Deep Learning Based Service Composition in Integrated Aerial-Terrestrial Networks

Mohammad Farhoudi[1], Masoud Shokrnezhad[1,2], Somayeh Kianpisheh[1], and Tarik Taleb[3]
[1] *Oulu University, Finland*; {mohammad.farhoudi, masoud.shokrnezhad, somayeh.kianpisheh}@oulu.fi
[2] *ICTFICIAL Oy, Espoo, Finland*; masoud.shokrnezhad@ictficial.com
[3] *Ruhr University Bochum (RUB), Germany*; {tarik.taleb}@rub.de

***Abstract*—The explosive growth of user devices and emerging applications is driving unprecedented traffic demands, accompanied by stringent Quality of Service (QoS) requirements. Addressing these challenges necessitates innovative service orchestration methods capable of seamless integration across the edge-cloud continuum. Terrestrial network-based service orchestration methods struggle to deliver timely responses to growing traffic demands or support users with poor or lack of access to terrestrial infrastructure. Exploiting both aerial and terrestrial resources in service composition increases coverage and facilitates the use of full computing and communication potentials. This paper proposes a service placement and composition mechanism for integrated aerial-terrestrial networks over the edge-cloud continuum while considering the dynamic nature of the network. The service function placement and service orchestration are modeled in an optimization framework. Considering the dynamicity, the Aerial Base Station (ABS) trajectory might not be deterministic, and their mobility pattern might not be known as assumed knowledge. Also, service requests can traverse through access nodes due to users' mobility. By incorporating predictive algorithms, including Deep Reinforcement Learning (DRL) approaches, the proposed method predicts ABS locations and service requests. Subsequently, a heuristic isomorphic graph matching approach is proposed to enable efficient, latency-aware service orchestration. Simulation results demonstrate the efficiency of the proposed prediction and service composition schemes in terms of accuracy, cost optimization, scalability, and responsiveness, ensuring timely and reliable service delivery under diverse network conditions.**

***Index Terms*—Service Placement, Composition, and Orchestration, Resource Allocation, and 6G Aerial-Terrestrial Networks**

I. Introduction

Today's landscape exhibits a considerable increase in the number of users, leading to an immense rise in traffic demand [1]. The rapid growth of emerging applications like the Industrial Internet of Things (IIoT) has further driven this demand [2]. Resultantly, there is a need to reassess service orchestration methods to meet evolving capacity and Quality of Service (QoS) criteria [3]. This imperative aims to ensure the efficient allocation of resources throughout the edge-cloud continuum [4]. The placement of service functional block instances (or simply instances) and traffic routing emerge as challenges, particularly in light of the escalating complexity of service structures and stringent QoS demands.

Service composition solutions focusing on terrestrial networks have been studied in the literature [5]. However, the mobile nature of users and growing traffic demand highlight the inefficiency of terrestrial network-based solutions. Unmanned

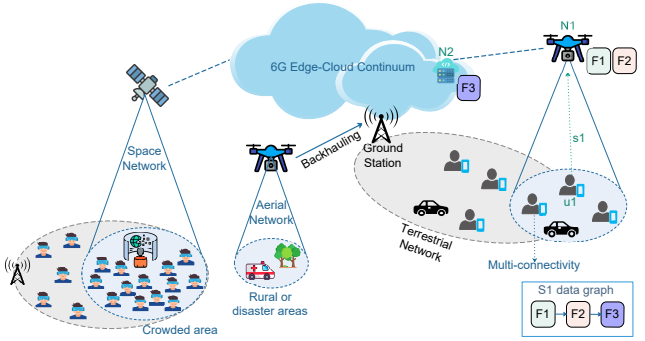

Figure 1. A conceptual diagram illustrating the integration of aerial and terrestrial networks in the 6G edge-cloud continuum.

Aerial Vehicles (UAVs) deployed as Aerial Base Stations (ABSs) present a viable option for leveraging edge computing resources to support users with poor or no terrestrial connectivity. This becomes particularly relevant in scenarios where ground networks encounter coverage limitations (Fig. 1).

Some studies have been conducted on service orchestration for non-terrestrial networks. Wei *et al.* [6] optimized UAV trajectory planning and service deployment in scenarios with obstacles. He *et al.* [7] studied the joint optimization of virtualized service provisioning and UAV trajectory planning. An optimization approach for service relocation and handover in UAV networks was provided by Bekkouche *et al.* [8]. Qu *et al.* [9] presented an offline learning-based orchestration scheme. Lastly, Wang *et al.* [10] presented Joint Composition Assignment and Placement (Jcap), integrating service composition with Virtual Network Function (VNF) placement and assignment to enhance resource allocation efficiency.

