5 through two-hop. Nevertheless, multi-hop forwarding may introduce challenges such as increased response latency and instability. For example, if relay node v_7 fails or experiences congestion while handling requests from v_5, it may degrade user experience. These issues worsen with longer forwarding paths. Therefore, considering both the advantages and challenges of network topology is essential for optimizing deployment strategies and achieving better performance.

III. System Model

A. Edge Computing Environment Modeling

In edge service deployment scenarios that consider geographical contexts, we focus on the edge servers, edge network topology, Points of Interest and edge service. The related definitions and constraints are detailed below, and the notations used in this paper are summarized in Table I.

Definition 1 (Edge Servers). Edge servers are denoted as a set $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$. Each edge server is represented by a tuple $v_i = \langle lat_i, lon_i, P_i, c_i, b_i \rangle$, where lat_i and lon_i

TABLE I
NOTATIONS

Notation	Description
$\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$	Set of edge servers
$v_i = \langle \text{lat}_i, \text{lon}_i, P_i, c_i, b_i \rangle$	Edge server with latitude, longitude, transmission power, resource capacity and deployment cost
$\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$	Set of Points of Interest (POIs)
$p_j = \langle \text{lat}_j, \text{lon}_j, l_j, n_j \rangle$	POI with latitude, longitude, label and name
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Edge network topology graph
$s = \langle t, d, \omega, \mathcal{B} \rangle$	Service with type, description, resource requirement and budget
$RSSI_{i,j}$	Signal strength between v_i and p_j
$PL_{i,j}$	Path loss between v_i and p_j
$d_{i,j}$	Distance between v_i and p_j
$c_{i,j}$	Binary indicator whether POI p_j is within coverage radius of server v_i
$r(p, s)$	Relevance score between POI p and service s
$\mathcal{P}_i$	Set of POIs directly covered by v_i
$\mathcal{P}_i^{(k)}$	k-th ranked POI in $\mathcal{P}_i$ based on relevance score
$w_i^{(k)}$	Weight assigned to k-th ranked POI of server v_i
$\mathcal{N}_i^k$	Set of k-hop neighbor servers of v_i
$\mathcal{A}_i^k$	Set of POIs accessible from v_i through k hops
a_j^k	Whether p_j can access any service deployed via exactly k hops
γ	Benefit decay factor per hop in service sharing
h_T	hop count threshold allowed for service sharing

denote its geographical coordinates, P_i is the transmission power, c_i represents the resource capacity, and b_i indicates the deployment cost.

Definition 2 (Edge Network Topology). The edge network topology is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of edge servers and $\mathcal{E}$ denotes the set of high-speed links between them. Each high-speed link $e \in \mathcal{E}$ connects two edge servers (v_j, v_k), enabling direct communication and resource coordination.

Definition 3 (Points of Interest). Points of Interest (POIs) are represented by a set $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$. Each POI $p_j \in \mathcal{P}$ is depicted by a tuple $p_j = \langle \text{lat}_j, \text{lon}_j, l_j, n_j \rangle$, where lat_j and lon_j denote the geographical coordinates, l_j indicates the POI type, and n_j represents its specific name.

POIs provide important geographical contexts through their locations (lat_j, lon_j) and types l_j. By describing the locations of various facilities in the real world, such as restaurants, schools, and hospitals, POIs can reflect the relatively stable spatiotemporal characteristics of potential user demands [7].

Definition 4 (Edge Service). Given an edge service to be deployed, it is represented by a tuple $s = \langle t, d, \omega, \mathcal{B} \rangle$, where t denotes the service category tag, d represents the functional description, ω indicates the resource requirement, and $\mathcal{B}$ represents the deployment budget.

Different from cloud centers with wired access to the core network, edge servers rely on base stations for wireless coverage. To ensure reliable communication between edge servers and users around POIs, we characterize their coverage relationship based on the signal strength received by user devices. As discussed in [22], the Free Space Path Loss (FSPL) model [23] can be used to model the process of wireless transmission attenuation in outdoor scenarios with minimal obstacles. The coverage relationship between an edge server p_j and a POI v_i is determined by comparing the received signal strength $RSSI_{i,j}$ with a sensitivity threshold θ:

$$c_{i,j} = \begin{cases} 1 & \text{if } RSSI_{i,j} \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $RSSI_{i,j}$ is calculated by considering the transmission power P_i, antenna gains (G_t for transmitter and G_r for receiver), and the path loss $PL_{i,j}$ in free space:

$$RSSI_{i,j} = P_i + G_t + G_r - PL_{i,j} \tag{2}$$

Following the FSPL model, the path loss $PL_{i,j}$ between v_i and p_j is determined by the Euclidean distance $d_{i,j}$ calculated by their geographical coordinates, the carrier frequency f (2.4 GHz in this paper), and the speed of light c:

$$PL_{i,j} = 20 \log_{10} \left(\frac{4\pi d_{i,j} f}{c} \right) \tag{3}$$

B. POI-Based Matching Benefit Model

Edge service deployments aim to locate services closer to users, thereby delivering low latency and high-quality experiences. Since user demands are strongly tied to geographical contexts, POIs play a crucial role as they carry rich local environmental information and reflect potential user demand distributions through their diverse functional attributes. However, there is currently no standard method to quantify the correlation between service characteristics and geographic context [24]. To address this, we propose to quantify the correlation by computing the semantic similarity between the textual descriptions of services and POIs. Leveraging SBERT [25], we embed these semantic descriptions into a dense vector space. For a POI p with type label l and name n, and a service s with category tag t and functional description d. Their embeddings are computed as:

$$e_p = \text{SBERT}(\text{concat}(l, n)), e_s = \text{SBERT}(\text{concat}(t, d)) \tag{4}$$

The relevance score $r(p, s)$ between p and s is then calculated using cosine similarity:

$$r(p, s) = \frac{e_p \cdot e_s}{\|e_p\| \|e_s\|} \tag{5}$$

Since each edge server covers a distinct set of POIs, these scores could be aggregated to assess how well a server aligns with the service's functional requirements. POIs with higher

semantic relevance to a service naturally attract more potential users and generate higher service demands. Therefore, we introduce a ranking-weighted calculation that assigns higher importance to more relevant POIs, ensuring the deployment benefit better reflects each server's potential value.

Let $\mathcal{P}_i = \{p_j \in \mathcal{P} \mid c_{i,j} = 1\}$ denote the set of POIs covered by edge server v_i. Each POI $p_j \in \mathcal{P}_i$ has an associated relevance score $r(p_j, s)$ with respect to service s. To reflect the varying contributions of different POIs, we rank the POIs they cover in descending order of relevance scores, where $\mathcal{P}_i^{(k)}$ denotes the k-th ranked POI, such that $r(\mathcal{P}_i^{(j)}, s) \geq r(\mathcal{P}_i^{(k)}, s)$ for any $j < k$. To emphasize the impact of the higher-ranked POIs, we assign the following weights to the ranked POIs:

$$w_i^{(k)} = |\mathcal{P}_i| - k + 1 \tag{6}$$

Definition 6 (POI-based Matching Benefit). Given an edge server v_i and its ranked POIs $\{\mathcal{P}_i^{(1)}, \mathcal{P}_i^{(2)}, \ldots, \mathcal{P}_i^{(|\mathcal{P}_i|)}\}$, the matching benefit of v_i for service s is defined as:

$$\mathcal{M}(v_i, s) = \frac{\sum_{k=1}^{|\mathcal{P}_i|} w_i^{(k)} \cdot r(\mathcal{P}_i^{(k)}, s)}{|\mathcal{P}_i|(|\mathcal{P}_i| + 1)}. \tag{7}$$

where the denominator normalizes the sum to allow fair comparison across servers with different POI densities.

Given a deployment strategy $\mathcal{V}_s \subseteq \mathcal{V}$ that selects a subset of edge servers, the overall matching benefit is calculated as the average matching benefits across the selected servers:

$$\mathcal{M}(\mathcal{V}_s, s) = \frac{1}{|\mathcal{V}_s|} \sum_{v_i \in \mathcal{V}_s} \mathcal{M}(v_i, s) \tag{8}$$

C. Topology-Aware Decay Benefit Model

Edge servers are interconnected through network topology that extends service accessibility beyond their immediate coverage through multi-hop forwarding. However, as the number of hops increases, service quality inevitably degrades due to accumulated transmission delays, network congestion, and potential server failures [26]. To capture these effects, we propose a Topology-Aware Decay Benefit Model, which quantifies the diminishing service quality as requests traverse through additional hops. Specifically, a decay factor $\gamma \in [0, 1]$ is introduced to synthesize various network effects including signal attenuation, queuing delays, and link reliability, ensuring a comprehensive evaluation of multi-hop service degradation.

For an edge server $v_i \in \mathcal{V}$, let $\mathcal{N}_i^k$ denote the set of servers that can cooperatively share services with v_i through k hops of high-speed links. Specifically, the first-hop neighboring servers are given by $\mathcal{N}_i^1 = \{v_j \mid (v_i, v_j) \in \mathcal{E}\}$. Based on these server connections, we define the set of POIs that are reachable from v_i through exactly k hops, denoted as $\mathcal{A}_i^k$. For $k = 0$, $\mathcal{A}_i^0$ represents the set of POIs $\mathcal{P}_i$ directly covered by v_i. For $k > 0$, $\mathcal{A}_i^k$ consists of POIs covered by v_i's k-hop neighboring servers $\mathcal{N}_i^k$ but not reachable through fewer hops:

$$\mathcal{A}_i^k = \begin{cases} \mathcal{P}_i & k = 0 \\ \left(\bigcup_{v_m \in \mathcal{N}_i^k} \mathcal{P}_m\right) \setminus \bigcup_{l=0}^{k-1} \mathcal{A}_i^l, & k > 0 \end{cases} \tag{9}$$

Definition 7 (Topology-aware Decay Benefit). Given an edge server v_i and a hop count threshold h_T, the topology-aware decay benefit of v_i is defined as:

$$\mathcal{T}(v_i, s) = \sum_{k=1}^{h_T} \sum_{p_j \in \mathcal{A}_i^k} r(p_j, s) \cdot \gamma^k \tag{10}$$

where γ^k accounts for the cumulative decay over k hops of transmission.

For a deployment strategy $\mathcal{V}_s \subseteq \mathcal{V}$, each POI may access the deployed service through multiple paths in the edge network topology. We consider the maximum achievable benefit for each POI through its optimal access path. Therefore, topology-aware decay benefit of $\mathcal{V}_s$ is defined as:

$$\mathcal{T}(\mathcal{V}_s, s) = \sum_{p_j \in \mathcal{P}} r(p_j, s) \cdot \max_{k=1}^{h_T} \{\gamma^k \cdot a_j^k\} \tag{11}$$

where $a_j^k \in \{0, 1\}$ indicates whether POI p_j can access a service deployed on any server through exactly k hops:

$$a_j^k = \mathbb{I}\left(\sum_{v_i \in \mathcal{V}_s} |\mathcal{A}_i^k \cap \{p_j\}| > 0\right), \quad \forall p_j \in \mathcal{P},\, 1 \leq k \leq h_T \tag{12}$$

where $\mathbb{I}(\cdot)$ is an indicator function that equals 1 if the condition inside holds true, and 0 otherwise.

D. POI-ESD Problem Formulation

Definition 8 (POI-based Edge Service Deployment). A POI-ESD problem can be defined as a six-tuple POI-ESD = $\langle \mathcal{V}, \mathcal{P}, \mathcal{G}, s, \gamma, h_T \rangle$, where:

- $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ is a set of edge servers;
- $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ is a set of POIs;
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the edge network topology, where $\mathcal{E}$ is a set of high-speed links between edge servers;
- $s = \langle t, d, \omega, \mathcal{B} \rangle$ represents the service to be deployed;
- γ is the benefit decay factor per hop in service sharing;
- h_T is the hop count threshold allowed for service sharing.

Given a POI-ESD problem, the solution is defined by selecting a subset of edge servers $V_s \subseteq \mathcal{V}$ to deploy services for handling requests from user groups associated with different POIs. The principle of generating an optimized solution is to maximize matching benefits based on POI distribution while aligning service deployment with user demands and optimizing overall decay benefits through edge network topology. To this end, we convert the problem into a multi-objective constrained optimization problem and propose an approximate optimal algorithm in the next section.

IV. Approach

A. POI-ESD Multi-Objective Optimization Transition

Given a *POI-ESD* $= \langle \mathcal{V}, \mathcal{P}, \mathcal{G}, s, \gamma, h_T \rangle$, we transform it as a multi-constrained multi-objective optimization problem: (1) maximizing the average matching benefit that evaluates the relevance-weighted contributions of POIs covered by the selected servers, and (2) maximizing the topology-aware decay

benefit that assesses service sharing capabilities with multi-hop forwarding in edge network topology, while satisfying network accessibility, coverage relationship and deployment budget constraints. The transition from an original POI-ESD problem to its corresponding optimization problem is formulated as follows:

$$\max_{x} \quad \frac{1}{\sum_{v_i \in \mathcal{V}} x_i} \sum_{v_i \in \mathcal{V}} \mathcal{M}(v_i, s) \cdot x_i \tag{13}$$

$$\max_{x} \quad \sum_{p_j \in \mathcal{P}} r(p_j, s) \cdot \max_{k=1}^{h_T} \{\gamma^k \cdot a_j^k\} \tag{14}$$

s.t.

$$a_j^k = \mathbb{I}\left(\sum_{v_i \in \mathcal{V}} |\mathcal{A}_i^k \cap \{p_j\}| \cdot x_i > 0 \right), \quad \forall p_j \in \mathcal{P},\, 1 \le k \le h_T \tag{15}$$

$$a_j^k \in \{0, 1\}, \quad \forall p_j \in \mathcal{P}, 0 \le k \le h_T \tag{16}$$

$$c_{i,j} = \mathbb{I}(RSSI_{i,j} \ge \theta), \quad \forall v_i \in \mathcal{V}, p_j \in \mathcal{P} \tag{17}$$

$$\sum_{v_i \in \mathcal{V}} b_i \cdot x_i \le \mathcal{B} \tag{18}$$

where the decision variable x_i indicates whether an edge service s is deployed on an edge server v_i:

$$x_i = \begin{cases} 1 & \text{if } s \text{ is deployed on } v_i, \\ 0 & \text{if } s \text{ is not deployed on } v_i. \end{cases} \tag{19}$$

The objective function (13) maximizes the average POI-based matching benefit by considering the weighted relevance scores of POIs covered by the selected servers. The objective function (14) optimizes the topology-aware decay benefit through the edge network topology with a hop-based decay factor γ. Constraints (15) and (16) specify the service accessibility relationships and their binary nature through the edge network topology. Constraint (17) defines the basic coverage relationship between edge servers and POIs based on signal strength. Constraint (18) requires that the total cost spent on the servers chosen to deploy the service must not exceed the maximum budget $\mathcal{B}$ that the provider of s can afford.