Although the above-mentioned methods address orchestration solutions, some are limited to aerial networks. Conversely, 6G network requirements encourage aerial-terrestrial resource integration to maximize network coverage and exploit the full potential of the network infrastructure. Considering the mobility of ABSs, the literature either plans the trajectory [7] or assumes deterministic and known mobility patterns for ABSs [8], [11]. Under multi-administrative/operator-based scenarios, ABSs are managed by several operators [12]. Hence, UAV trajectories exhibit multiplexity or a lack of opportunity to reveal their trajectory due to conventional purposes.

To address the above-mentioned gaps, this paper proposes a service placement and composition mechanism for integrated

aerial-terrestrial networks. To deal with the dynamic nature of ABS mobility patterns, a Deep Reinforcement Learning (DRL)-based scheme is provided to predict ABS locations. In contrast with a random stochastic behavior assumption for ABSs, exploiting the prediction results leads to a more efficient optimization of resource allocation. Also, to deal with users' mobility, an approach is followed in which requests can traverse through multiple Base Stations (BSs) and a DRL-based scheme is employed to predict requests at BSs. A heuristic method based on the Hungarian isomorphic graph matching approach is then provided to solve the near-optimal service placement and composition problem.

In the rest, Section II outlines the system model, and Section III gives the optimization framework. Section IV elaborates on deep learning based service placement and composition. Section V presents numerical findings, while Section VI offers concluding remarks and outlines future research directions.

II. System Model

Edge-Cloud Infrastructure: The network infrastructure contains aerial and terrestrial nodes equipped with computing, storage, and networking, modeled as a network graph $\mathcal{G}(\boldsymbol{\mathcal{N}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}})$. Network nodes $\boldsymbol{\mathcal{N}}$ include edge-cloud, Ground Base Station (GBS), and ABS nodes. Each node has a usage cost $\overline{\mathcal{C}}_n$, remaining battery power $\widehat{\mathcal{P}}_n$, and a capacity threshold $\widehat{\mathcal{C}}_n$. The network graph contains links ($\boldsymbol{\mathcal{L}} \subset \{l : (n, n') | n, n' \in \mathcal{N}\}$), with each link's maximum bandwidth $\widehat{\mathcal{L}}_l$ and the cost of using links for packet transmission $\overline{\mathcal{L}}_l$. Users and edge-cloud nodes should exist in each other's coverage to be considered linked. To model dynamicity in the network due to ABs and users mobility, directional paths $\boldsymbol{\mathcal{P}}^t = p : (\mathcal{H}_p^t, \mathcal{T}_p^t) | p \subset \boldsymbol{\mathcal{L}}$ varies over time t. Paths are defined by their head ($\mathcal{H}_p^t$) and tail ($\mathcal{T}_p^t$) nodes, with $\mathcal{J}_{p,l}^t$ indicating whether path p includes link l. Finally, the network architecture is tiered, with varying computing capacities in which edge nodes closer to users possess limited yet costly resources, whereas cloud nodes offer cost-effective and virtually limitless capabilities [13]. The nodes that form a particular path p are represented as $\mathcal{N}_p^\star$.

Service Providers: A set of service providers within the edge-cloud infrastructure offer various services, denoted by $\boldsymbol{\mathcal{S}} = \{1, 2, ..., \mathcal{S}\}$, distinguished by their definitions and requirements. The requirements include composition elements - inputs, outputs, preconditions - and QoS aspects such as cost and latency. Each service $s \in \boldsymbol{\mathcal{S}}$ has a predefined service data graph $\mathcal{G}_s$ and maintains a set of $\mathcal{C}_s$ atomic functional blocks, dubbed $\boldsymbol{\mathcal{F}}_s = \{1, 2, ..., \mathcal{C}_s\}$. An example can be found in Fig. 1, where user u_1 sends s_1 request with the data graph $\mathcal{G}_{s1}$, which has its functions deployed on N_1 and N_2. The services have an overall time requirement of $\mathcal{W}_s$ to orchestrate the organizational structure. Besides, each function $f \in \boldsymbol{\mathcal{F}}_s$ is implemented by instantiating an instance from $\boldsymbol{\mathcal{I}}_f = \{1, 2, ..., \mathcal{I}_f\}$, each of which capable of managing multiple requests at the cost of use $\overline{\mathcal{I}}_{f,i}$ but constrained by a capacity threshold $\widehat{\mathcal{I}}_{f,i}$.