B. $\mathcal{NP}$ Hardness of POI-ESD Problem

Based on the transformed multi-constrained multi-objective optimization problem, we now prove that the POI-ESD problem is $\mathcal{NP}$-hard. First, we introduce the Maximum Set Coverage Problem, a classic $\mathcal{NP}$-hard problem.

Definition 9 (Maximum Set Coverage Problem). Given a set of elements $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and a set of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, where each subset S_i is a subset of $\mathcal{U}$, the objective is to select a subset of subsets $\mathcal{S}' \subseteq \mathcal{S}$ such that the total number of elements covered by the subsets in $\mathcal{S}'$ is maximized, subject to a constraint on the maximum number of subsets $\mathfrak{n}$. This problem can be formally modeled as follows:

$$\max_{x,z} \quad \sum_{j=1}^{m} z_j \tag{20}$$

s.t.

$$\sum_{i=1}^{n} x_i \le \mathfrak{n} \tag{21}$$

$$z_j \le \sum_{i=1}^{n} \mathbb{I}(u_j \in S_i) \cdot x_i, \quad \forall j \in \{1, 2, \ldots, m\} \tag{22}$$

$$z_j \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, m\} \tag{23}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, 2, \ldots, n\} \tag{24}$$

where z_j is a binary variable indicating whether element u_j is covered by at least one subset in $\mathcal{S}'$, and x_i is a binary variable indicating whether subset S_i is selected.

Theorem 1. *The Maximum Set Coverage Problem is reducible from the POI-ESD problem, thus proving that the POI-ESD problem is $\mathcal{NP}$-hard.*

Proof. To prove the $\mathcal{NP}$-hardness of the POI-ESD problem, we start from the multi-objective optimization problem defined earlier and construct a reduction to the Maximum Set Coverage Problem (MSCP) through the following steps:

- All edge servers have the same deployment cost $b_i = b$. The budget constraint (18) reduces to $\sum_{v_i \in \mathcal{V}} x_i \le \mathfrak{n}$ where $\mathfrak{n} = \lfloor \mathcal{B}/b \rfloor$. This transformed constraint limits the total number of selected servers, which corresponds exactly to the cardinality constraint (21) in MSCP that bounds the number of selected subsets.
- The relevance between all POIs $p_j \in \mathcal{P}$ and service s is set as $r(p_j, s) = 1$. With the normalization factor $|\mathcal{P}_i|(|\mathcal{P}_i|+1)$ in the matching benefit calculation (7), each selected server contributes identical benefit $\frac{1}{2}$ regardless of its covered POIs. Therefore, the objective (13) will also be simplified to $\frac{1}{2}$, becoming a constant value.
- Setting $h_T = 1$ restricts the objective (14) to single-hop service sharing, while $\gamma = 1$ removes decay effects. Under these settings, $\max_{k=1}^{h_T} \{\gamma^k \cdot a_j^k\}$ simplifies to a_j^1. With $r(p_j, s) = 1$, the objective (14) reduces to $\sum_{p_j \in \mathcal{P}} a_j^1$, aligning with the MSCP objective (20) that maximizes the number of elements covered by selected subsets. Constraints (15) and (16) on accessibility variables a_j^k are naturally aligning with the constraint (22) and (23).

Based on the alignment of the above objectives and constraints, a solution to a POI-ESD problem exists if and only if there is a solution to its corresponding MSCP, proving that the POI-ESD problem is $\mathcal{NP}$-hard. □

C. Approximation Algorithm for POI-ESD Problem

The $\mathcal{NP}$-hard and multi-objective nature of the POI-ESD problem poses significant challenges for optimizing edge service deployment due to its high computational complexity that may lead to difficulty in handling large-scale edge servers in real-world service-oriented application scenarios. To address it, we propose **MTGA**, a graph-encoded heuristic **G**enetic **A**lgorithm that simultaneously optimizes POI-based **M**atching benefit and **T**opology-aware decay benefit. Leveraging a custom fitness function, MTGA converts the multi-

Algorithm 1: Graph-encoded Genetic Algorithm for POI-ESD Problem (MTGA)

Input: A POI-ESD problem $\langle \mathcal{V}, \mathcal{P}, \mathcal{G}, s, f, h_T \rangle$
Output: An optimal deployment strategy $\mathbf{x}$

Set parameters N_{pop}, N_{it}, N_{mu}, P_{cr}, and P_{mu};
Initialize N_{pop} heuristic chromosomes $\Gamma = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{N_{\text{pop}}}\}$ as the initial population;
Calculate the fitness of each chromosome $\mathcal{G}_i$ in Γ: $f(\mathcal{G}_i) = \frac{2 \cdot \mathcal{M}(\mathcal{V}_i,s) \cdot \mathcal{T}_{\log}(\mathcal{V}_i,s)}{\mathcal{M}(\mathcal{V}_i,s) + \mathcal{T}_{\log}(\mathcal{V}_i,s)}$;
for $i = 1$ **to** N_{it} **do**
    Create an empty set Γ' to save the populations of the next generation;
    for $j = 1$ **to** $\frac{N_{pop}}{2}$ **do**
        Select two chromosomes $\mathcal{G}_u, \mathcal{G}_v$ by tournament selection;
        if *rand*$(0,1) < P_{cr}$ **then**
            Determine segments of $\mathcal{G}_u$ and $\mathcal{G}_v$ for crossover by topology-aware probability π_u^i, π_v^i;
            Crossover: $\mathcal{G}_u \to \mathcal{G}'_u$; $\mathcal{G}_v \to \mathcal{G}'_v$;
            if *rand*$(0,1) < P_{mu}$ **then**
                Mutation: randomly select N_{mu} loci to replace genes with candidate servers by match-aware probability π_k;
            end
            Put $\mathcal{G}'_u$ and $\mathcal{G}'_v$ into Γ';
        end
        else
            Put $\mathcal{G}_u$ and $\mathcal{G}_v$ into Γ';
        end
    end
    Calculate the fitness of each chromosome $\mathcal{G}'_i$ in Γ';
    $\mathcal{G}_{\text{best}} \leftarrow \arg\max_{\mathcal{G}'_i \in \Gamma'} f(\mathcal{G}'_i)$;
    $\Gamma \leftarrow \Gamma'$;
end
return $\mathbf{x} = \langle x_1, x_2, \ldots, x_m \rangle$ *decoded from* $\mathcal{G}_{best}$

objective problem into tractable single-objective evaluations. The pseudo-code is presented in Algorithm 1.