Service Requests: A group of users initiates service requests represented by $\boldsymbol{\mathcal{R}} = \{1, 2, ..., \mathcal{R}\}$. Each request enters the system at time $\mathcal{T}_e$, demands a service denoted as $\mathcal{S}_r$, and has a request lifetime $\mathcal{W}_s$. Users exhibit dynamic behavior, with their locations varying over time as tracked by $\mathcal{L}_r^t$ and their Point of Attachment (PoA)–the user's network gateway–identified at each time slot $\mathcal{E}_r^t$. Upon request initiation, the selection of the most appropriate atomic service function instances for the request r becomes paramount. This selection process takes into account various service requirements, including the minimum capacity for each function $\breve{\mathcal{I}}_{r,f}^t$, minimum network bandwidth $\breve{\mathcal{L}}_r^t$, maximum acceptable End-to-End (E2E) latency $\breve{\mathcal{D}}_r^t$, traffic burst $\breve{\mathcal{B}}_r^t$, and maximum packet size for each function $\breve{\mathcal{Z}}_{r,f}^t$. It also considers $\breve{\mathcal{Y}}_r$, which signifies the upper limit of tolerable overall E2E latency for requests throughout $\mathcal{W}_s$ time slots commonly referred to as the Service-Level Agreement (SLA) requirement [14].

III. Optimization Framework

The service placement and composition problem is a Mixed-Integer Nonlinear Programming (MINLP) optimization with the formulation defined in (1). The formulation includes function instance placement for deploying them on nodes, path selection, functions assignment, and constraints associated with resources and service requirements. The primary goal (OF) is to maximize service request acceptance while reducing costs within a given period $\boldsymbol{\mathcal{T}}$. The binary variables $\ddot{\mathcal{A}}_{r,f,i}^t$ and $\ddot{\mathcal{E}}_{f,i,n}^t$ signify the selection of instance i of function f for request r, and the selection of the hosting node n for instance i, respectively. Scaling factor Ψ adjusts the relative impact of criteria for service request acceptance and total computational, request forwarding, and deployment costs.

$$\max \quad \text{OF} \quad \text{s.t} \quad \text{C1 - C11.} \tag{1}$$

$$\sum_{\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{F}}_{s_r}, \boldsymbol{\mathcal{I}}_f} \ddot{\mathcal{A}}_{r,f,i}^t - \Psi \left(\sum_{\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{F}}_s, \boldsymbol{\mathcal{I}}_f, \boldsymbol{\mathcal{N}}} \ddot{\mathcal{E}}_{f,i,n}^t \overline{\mathcal{C}}_n + \sum_{\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{F}}_{s_r}, \boldsymbol{\mathcal{I}}_f} \ddot{\mathcal{A}}_{r,f,i}^t \overline{\mathcal{I}}_{f,i} + \sum_{\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{R}}} \ddot{\mathcal{L}}_r^t \right) \tag{OF}$$

Resource Allocation: Allocating resources involves deploying function instances on network nodes, assigning requests to deployed instances, and routing packets within a capacity-constrained environment. Constraint (C1) assures that each request belongs to a unique instance of a function. Constraint (C2) ensures that each function instance selected by a request should be placed on an available infrastructure node for the duration of the requested service $\boldsymbol{\mathcal{T}}_r = [\mathcal{T}_e, \mathcal{T}_e + \mathcal{W}_s]$.

$$\sum_{\boldsymbol{\mathcal{I}}_f} \ddot{\mathcal{A}}_{r,f,i}^t \leq 1 \qquad \forall r, f, t \in \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{F}}_{s_r}, \boldsymbol{\mathcal{T}}_r, \tag{C1}$$

$$\sum_{\boldsymbol{\mathcal{N}}} \ddot{\mathcal{E}}_{f,i,n}^t > (\sum_{\boldsymbol{\mathcal{R}}} \ddot{\mathcal{A}}_{r,f,i}^t)/\mathcal{R} \qquad \forall f, i, t \in \boldsymbol{\mathcal{F}}_s, \boldsymbol{\mathcal{I}}_f, \boldsymbol{\mathcal{T}}. \tag{C2}$$

Maintaining stability while handling user demands and network conditions requires capacity constraints. Given the finite capacity, storage, and computational resources, the total number of requests assigned to each service instance (C3) and deployed on each node (C4) should not surpass capacity limits.

$$\sum_{\boldsymbol{\mathcal{R}}} \ddot{\mathcal{A}}_{r,f,i}^t \breve{\mathcal{I}}_{r,f}^t \leq \widehat{\mathcal{I}}_{f,i} \qquad \forall f, i, t \in \boldsymbol{\mathcal{F}}_s, \boldsymbol{\mathcal{I}}_f, \boldsymbol{\mathcal{T}}, \tag{C3}$$

$$\sum_{\boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{F}}_{s_r}, \boldsymbol{\mathcal{I}}_f} \ddot{\mathcal{E}}_{f,i,n}^t \ddot{\mathcal{A}}_{r,f,i}^t \breve{\mathcal{I}}_{r,f}^t \leq \widehat{\mathcal{C}}_n \qquad \forall n, t \in \boldsymbol{\mathcal{N}}, \boldsymbol{\mathcal{T}}, \tag{C4}$$