In MTGA, each feasible solution is encoded as a chromosome based on a graph structure $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, where $\mathcal{V}_i$ represents the set of selected edge servers, and $\mathcal{E}_i$ denotes the high-speed links between these servers. Each node has a Boolean state, indicating whether it is selected to deploy a service. The genetic operations in MTGA are as follows.

1) Initialization of MTGA: At the beginning, MTGA generates N_{pop} chromosomes to form the initial population $\Gamma = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{N_{\text{pop}}}\}$. Rather than relying on random selection, the initialization is guided by heuristic information. Specifically, for each edge server v_j, both the number of covered POIs $|\mathcal{P}_j|$, and the relevance $r(p,s)$ between each POI $p \in \mathcal{P}_j$ and the service s are evaluated. Edge servers with higher POI coverage or greater relevance receive increased selection weights in forming the deployment strategy.

2) Fitness Evaluation of MTGA: To rigorously evaluate deployment strategies, MTGA involves a fitness function based on the harmonic mean, which simultaneously maximizes the POI-based matching benefit $\mathcal{M}(\mathcal{V}_i, s)$ and topology-aware decay benefit $\mathcal{T}(\mathcal{V}_i, s)$, defined as:

$$f(\mathcal{G}_i) = \frac{2\,\mathcal{M}(\mathcal{V}_i, s) \cdot \mathcal{T}_{\log}(\mathcal{V}_i, s)}{\mathcal{M}(\mathcal{V}_i, s) + \mathcal{T}_{\log}(\mathcal{V}_i, s)} \tag{25}$$

Here, $\mathcal{T}_{\log}(\mathcal{V}_i, s)$ denotes the decay benefit after logarithmic transformation. This transformation effectively mitigates the scale disparity between the two objectives, thereby preventing the decay benefit from unduly dominating the evaluation of deployment quality, and is defined as:

$$\mathcal{T}_{\log}(\mathcal{V}_i, s) = \log\big(1 + \mathcal{T}(\mathcal{V}_i, s)\big) \tag{26}$$

3) Selection of MTGA: To ensure that high-quality deployment strategies are propagated to subsequent generations, we first calculate the fitness of each chromosome in the current population, as defined in (25). Subsequently, we employ the well-known *tournament selection* method to execute the selection process, which not only increases the likelihood of choosing superior chromosomes but also preserves population diversity. Specifically, a subset $\Gamma^k = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k\}$ of k chromosomes is randomly selected from the population Γ, and their fitness values are compared. The chromosome with the highest fitness, denoted by $\mathcal{G}_s$, is then chosen for the next generation, as given by (27).

$$\mathcal{G}_s = \arg\max_{\mathcal{G}_i \in \Gamma^k} f(\mathcal{G}_i) \tag{27}$$

4) Crossover in MTGA: For the POI-ESD problem, MTGA introduces a heuristic topological subgraph crossover operation, which leverages the topology-aware decay benefit to improve the quality of edge service deployment solution. Given two chromosomes $\mathcal{G}_u = (\mathcal{V}_u, \mathcal{E}_u)$ and $\mathcal{G}_v = (\mathcal{V}_v, \mathcal{E}_v)$ to be crossed, topological subgraph $\mathcal{G}_u^i = (\mathcal{V}_u^i, \mathcal{E}_u^i)$ is defined for each deployed edge server $v_i \in \mathcal{V}_u$ (or $\mathcal{V}_v$) as the subgraph induced by v_i and its h_T-hop neighbors. The crossover process evaluates the quality of these subgraphs based on their topology-aware decay benefits and selects one from each parent for crossover. Specifically, the quality of a subgraph is defined as:

$$Q(\mathcal{G}_u^i) = \frac{\mathcal{T}(\mathcal{V}_u^i, s)}{|\mathcal{V}_u^i|} \tag{28}$$

where $|\mathcal{V}_u^i|$ represents the number of servers that have deployed service in subgraph $\mathcal{V}_u^i$.

The selection probability π_u^i for a topological subgraph $\mathcal{G}_u^i$ centered at v_i in chromosome $\mathcal{G}_u$ is calculated using a softmax function applied to the quality scores of all subgraphs in $\mathcal{G}_u$:

$$\pi_u^i = \frac{\exp(Q(\mathcal{G}_u^i))}{\sum_{v_k \in \mathcal{V}_u} \exp(Q(\mathcal{G}_u^k))} \tag{29}$$

Chromosome $\mathcal{G}_v$ also undergoes the same process. The selected subgraphs are then exchanged to create two offspring chromosomes which inherit the high-quality subgraphs from their parents.

5) Mutation of MTGA: Although crossover exchanges subgraphs to optimize deployment strategies, it does not directly improve the POI-based matching benefit of individual nodes. Therefore, we introduce a multi-point mutation operation to replace nodes with low POI-based matching benefits with potentially superior undeployed nodes through a heuristic selection process.

Taking chromosome $\mathcal{G}_u = (\mathcal{V}_u, \mathcal{E}_u)$ as an example, in each mutation step, N_{mu} nodes are randomly selected from $\mathcal{V}_u$ as mutation points. For each mutation point v_j, a replacement server is chosen from the set of undeployed servers $\mathcal{V} \setminus \mathcal{V}_u$. The probability π_k of selecting an edge server v_k to replace v_j is determined using a softmax function based on the POI-based matching benefit:

$$\pi_k = \frac{\exp(\mathcal{M}(v_k, s))}{\sum_{v_m \in \mathcal{V} \setminus \mathcal{V}_u} \exp(\mathcal{M}(v_m, s))} \tag{30}$$

D. Computational Complexity Analysis of MTGA

MTGA's complexity primarily stems from evaluating fitness for each individual in the population, together with selection, crossover, and mutation. If the traditional fitness evaluation approach is adopted, each solution is re-evaluated across all servers in every iteration, thus incurring a computational cost of $O(|\Gamma| \cdot |\mathcal{V}| \cdot h_T)$. However, MTGA adopts an incremental evaluation mechanism that updates only newly modified deployments and their local neighborhoods, thereby reducing the main cost to $O(|\Gamma| \cdot c \cdot h_T)$, where c is the number of altered servers per generation.

In addition, because MTGA utilizes a tournament-based selection procedure, each selection operation can be performed in $O(|\Gamma| \cdot k)$, where k denotes the number of individuals randomly drawn from the population in each tournament. Crossover and mutation focus on the $c \ll |\mathcal{V}|$ updated servers and therefore incur $O(|\Gamma| \cdot c \cdot h_T)$ per iteration. Summing these contributions within each generation leads to $O(|\Gamma| \cdot k) + O(|\Gamma| \cdot c \cdot h_T) \approx O(|\Gamma| \cdot c \cdot h_T)$ and, repeated for N_{it} generations, yields $O(N_{it} \cdot |\Gamma| \cdot c \cdot h_T)$ overall.

V. Experiments

A. Experimental Setup and Datasets

All experiments were conducted on a workstation equipped with two NVIDIA GeForce 1080Ti GPUs and an Intel Xeon Gold 6132 CPU running at 2.60 GHz. All competing approaches were implemented using Python 3.12.

To evaluate the effectiveness of the proposed approach, we conducted experiments using data derived from the *Shanghai Telecom dataset* [27] and POI information collected via Baidu Maps API[1]. The dataset comprises locations of 3,233 base stations across Shanghai, while the POI data includes geographic coordinates, categories, and other attributes for locations in the selected experimental regions. Table II provides some samples of the POI data, illustrating the diversity in types and locations. To ensure robust evaluation, three representative regions were chosen, whose statistical details are summarized in Table III.