Establishing feasible E2E routes for each request is necessary to facilitate the transmission of inquiry traffic from a user to its designated instances and the return of the response. ABS movements result in a variation of paths $\boldsymbol{\mathcal{P}}^t$ during different time slots. To achieve round-trip path selection, a distinct inquiry path is chosen for each request. It originates at its network's entry node (PoA) and culminates at the specified first function deployed node. This path traverses through the other functions in an order adhering to $\mathcal{G}_s$ and returns to the PoA (C5). Binary variable $\overrightarrow{\mathcal{R}}^t_{r,f,p}$ indicates the assignment of path p for traffic steering to the function f of request r. Capacity limitation (C6) regulates the number of requests assigned to each link at any given time, ensuring optimal path allocation and further optimizing link allocation efficiency. Using (C7), we calculate the total costs associated with (OF).

$$\sum_{\boldsymbol{\mathcal{P}}^t|\mathcal{H}^t_p=\mathcal{E}^{\mathcal{T}_e}_{ur}\ \&\ \mathcal{T}^t_p=\mathcal{E}^{\mathcal{T}_r}_{ur}\ \&\ \forall n,f\in\mathcal{N}^\star_p,\mathcal{C}_{s_r}} \overrightarrow{\mathcal{R}}^t_{r,f,p}(\ddot{\mathcal{E}}^t_{i,f,n}==1)=1 \quad \forall r,t\in\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{T}}_r, \quad \text{(C5)}$$

$$\sum_{\boldsymbol{\mathcal{R}}}\check{\mathcal{L}}^t_r \sum_{\boldsymbol{\mathcal{F}_{s_r}},\boldsymbol{\mathcal{P}}^t} \mathcal{J}^t_{p,l}\overrightarrow{\mathcal{R}}^t_{r,f,p}\le\hat{\mathcal{L}}_l \quad \forall l,t\in\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{T}}, \quad \text{(C6)}$$

$$\ddot{\mathcal{L}}^t_r=\sum_{\boldsymbol{\mathcal{L}}}\overline{\mathcal{L}}_l \sum_{\boldsymbol{\mathcal{F}_{s_r}},\boldsymbol{\mathcal{P}}^t}\mathcal{J}^t_{p,l}\overrightarrow{\mathcal{R}}^t_{r,f,p} \quad \forall r,t\in\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{T}}. \quad \text{(C7)}$$

QoS Requirements: In each time slot, the maximum acceptable latency should be maintained to ensure compliance with E2E latency thresholds (C10). Also, the cumulative latencies experienced by requests from each user across all time slots from different atomic service instances should not exceed the request SLA (C11). Continuous variables $\mathcal{D}^t_{r,l}$ and $\mathcal{D}^t_r$ quantify the latency of link l and each request's E2E latency respectively, including both network and computing latencies for r [15]. These constraints collectively safeguard timely and reliable service delivery, maintaining a high-performance standard in line with users' stringent requirements.

$$\mathcal{D}^t_{r,l}=(\sum_{\boldsymbol{\mathcal{R}}|r'\neq r}\check{\mathcal{B}}^t_{r'}+\sum_{\boldsymbol{\mathcal{F}}_{s_r}}\check{\mathcal{Z}}^t_{r',f})/\hat{\mathcal{L}}_l \quad \forall r,l,t\in\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{T}}, \quad \text{(C8)}$$

$$\mathcal{D}^t_r=\sum_{\boldsymbol{\mathcal{F}_{s_r}},\boldsymbol{\mathcal{P}}^t,\boldsymbol{\mathcal{L}}}\mathcal{J}^t_{p,l}\mathcal{D}^t_{r,l}\overrightarrow{\mathcal{R}}^t_{r,f,p}+\sum_{\boldsymbol{\mathcal{F}}_{s_r}}\check{\mathcal{Z}}^t_{r,f}/\check{\mathcal{I}}^t_{r,f} \quad \forall r,t\in\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{T}}, \quad \text{(C9)}$$

$$\mathcal{D}^t_r\le\check{\mathcal{D}}^t_r \quad \forall r,t\in\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{T}}, \quad \text{(C10)}$$

$$\sum_{\boldsymbol{\mathcal{T}}_r}\mathcal{D}^t_r\le\check{\mathcal{Y}}_r \quad \forall r\in\boldsymbol{\mathcal{R}}. \quad \text{(C11)}$$

IV. Proposed Method

The problem of service function placement and composition is reduced to the multidimensional knapsack problem and shown to be NP-hard [16]. The total number of possible permutations of placements in (1) is of order $\mathcal{R}!\mathcal{T}!\mathcal{N}\mathcal{S}\mathcal{C}_s\mathcal{P}$ which also illustrates the complexity of the problem. To deal with the dynamic nature of the network due to the non-determinism of ABS locations, as well as the dynamicity in requests' arrival due to users' mobility, a proactive service composition scheme called a predIction based huNgariaN isOmorphic serVice orchestrATION (INNOVATION) is provided. This method addresses imperfect knowledge constraints, overcomes problem complexity, and ensures high-quality service delivery. We adopt Dueling Double Deep Q-Learning (D3QL), since it overcomes over-optimistic and unstable approximations of Q-values by exploiting two separate Q-networks.

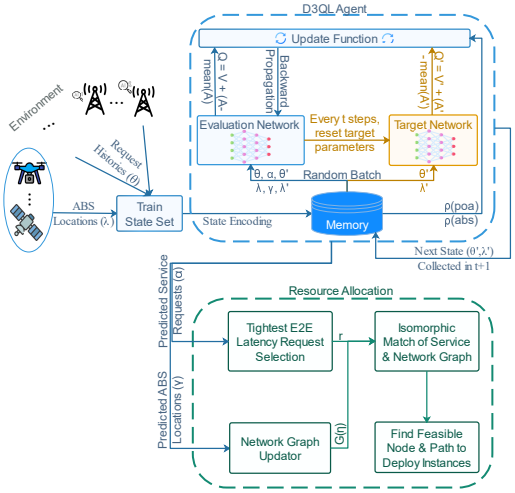

Figure 2. INNOVATION learning algorithm receives environment responses, stores them, and updates the evaluation network.

ABS Locations Prediction: The edge-cloud environment is divided into zones, with ABSs connected to the core network. Each ABS employs a D3QL agent to estimate ABS zone distribution (Algorithm 1). The Markov Decision Process (MDP) state λ represents ABS location histories, while the MDP action γ indicates the anticipated zone for the next time slot. Each neuron in the output layer of the D3QL Neural Networks (NNs) represents the probability of being in a specific zone, predicted proportionately to the Q-values using an ϵ-greedy strategy. The value of ϵ is high for exploration in early iterations and decreases linearly by ϵ' for exploitation. To motivate high accuracy gains, the MDP *reward* is defined based on prediction performance. The reward is 1 when the forecast and actual zones are the same, and -1 when they are different. The D3QL agent uses two Q-networks for action selection and evaluation (Fig. 2). After predicting ABS locations, the network graph and paths $\boldsymbol{\mathcal{P}}^t$ are updated by removing outdated links to the core network and establishing new links at anticipated locations.

Service Requests Prediction: As users move, service request demands vary time-wise. Users under its coverage, which varies with their movement, determine the demand for a specific service from a BS at a particular time slot. For efficient service function placement, the prediction of service request arrivals is required. At each network node that operates as a PoA, a D3QL agent is employed to predict the distribution of the request's arrival, i.e., the probability that a request r is being requested at the next time slot (Algorithm 1). Each agent considers state θ as the vector of arrived requests to the PoA during the last m time slot. The agent provides the action α returning a list of z requests with the highest likelihood. Finally, the reward ρ^t_{poa} is assigned based on arrival requests' prediction accuracy. Comparing the predicted and the actual arrived requests, rewards 1 and 0 are issued for correct and incorrect predictions, respectively.

Service Placement and Resource Allocation: This phase

focuses on determining optimal service function placements, assignments, and traffic steering for predicted requests across the edge-cloud continuum. A heuristic method is provided to allocate resources based on anticipated requests and the expected network environment. The service graph consists of nodes denoting atomic functions and edges representing data flows or dependencies between these functions. Similarly, the network graph contains network nodes such as ABSs, GBSs, and edge-cloud nodes interconnected by links with latency and capacity constraints. Isomorphic graph matching aligns the service graph with the network graph by identifying a feasible correspondence between their nodes and edges. This mapping ensures service functional relationships and dependencies are preserved while satisfying resource constraints.

The proposed method employs a Hungarian heuristic algorithm to prioritize requests with the most stringent latency needs. This prioritization ensures that time-sensitive requests are addressed first, reducing QoS violations. For each request, the algorithm evaluates candidate nodes $\overline{\mathcal{F}}(\boldsymbol{\eta})$ based on computational capacity, E2E latency, and connectivity. The total deployment cost is the sum of the node (computational) and path (communication) costs. The optimal node-path combination with the lowest cost that meets all constraints is selected for deployment (steps 7–9). After selecting optimal nodes in $\overline{\mathcal{F}}$, the method deploys function instances and establishes communication paths (steps 10–17). Existing instances are reused to improve resource utilization, and if no feasible node or path is available, the algorithm searches for alternatives. The network graph is dynamically updated based on predicted ABS locations. The iterative nature of the algorithm ensures that all service requests are processed, with priority given to the most critical ones. If a request cannot be supported due to resource limitations, it is flagged as unsupported. A global variable σ tracks instance placements, enabling efficient resource reuse, reducing redundancy, and streamlining the solution space.