[1]https://lbsyun.baidu.com/

TABLE II
POI Data Samples in Experimental Scenarios

id	District	Coordinate	Type	Name
1	YangPu	(31.306, 121.525)	Shopping	Heshenghui Plaza
2	HuangPu	(31.223, 121.484)	Residential	Cuihu Tiandi
3	HuangPu	(31.224, 121.479)	Healthcare	Ruijin Hospital
4	PuDong	(31.239, 121.514)	Office	Shanghai World Financial Center
5	PuDong	(31.245, 121.506)	Scope	Oriental Pearl Tower
6	PuDong	(31.241, 121.506)	Hotel	Shangri-La Hotel

TABLE III
Statistical Information on Experimental Datasets

Parameters	Value		
	SHH Telecom@1	SHH Telecom@2	SHH Telecom@3
Location	Wujiaochang	Huaihai Park	Lujiazui
Geographical Range (m)	2000 × 1500	3000 × 2500	5000 × 4000
Number of Servers	27	51	85
Number of POIs	182	321	483

B. Competing Methods and Evaluation Metrics

In the experiments, we evaluate the performance of MTGA against six representative approaches.

- *Random*: It randomly selects edge servers for service deployment until the budget is exhausted.
- *BEAD* [19]: It uses integer linear programming to maximize the distinct POIs covered under budget constraints.
- *RBEAD* [28]: It maximizes the redundancy of POI coverage, ensuring reliability by increasing the number of times each POI is covered.
- *CRBEAD* [20]: It jointly optimizes distinct POI coverage and POI redundancy using the harmonic mean.
- *MTGA-M*: It is a variant of MTGA that emphasizes matching benefit by employing heuristic mutation.
- *MTGA-T*: It is a variant of MTGA that enhances decay benefit using topology-aware crossover operations.

In the experiments, we employ four evaluation metrics to compare and analyze the experiment results:

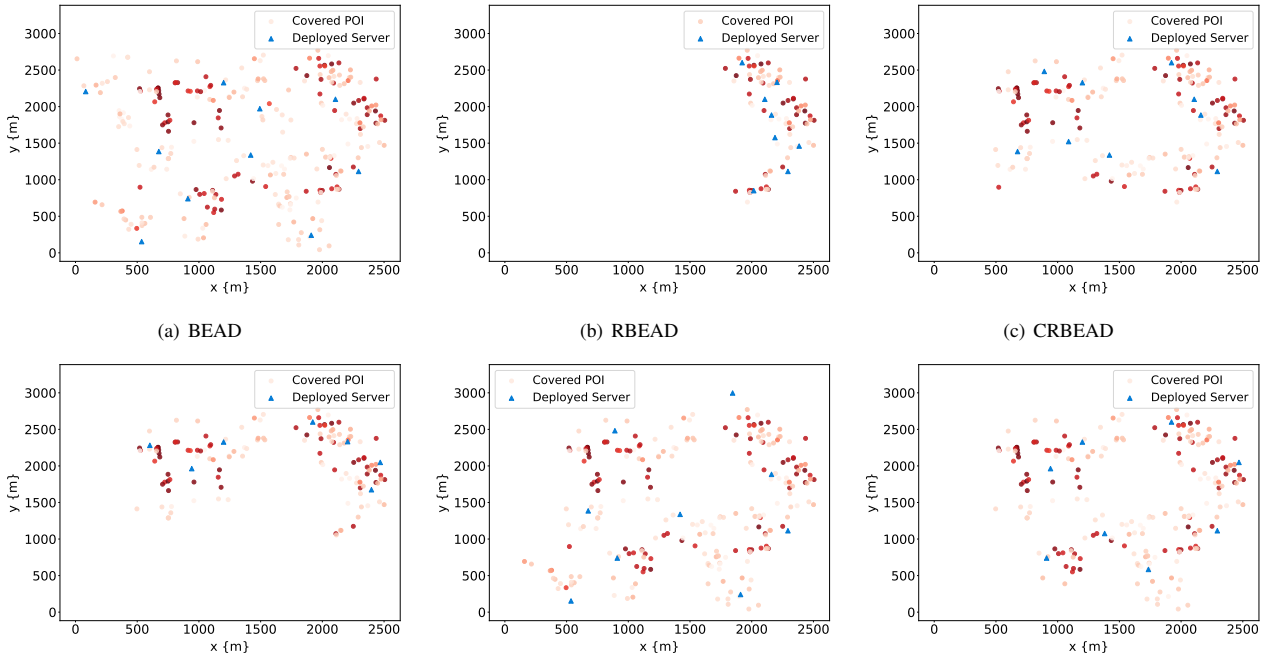
- *Matching Benefit (**MB**)*: It evaluates the average relevance between deployed servers and their directly covered POIs, based on the weighted relevance score of each POI as defined in (8).
- *Topology-aware Decay Benefit (**TB**)*: It aggregates the relevance of POIs reachable via multi-hop paths, decaying with increasing path length as calculated in (11).
- *Global Coverage Benefit (**GCB**)*: It represents the total relevance benefit of POIs directly covered by deployed servers and those reachable through multi-hop, which can be obtained by setting k in (11) start from 0 to h_T.
- *CPU Time*: It is measured by the computation time taken to find an edge service deployment, reported in milliseconds (ms).

C. Experimental Results and Analysis

In the experiments, the hop count $h_T = 2$ enables servers to extend their coverage through intermediary nodes, while the decay factor $\gamma = 0.5$ reflects relevance attenuation over multi-hop connections. For the genetic algorithm, $N_{\text{pop}} = 20$,

TABLE IV
PERFORMANCE COMPARISON OF MTGA AND COMPETING METHODS ON SHANGHAI-TELECOM DATASETS

Algorithm	SHH Telecom@1				SHH Telecom@2				SHH Telecom@3			
	MB	TB	GCB	CPU Time	MB	TB	GCB	CPU Time	MB	TB	GCB	CPU Time
Random	0.6625	8.865	28.52	**0.026**	0.6165	20.21	44.25	**0.034**	0.6501	36.64	122.6	**0.051**
BEAD	0.7160	17.71	39.50	19.86	0.7023	46.81	85.03	38.99	0.7381	69.04	171.4	79.81
RBEAD	0.6455	1.573	21.10	23.24	0.7866	6.44	51.13	60.79	0.6885	8.615	107.9	145.4
CRBEAD	0.7006	3.464	25.71	19.98	0.7011	30.41	77.59	48.23	0.7379	26.33	141.4	100.2
MTGA-M	**0.9366**	5.348	40.17	5.524	**0.9025**	8.75	80.13	7.951	**0.8847**	9.641	174.5	51.60
MTGA-T	0.6895	**22.17**	32.94	55.82	0.6807	**52.99**	79.64	176.8	0.5936	**83.09**	127.0	349.9
MTGA	0.8999	15.34	**49.39**	35.92	0.8694	31.37	**95.85**	72.91	0.8547	55.64	**196.5**	216.9