V. Simulation Results

This section evaluates the proposed method's performance. We compare our proposed method with the following methods: 1) the optimal solution that possesses omniscient knowledge of ABS locations and actual requests in advance; 2) a random strategy in which function instances and nodes are placed randomly and assigned to service requests randomly; 3) Jcap method [10] that composes a feasible composition by considering nodes' capacity while placing services and assigning VNFs within users' close distance. The request numbers are strategically increased to assess the scalability of the proposed method. By subjecting our method to a diverse range of request volumes, from moderate to intense, we aim to examine its ability to efficiently allocate resources and meet SLA requirements across fluctuating demand levels. The outcomes of comparative methods are illustrated in Fig. 3.

Fig. 3.1 depicts the deployment and resource allocation costs per request. The optimized method, knowing mobility patterns, achieves the lowest cost through cost optimization. Jcap favors low-cost cloud nodes, achieving lower costs than other methods. INNOVATION closely matches the optimized method by accurately estimating ABS locations and service request arrivals, as well as a cost-driven approach to resource allocation. As the number of requests increases, the cost of the proposed INNOVATION method remains relatively stable.

Fig. 3.2 illustrates INNOVATION's low latency for accepted requests, achieved by the latency-aware heuristic algorithm. It improves latency up to 20% over the optimized method. Jcap, despite its multi-hop access to functions, overlooks latency in resource allocation, leading to longer paths and higher latency. High latency is experienced by the optimized method since it focuses on cost optimization while ensuring deadline satisfaction. As expected, random placement and assignment of functions have the highest latency, as arbitrary resource allocation and inefficient routing result in prolonged data transmission times. Overall, INNOVATION maintains SLA-compliant latency, with only slight increases as requests grow.

Fig. 3.3 shows the number of unsupported requests, which serves as a metric for evaluating service continuity. The optimized method supports the most number of requests due

Algorithm 1: DRL method for predictions

Input: $\tau, \epsilon, \epsilon', \widetilde{\epsilon}, \theta_0 \leftarrow \{\}, \lambda_0 \leftarrow \{\}, mem_r \leftarrow \{\}, mem_a \leftarrow \{\}$
Result: $\alpha_{\tau+1}, \gamma_{\tau+1}$
1 update states $(\theta_\tau, \lambda_\tau) \leftarrow$ PoA's requests & ABS' locations
2 **if** $\tau < m$ **then**
3 $\quad \alpha_{\tau+1}, \gamma_{\tau+1} \leftarrow$ random z services, random locations
4 **else**
5 $\quad \zeta \leftarrow$ generate a random number from $[0:1]$
6 $\quad$ **if** $\zeta > \epsilon$ **then**
7 $\quad\quad \alpha_{\tau+1}, \gamma_{\tau+1} \leftarrow$ select top Q-values
8 $\quad$ **else**
9 $\quad\quad \alpha_{\tau+1}, \gamma_{\tau+1} \leftarrow$ select random values
10 $\quad$ calculate rewards $(\rho^\tau_{poa}, \rho^\tau_{abs})$
11 $\quad mem_r \leftarrow mem_r \cup \{(\theta_{\tau-1}, \alpha_{\tau-1}, \rho^\tau_{poa}, \theta_\tau)\}$
12 $\quad mem_a \leftarrow mem_a \cup \{(\lambda_{\tau-1}, \gamma_{\tau-1}, \rho^\tau_{abs}, \lambda_\tau)\}$
13 $\quad$ choose a sample form mem_r, mem_a and train
14 $\quad$ **if** $\epsilon > \widetilde{\epsilon}$ **then**
15 $\quad\quad \epsilon \leftarrow \epsilon - \epsilon'$