(a) BEAD (b) RBEAD (c) CRBEAD

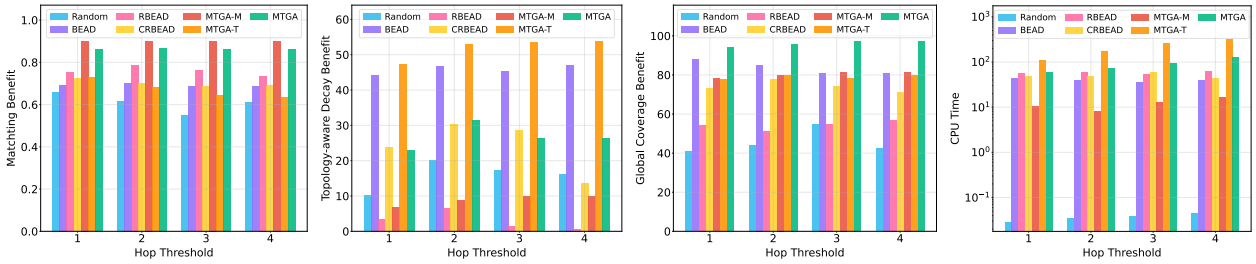
(d) MTGA-M (e) MTGA-T (f) MTGA

Fig. 2. Deployment strategies on the *SHH Telecom@2* dataset. Blue triangles (△) indicate deployed servers, and red points (●) represent covered POIs. Darker red points denote POIs with higher relevance to the deployed service.

$N_{\text{it}} = 50$, $N_{\text{mu}} = 5$, $P_{cr} = 0.8$, and $P_{mu} = 0.4$. Deployment costs of edge servers follow a Gaussian distribution $N(10, 4)$, with a mean of \$10 and a variance of 4. Different services and budgets are specified for each scenario: \$50 for healthcare in *SHH Telecom@1*, \$80 for office services in *SHH Telecom@2*, and \$220 for tourism in *SHH Telecom@3*.

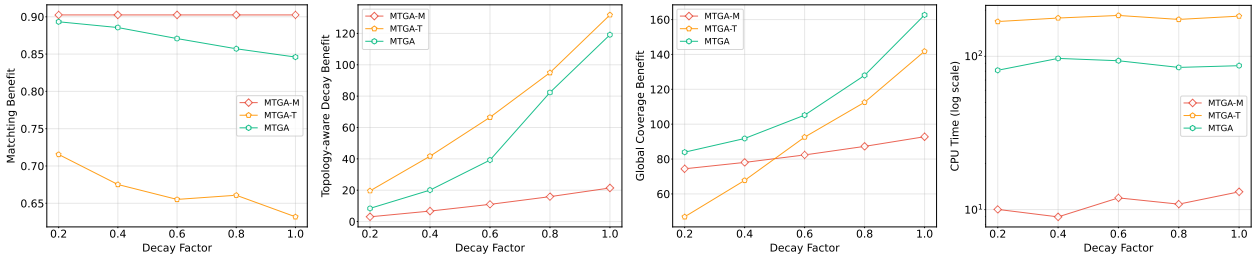
Table IV summarizes the comparative performance of MTGA, the baseline Random method, three existing approaches, and two variants across the three datasets. The best result for each evaluation metric is highlighted in dark grey, and the second-best in light grey. Firstly, compared to the Random method and the existing approaches, MTGA achieves significant improvements across all evaluation metrics on average across the three datasets. Specifically, in terms of MB, MTGA exceeds Random by 36.1%, BEAD by 21.8%, RBEAD by 24.7%, and CRBEAD by 22.8%. For TB, MTGA improves 60.0% over Random and 152.4% over CRBEAD. Although BEAD achieves a higher average TB than MTGA by 29.6%, its MB and GCB are consistently lower than those of MTGA. Regarding GCB, MTGA surpasses Random by 83.4%, BEAD by 17.5%, RBEAD by 101.2%, and CRBEAD by 51.5%. BEAD prioritizes POI coverage but ignores relevance and topology, limiting its deployment effectiveness. Besides, Random consistently exhibits the shortest CPU Time across all datasets. RBEAD, focusing on redundant POI coverage, results in a larger solution space and 51.8% higher CPU Time than BEAD and 29.1% higher than CRBEAD. MTGA has higher runtime due to complexity but is only 41.3% more than RBEAD, staying within an acceptable range.

Nextly, we compare MTGA with its two variants, MTGA-M and MTGA-T. MTGA-M focuses on maximizing MB and achieves the highest MB across all datasets. In the *SHH*



(a) MB (b) TB (c) GCB (d) CPU Time

Fig. 3. Performance comparisons on different hop constraints among MTGA and competing approaches



(a) MB (b) TB (c) GCB (d) CPU Time

Fig. 4. Performance comparisons on different decay factors among MTGA and variants

Telecom@1 dataset, it achieves an MB 4.1% higher than MTGA. However, its weaker network expansion limits TB to 34.9% of MTGA's. Consequently, in terms of GCB, MTGA outperforms MTGA-M by 18.4% on average. MTGA-T prioritizes topology expansion, achieving the highest TB. In *SHH Telecom@3* dataset, it reaches a TB of 83.09, 49.4% higher than MTGA's 55.64. However, MTGA-T's MB is consistently lower than MTGA's across all datasets. In terms of GCB, MTGA's balanced performance gives it a consistent advantage, with an average improvement of 18.4% over MTGA-M and 41.7% over MTGA-T. For CPU Time, MTGA-M focuses on a small number of servers with high POI-based matching benefit, resulting in the shortest runtime. MTGA-T, emphasizing global topology expansion, experiences slower iteration speed. MTGA falls between the two, reducing MTGA-T's time by an average of 44.1% across all datasets, demonstrating its superior performance in balancing both deployment solution generation and computational cost.

Finally, Fig.2 visualizes the deployment strategies of six competing approaches on the *SHH Telecom@2* dataset. Blue triangles indicate the servers selected for service deployment under the budget constraint $\mathcal{B}$, while red dots represent POIs that are directly or indirectly covered within the hop constraint h_T, with deeper red denoting higher relevance between the POI and the deployed service. Among methods from related studies, as shown in Fig.2(a), BEAD adopts a widely distributed strategy that covers the largest number of POIs, though many servers are located in areas with low-relevance POIs, as indicated by lighter red colors. In Fig.2(b), RBEAD deploys servers in the upper-right region where POIs are densely clustered, yet its overall coverage is minimal. Fig.2(c) shows that CRBEAD achieves a more balanced distribution by reducing servers in the upper-right area to expand its coverage, while some servers still remain in low-relevance regions.

In contrast, the MTGA variants exhibit different emphases. As depicted in Fig.2(d), MTGA-M clusters servers in areas with densely distributed, high-relevance POIs for optimal direct coverage, while in Fig.2(e), MTGA-T adopts a more dispersed strategy to increase the total number of covered POIs. Unlike BEAD, it favors proximity to high-relevance clusters to enhance network expansion. As a result, Fig.2(f) illustrates that MTGA combines the advantages of direct coverage and topological expansion by selecting servers that not only cover numerous high-relevance POIs directly but also extend coverage to surrounding POIs via the connectivity of edge topological network.

D. Performance Impact of Parameters

To comprehensively evaluate the performance of our approach, we vary the following parameters on *SHH Telecom@2* dataset, observing their impact on edge service deployment performance under different evaluation metrics. The parameters are defined as follows:

- Hop threshold h_T. It limits the number of hops each node in the edge topology can forward a service request and is varied from 1 to 3.
- Decay factor γ. It quantifies the decay rate of the effective benefit achieved through multi-hop forwarding of a deployed service, and is varied from 0.2 to 1.

1) Impact of Hop Threshold h_T: Fig.3(a) indicates that variations in h_T have a negligible effect on the MB, with methods such as BEAD showing a slight decline, while MTGA-

M and MTGA consistently maintain significant advantages. A larger h_T permits POIs to reach services via more intermediate nodes, thereby expanding the coverage. However, as demonstrated in Fig.3(b)(c), it amplifies BEAD's reliance on the sheer number of covered POIs, resulting in a decline in its TB and GCB at higher h_T. In contrast, MTGA-T consistently achieves the highest TB by accounting for the decay in relevance with increasing hops, and MTGA outperforms all methods in GCB by effectively integrating direct coverage with multi-hop expansion. In terms of CPU Time, Fig.3(d) shows that although the computational complexity of MTGA and its variants increases with h_T, MTGA's efficient fitness function keeps its runtime comparable to that of BEAD and RBEAD, thereby demonstrating its adaptability to complex edge topological network in real service-oriented application scenarios.

2) Impact of Decay Factor γ: Fig. 4(a) shows that γ does not affect the MB of MTGA-M, as its strategy is solely based on directly covered POIs. In contrast, MTGA-T experiences a notable decline in MB due to its dual focus on POI quantity and relevance, whereas MTGA's MB has slightly decreased but remains at a high level. As shown in Fig.4(b)(c), both TB and GCB of MTGA and MTGA-T increase significantly with larger γ, while the improvements for MTGA-M are minimal. Fig.4(d) further indicates that the CPU time remains consistent across different γ values among all competing methods, suggesting that the decay factor does not affect computational complexity. Overall, MTGA effectively balances MB and TB while maintaining stable CPU time, offering a cost-effective deployment strategy across various γ settings.

VI. Related Work

The rapid growth of IoT and data-intensive applications [29] has accelerated the deployment of various services and applications at the network edge to meet user demands for low latency and high-quality responses. Existing research on ESD has proposed various strategies from different perspectives. Some studies emphasize maximizing user coverage to enhance service accessibility [19]. Other works prioritize minimizing deployment costs [11], [30]. In recent years, more studies have shifted towards jointly considering service request scheduling or resource allocation to minimize latency and energy consumption [12], [16], [31], [32]. Additionally, there is growing interest in integrating resource utilization and transmission costs to assess the utility of service deployment [9], [33].

Given the heterogeneity of edge resources and storage-computation imbalance, edge server collaboration is crucial to leveraging edge advantages [34]. Therefore, many studies have started to address edge tasks through collaboration enabled by edge network topology. Luo et al. [35] formulated a combinatorial optimization problem to maximize data deduplication, while Peng et al. [36] investigated cache sharing and collaborative caching to enhance service quality and load balancing, despite the resulting transmission delays and energy overheads. Moreover, most works assume fully reliable edge servers, overlooking the challenges posed by dynamic and volatile environments. To enhance reliability, Liu et al. [37] proposed an online backup approach, while Wang et al. [38] developed a concept drift-based restart method. However, few studies assess the trade-offs of topology expansion and quantify its benefits from the service provider's perspective.

Additionally, the rising demand for location-sensitive applications, understanding service requirement distribution [39] is increasingly crucial to deployment strategy for effective edge service usability. Existing studies have demonstrated that POI data can effectively reveal geographic demand patterns and predict application usage [40]. In the context of Location-Based Social Networks (LBSN), in order to solve the cold start problem of lack of user historical access records, Tu et al. [41] proposed a generative model to infer users' location preferences from application data, further corroborating Yu et al.'s [24] findings on the close relationship between application usage and geographic location. Recently, some studies have begun incorporating POI information into edge scenarios, Wu et al. [42] analyzed spatiotemporal patterns from base station requests and POI distributions to develop an online learning-based service provisioning system. However, user demand distribution in heterogeneous edge scenarios with real-world POI data remains unexplored in ESD problem.

To address the limitations of traditional edge service deployment strategies, we first integrate POI attributes with geographic distribution. Meanwhile, considering the coverage, resource constraints and network characteristics, we evaluate the trade-offs of multi-hop topology expansion, aiming to achieve effective service deployment strategy with the optimization of both POI-based matching benefit and topology-aware decay benefit.

VII. Conclusion

This paper formulated the problem of POI-based edge service deployment with topology-aware optimization (POI-ESD). It is transformed as a multi-constrained multi-objective optimization problem that combines POI-based matching benefit and topology-aware decay benefit, and we proved its $\mathcal{NP}$-hardness. To solve this problem, we proposed a graph-encoded genetic algorithm, MTGA, to effectively and efficiently generate deployment strategies. Comprehensive experiments were conducted on a real-world dataset. The results demonstrate that MTGA consistently achieved superior performance on both functionality matching and topological decay on multiple evaluation metrics, proving its robustness and potentially practical applicability.

In future work, we aim to leverage spatio-temporal characteristics to design more efficient online deployment strategies, adapting to the dynamics of real service-oriented application demands.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088).

REFERENCES

[1] W. Chu, X. Jia, Z. Yu, J. C. Lui, and Y. Lin, "Joint Service Caching, Resource Allocation and Task Offloading for MEC-Based Networks: A Multi-Layer Optimization Approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 2958–2975, 2024.

[2] X. Xue, X. Yu, and F.-Y. Wang, "ChatGPT Chats on Computational Experiments: From Interactive Intelligence to Imaginative Intelligence for Design of Artificial Societies and Optimization of Foundational Models," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 6, pp. 1357–1360, 2023.

[3] J. Yang, A. K. Bashir, Z. Guo, K. Yu, and M. Guizani, "Intelligent cache and buffer optimization for mobile VR adaptive transmission in 5G edge computing networks," *Digital Communications and Networks*, vol. 10, no. 5, pp. 1234–1244, 2024.

[4] L. Wang, L. Jiao, T. He, J. Li, and H. Bal, "Service Placement for Collaborative Edge Applications," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 34–47, 2021.

[5] G. Liu, X. Xu, X. Xu, X. Ji, L. Qi, and X. Zhang, "Optimized MARL for Latency-Sensitive Collaborative Service Placement in Edge Computing," in *IEEE International Conference on Web Services (ICWS)*, 2024, pp. 1089–1096.

[6] J. Wang, K. Liu, B. Li, T. Liu, R. Li, and Z. Han, "Delay-Sensitive Multi-Period Computation Offloading with Reliability Guarantees in Fog Networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2062–2075, 2020.

[7] X. He, J. Zheng, H. Dai, B. Liu, W. Dou, G. Chen, and F. Xiao, "History-Assisted Online User Allocation in Mobile Edge Computing," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 140–149.

[8] P. Lai, Q. He, G. Cui, X. Xia, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Edge User Allocation with Dynamic Quality of Service," in *International Conference on Service-Oriented Computing (ICSOC)*, 2019, pp. 86–101.