Algorithm 2: Hungarian isomorphic service placement

Input: $\mathcal{T}$, $\alpha_0 \leftarrow \{\}$, $\gamma_0 \leftarrow \{\}$, $\sigma \leftarrow \{\}$
1 **for** *each* τ *in* $[1:\mathcal{T}]$ **do**
2 $\quad \alpha_{\tau+1}, \gamma_{\tau+1} \leftarrow$ Predict Requests and ABS locs (Algorithm 1)
3 $\quad$ update $\mathcal{G}(\boldsymbol{\mathcal{N}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}^\tau)$ using ABS locations $(\gamma_{\tau+1})$
4 $\quad (\boldsymbol{\mathcal{R}}$, PoAs$) \leftarrow$ collect $\alpha_{\tau+1}$ of all PoAs & convert to table
5 $\quad$ **while** $\boldsymbol{\mathcal{R}}$ *is not empty* **do**
6 $\quad\quad r \leftarrow$ the tightest E2E latency required request
7 $\quad\quad$ **for** *function* f *& isomorphic match* $\mathcal{G}(\boldsymbol{\eta})$ *&* $\mathcal{G}_s$ **do**
8 $\quad\quad\quad$ **if** f *in* σ *& feasible n,i (node & instance) in* σ **then**
9 $\quad\quad\quad\quad \phi_1 \leftarrow \overline{\mathcal{C}}_n + \overline{\mathcal{I}}_{f,i}$
10 $\quad\quad\quad$ **for** *set of nodes* $n_f \in \overline{\mathcal{F}}(\boldsymbol{\eta})$ **do**
11 $\quad\quad\quad\quad \boldsymbol{\mathcal{P}} \leftarrow$ set of paths between n_f, n
12 $\quad\quad\quad\quad p_m \leftarrow$ lowest latency $p \in \boldsymbol{\mathcal{P}}$
13 $\quad\quad\quad\quad \phi_2 \leftarrow \overline{\mathcal{C}}_{n_f} + \overline{\mathcal{I}}_{f,i} + \sum_{p_m} \ddot{\mathcal{L}}^\tau_r$
14 $\quad\quad\quad\quad \mathcal{D}_p \leftarrow p_m$'s links latency
15 $\quad\quad\quad$ **if** $\phi_1 + \phi_2 < mincost$ *&* $\mathcal{D}_p \leq \breve{\mathcal{D}}^\tau_r$ **then**
16 $\quad\quad\quad\quad mincost = \phi_1 + \phi_2$
17 $\quad\quad\quad\quad \chi = \{f\}, \{n + n_f\}, \{i\}, \{p_m\}$
18 $\quad\quad \ddot{\mathcal{A}}^\tau_{r,\chi_f,\chi_i} \leftarrow 1, \ddot{\mathcal{E}}^\tau_{\chi_f,\chi_i,\chi_n}, \overrightarrow{\mathcal{R}}^\tau_{r,\overrightarrow{\chi_p}} \leftarrow 1$
19 $\quad\quad$ remove r from $\boldsymbol{\mathcal{R}}$
20 $\quad\quad$ **for** *each function* f *in* $\boldsymbol{\mathcal{F}}_{s_r}$ *of* $\mathcal{G}_{s_r}$ **do**
21 $\quad\quad\quad \sigma \leftarrow \sigma +$ $\{f : n, i$ (selected node and instance) $\}$

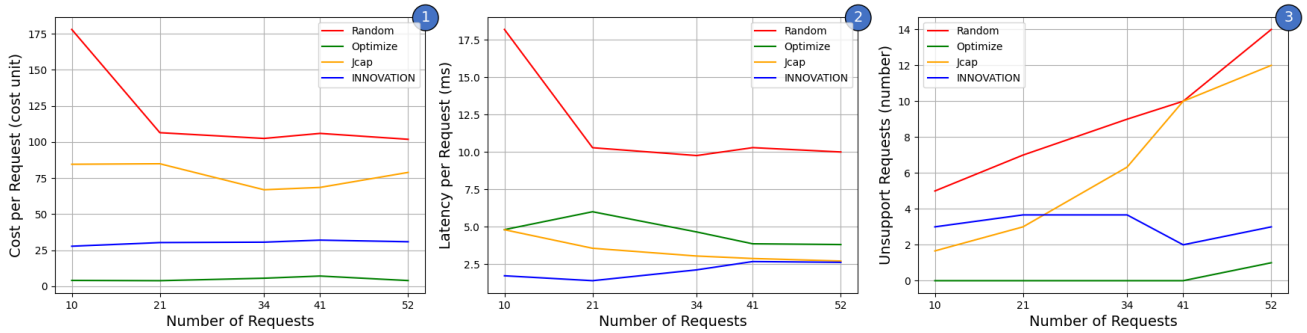

Figure 3. (1) cost per request, (2) E2E latency incurred per request, and (3) unsupported request numbers, while the number of requests is set to expand.

to its omniscient approach. INNOVATION performs competitively, with fewer than 4 unsupported requests, leveraging proactive resource allocation, low latency achievement, and non-terrestrial resource exploitation. Jcap's failure to account for actual request dynamics renders it inadequate at handling growing user demands as the number of requests expands. The random method fares the worst, with a large proportion of unsupported requests due to its arbitrary allocation strategy, leading to frequent resource shortages and unmet requests.