[9] L. Chen, C. Shen, P. Zhou, and J. Xu, "Collaborative Service Placement for Edge Computing in Dense Small Cell Networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 377–390, 2021.

[10] G. Zou, Y. Liu, Z. Qin, J. Chen, Z. Xu, Y. Gan, B. Zhang, and Q. He, "TD-EUA: Task-Decomposable Edge User Allocation with QoE Optimization," in *International Conference on Service-Oriented Computing (ICSOC)*, 2020, pp. 215–231.

[11] S. Deng, Z. Xiang, J. Taheri, M. A. Khoshkholghi, J. Yin, A. Y. Zomaya, and S. Dustdar, "Optimal Application Deployment in Resource Constrained Distributed Edges," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1907–1923, 2021.

[12] T. He, H. Khamfroush, S. Wang, T. La Porta, and S. Stein, "It's Hard to Share: Joint Service Placement and Request Scheduling in Edge Clouds with Sharable and Non-Sharable Resources," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 365–375.

[13] R. Bi, T. Peng, J. Ren, X. Fang, and G. Tan, "Joint Service Placement and Computation Scheduling in Edge Clouds," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 47–56.

[14] T. Ouyang, X. Chen, Z. Zhou, R. Li, and X. Tang, "Adaptive User-Managed Service Placement for Mobile Edge Computing via Contextual Multi-Armed Bandit Learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1313–1326, 2023.

[15] J. Jia and P. Wang, "Low Latency Deployment of Service-Based Data-Intensive Applications in Cloud-Edge Environment," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 57–66.

[16] W. Fan, L. Zhao, X. Liu, Y. Su, S. Li, F. Wu, and Y. Liu, "Collaborative Service Placement, Task Scheduling, and Resource Allocation for Task Offloading With Edge-Cloud Cooperation," *IEEE Transactions on Mobile Computing*, vol. 23, no. 1, pp. 238–256, 2024.

[17] P. Wang, J. Xu, M. Zhou, and A. Albeshri, "Budget-Constrained Optimal Deployment of Redundant Services in Edge Computing Environment," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9453–9464, 2023.

[18] X. Xue, X. Yu, D. Zhou, C. Peng, X. Wang, C. Zhou, and F. Wang, "Computational experiments: Past, present and perspective," *Acta Automatica Sinica*, vol. 49, no. 2, pp. 246–271, 2023.

[19] F. Chen, J. Zhou, X. Xia, H. Jin, and Q. He, "Optimal Application Deployment in Mobile Edge Computing Environment," in *IEEE International Conference on Cloud Computing (CLOUD)*, 2020, pp. 184–192.

[20] F. Chen, J. Zhou, X. Xia, Y. Xiang, X. Tao, and Q. He, "Joint Optimization of Coverage and Reliability for Application Placement in Mobile Edge Computing," *IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 3946–3957, 2023.

[21] L. Zhao, B. Li, W. Tan, G. Cui, Q. He, X. Xu, L. Xu, and Y. Yang, "Joint Coverage-Reliability for Budgeted Edge Application Deployment in Mobile Edge Computing Environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 3760–3771, 2022.

[22] Z. Xu, G. Zou, X. Xia, Y. Liu, Y. Gan, B. Zhang, and Q. He, "Distance-aware Edge User Allocation with QoE Optimization," in *IEEE International Conference on Web Services (ICWS)*, 2020, pp. 66–74.

[23] T. S. Rappaport, *Wireless Communications: Principles and Practice*. USA: Prentice Hall, 1996.

[24] D. Yu, Y. Li, F. Xu, P. Zhang, and V. Kostakos, "Smartphone App Usage Prediction Using Points of Interest," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–21, 2018.

[25] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," in *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[26] J. Liang, Z. Feng, H. Gao, Y. Chen, J. Huang, and H.-L. Truong, "Deep Reinforcement Learning based Reliability-aware Resource Placement and Task Offloading in Edge Computing," in *IEEE International Conference on Web Services (ICWS)*, 2024, pp. 686–695.

[27] Y. Li, A. Zhou, X. Ma, and S. Wang, "Profit-Aware Edge Server Placement," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 55–67, 2022.

[28] B. Li, Q. He, G. Cui, X. Xia, F. Chen, H. Jin, and Y. Yang, "READ: Robustness-Oriented Edge Application Deployment in Edge Computing Environment," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1746–1759, 2022.

[29] X. Xue, Y. Guo, S. Chen, and S. Wang, "Analysis and Controlling of Manufacturing Service Ecosystem: A Research Framework Based on the Parallel System Theory," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1598–1611, 2021.

[30] J. Huang, A. Zhou, and S. Wang, "Price-Aware Service Deployment in Hierarchical Mobile-Edge Computing," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11 533–11 541, 2022.

[31] T. Liu, S. Ni, X. Li, Y. Zhu, L. Kong, and Y. Yang, "Deep Reinforcement Learning Based Approach for Online Service Placement and Computation Resource Allocation in Edge Computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3870–3881, 2023.

[32] L. Yang, J. Jia, H. Lin, and J. Cao, "Reliable Dynamic Service Chain Scheduling in 5G Networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4898–4911, 2023.

[33] Y. Li, W. Dai, X. Gan, H. Jin, L. Fu, H. Ma, and X. Wang, "Cooperative Service Placement and Scheduling in Edge Clouds: A Deadline-Driven Approach," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3519–3535, 2022.

[34] X. Ma, A. Zhou, S. Zhang, and S. Wang, "Cooperative Service Caching and Workload Scheduling in Mobile Edge Computing," in *IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 2076–2085.

[35] R. Luo, H. Jin, Q. He, S. Wu, Z. Zeng, and X. Xia, "Graph-Based Data Deduplication in Mobile Edge Computing Environment," in *International Conference on Service-Oriented Computing (ICSOC)*, 2021, pp. 499–515.

[36] J. Peng, Q. Li, X. Tang, D. Zhao, C. Hu, and Y. Jiang, "A Cooperative Caching System in Heterogeneous Edge Networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 7, pp. 7635–7649, 2024.

[37] Y. Liu, X. Shang, Y. Mao, Z. Liu, and Y. Yang, "Availability Aware Online Virtual Network Function Backup in Edge Environments," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 3909–3922, 2024.

[38] L. Wang, J. Liu, and Q. He, "Concept Drift-Based Checkpoint-Restart for Edge Services Rejuvenation," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 1713–1725, 2023.

[39] X. Xue, D. Zhou, X. Yu, G. Wang, J. Li, X. Xie, L. Cui, and F.-Y. Wang, "Computational Experiments for Complex Social Systems: Experiment Design and Generative Explanation," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 4, pp. 1022–1038, 2024.

[40] J. Xie and Z. Chen, "Hierarchical Transformer with Spatio-temporal Context Aggregation for Next Point-of-interest Recommendation," *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1–30, 2023.

[41] Z. Tu, Y. Fan, Y. Li, X. Chen, L. Su, and D. Jin, "From Fingerprint to Footprint: Cold-Start Location Recommendation by Learning User Interest from App Data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–22, 2019.
[42] T. Wu, X. Fan, H. Wei, Y. Qu, C. Xiang, P. Yang, and F. Wu, "Predictive Service Provisioning With Online Learning in Wireless Edge Networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4076–4091, 2024.