INNOVATION achieves an average total cost of 90% of the optimal method. It excels at near-optimal service placement, ensuring low cost and latency regardless of request volume. Unlike the optimized method, which prioritizes cost reduction through comprehensive scenario examination, the proposed method reduce solution space and excels at providing timely responses, on average 86% faster than optimal. Overall, INNOVATION emerges as an efficient and scalable solution, marked by low latency, cost-effectiveness, and timely responsiveness.

VI. Conclusion

This study proposed a service placement and composition approach for aerial-terrestrial edge-cloud networks. The networks, particularly leveraging UAVs as ABSs, offer unparalleled opportunities for addressing user mobility challenges and enabling ubiquitous service access. The problem was modeled as an optimization framework, addressing ABS nondeterminism and user mobility with deep reinforcement learning algorithms to predict ABS locations and service requests. By integrating predictive algorithms and isomorphic matching techniques, the method enabled cost-effective, latency-aware resource allocation. The simulation results confirmed the proposed method's superiority over baseline methods in terms of service request admission, cost, and E2E latency. The mechanism maintained service continuity with minimal unsupported requests, showcasing its scalability and robustness which highlighted its potential to enhance service orchestration in futuristic networks. Future work includes enhancing predictive capabilities and exploring resource allocation in the quantum internet [17] as well as multi-objective optimization for energy efficiency and fairness.

Acknowledgment

This research work is partially supported by the Business Finland 6Bridge 6Core project under Grant No. 8410/31/2022, the European Union's Horizon Europe research and innovation programme under the 6GSandbox project with Grant Agreement No. 101096328, and the Research Council of Finland 6G Flagship Programme under Grant No. 369116.

References

[1] Z. Sasan *et al.*, "Joint network slicing, routing, and in-network computing for energy-efficient 6g," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2024, pp. 1–6.

[2] S. K. Poorazad *et al.*, "Blockchain and deep learning-based ids for securing sdn-enabled industrial iot environments," in *Proc. IEEE Global Telecommun. Conf.*, 2023, pp. 2760–2765.

[3] M. Shokrnezhad *et al.*, "Towards a dynamic future with adaptable computing and network convergence (ACNC)," *arXiv preprint arXiv:2403.07573*, 2024.

[4] H. Mazandarani *et al.*, "A semantic-aware multiple access scheme for distributed, dynamic 6g-based applications," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2024, pp. 1–6.

[5] T. Wu *et al.*, "Predictive service provisioning with online learning in wireless edge networks," *IEEE Trans. Mobile Comput.*, 2023.

[6] X. Wei *et al.*, "Joint UAV trajectory planning, DAG task scheduling, and service function deployment based on DRL in UAV-empowered edge computing," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12 826–12 838, 2023.

[7] Q. He *et al.*, "Online joint optimization of virtual network function deployment and trajectory planning for virtualized service provision in multiple-unmanned-aerial-vehicle mobile-edge networks," *Electronics*, vol. 13, no. 5, p. 938, 2024.

[8] O. Bekkouche *et al.*, "Toward proactive service relocation for UAVs in MEC," in *Proc. IEEE Global Telecommun. Conf.* IEEE, 2021, pp. 1–7.

[9] C. Qu and Others, "Learning-based multi-drone network edge orchestration for video analytics," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 6, pp. 6331–6348, 2024.

[10] Z. Wang *et al.*, "Service function chain composition, placement, and assignment in data centers," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1638–1650, 2019.

[11] H. Mazandarani *et al.*, "Semantic-aware dynamic and distributed power allocation: a multi-uav area coverage use case," 2025.

[12] J. Humann *et al.*, "Modeling and simulation of multi-uav, multi-operator surveillance systems," in *Proc. IEEE Int. Syst. Conf.* IEEE, 2018.

[13] M. Farhoudi, M. Shokrnezhad *et al.*, "Discovery of 6G services and resources in edge-cloud-continuum," *IEEE Netw.*, pp. 1–1, 2024.

[14] M. Farhoudi *et al.*, "QoS-aware service prediction and orchestration in cloud-network integrated beyond 5G," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2023, pp. 369–374.

[15] M. Shokrnezhad *et al.*, "Near-optimal cloud-network integrated resource allocation for latency-sensitive B5G," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2022, pp. 4498–4503.

[16] F. Faticanti *et al.*, "Cutting throughput with the edge: App-aware placement in fog computing," in *Proc. IEEE Int. Conf. Cyber Security Cloud Comput.*, 2019, pp. 196–203.

[17] J. Prados-Garzon *et al.*, "Deterministic 6gb-assisted quantum networks with slicing support: A new 6gb use case," *IEEE Netw.*, vol. 38, no. 1, pp. 87–95, 2024.