Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

HAP-assisted multi-aerial base station deployment for capacity enhancement via federated deep reinforcement learning

Lei Liu[1], Haoran He[2], Fei Qi[1*], Yikun Zhao[2], Weiliang Xie[1], Fanqin Zhou[2] and Lei Feng[2]

Abstract

Aerial base stations (AeBSs), as crucial components of air-ground integrated networks, are widely employed in cloud computing, disaster relief, and various applications. How to quickly and efficiently deploy multi-AeBSs for higher capacity gain has become a key research issue. In this paper, we address the 3D deployment optimization problem of multi-AeBSs with the objective of maximizing system capacity. To overcome communication overhead and privacy challenges in multi-agent deep reinforcement learning (MADRL), we propose a federated deep deterministic policy gradient (Fed-DDPG) algorithm for the multi-AeBS deployment decision. Specifically, a high-altitude platform (HAP)-assisted multi-AeBS deployment architecture is designed, in which low-altitude AeBS act as the local nodes to train its own deployment decision model, while the HAP acts as the global node to aggregate the weights of local models. In this architecture, AeBSs do not exchange raw data, addressing data privacy concerns and reducing communication overhead. Simulation results show that the proposed algorithm outperforms fully distributed MADRL algorithms and closely approximates the performance of multi-agent deep deterministic policy gradient (MADDPG), which requires global information during training, but with less training time.

Keywords Aerial base station (AeBS), Capacity enhancement, Deep reinforcement learning (DRL), Federated reinforcement learning

Introduction

With the exponential growth of data-driven applications and the increasing demand for high-speed and ubiquitous communication services, the integration of cloud computing and aerial wireless networks has become a cornerstone of modern communication systems [1]. Cloud computing offers scalable and on-demand access to shared computing resources, enabling seamless data processing, storage, and application hosting for a wide range of users [2], while aerial base stations (AeBSs), base stations mounted on the aerial platforms such as unmanned aerial vehicles (UAVs) and high altitude platforms (HAPs), have the 1advantage of mobility and flexibility and can fill coverage gaps and adapt to changing communication demands. By harnessing cloud computing resources, AeBSs can offload computationally intensive tasks to cloud servers, enabling efficient data processing, storage, and analysis. This capability facilitates the AeBSs to provide real-time data services, such as multimedia streaming [3–6], social networks [7, 8], and industrial applications [9, 10].

To address the ever-increasing demand for higher communication rates of data-intensive applications and services in future cloud computing, network capacity has always been a critical indicator for network optimization [11–13]. The capacity optimization is more important for

*Correspondence:
Fei Qi
qif1@chinatelecom.cn
[1] Beijing Research Institute, China Telecom Corporation Limited, Beijing 102209, China
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Springer Open

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

AeBSs, which are constrained by their limited onboard payload. AeBSs should also seek communication methods to enhance capacity, such as utilizing the rich frequency bands of millimeter wave bands, while also having advantages such as directional communication, high gain, tiny antenna size, and easy integration [14]. In addition, static deployment strategies in conventional terrestrial networks can not be applied to aerial networks as it fails to fully exploit the adaptability of AeBSs in terms of network capacity, especially in areas with varying user densities and traffic patterns. Therefore, designing a dynamic and adaptive AeBS deployment strategy is crucial to achieving higher capacity performance.

In the 3D deployment of AeBSs, the traditional heuristic algorithm needs repeated calculation. The emergence of machine learning presents new opportunities [2, 7, 15, 16]. Especially, deep reinforcement learning (DRL) offers a novel paradigm for decision systems to accumulate and leverage experience within the environment [17]. For the deployment scenario of multi-AeBS, the single agent deep reinforcement learning has the problem of explosive action space and observation space, so multi-agent deep reinforcement learning (MADRL) is considered to be a more effective solution recently[18]. MADRL demonstrates excellent algorithm performance, but it requires frequent information communication among different agents. In the multi-AeBS deployment scenario, this frequent communication between AeBSs can deplete communication resources and escalate the complexity of the optimization problem. Moreover, different AeBSs may belong to different operators, and they are not willing to exchange information due to privacy reasons. Federated learning offers a promising approach to address challenges in scenarios where multiple data sources can jointly participate in model training without sharing their original data. This accelerates training speed and enhances privacy protection [19]. To address the aforementioned challenges, the integration of federated learning and DRL provides an efficient solution to reduce communication costs, improve learning performance, and ensure data integrity through a semi-distributed learning architecture [20].

Related work

In recent years, breakthrough works in cloud computing have led to improved performance from multiple perspectives [4, 15, 16]. The fusion of aerial networks and cloud computing is considered to be a promising paradigm that enables real-time service provision in the air [1, 21]. Reference [22] considers the integration of a cloudlet processor on AeBS to and optimizes the number of input and output bits transmitted between mobile devices and AeBS in the uplink and downlink, respectively, in each time slot. To measure the performance of AeBS-assisted cloud computing systems, capacity is one of the key metrics and researchers start to introduce mmWave technology into the AeBS to further improve the capacity of the system, such as in Refs. [14, 23]. Reference [24] compares the capacity gains achieved by static and UAV-based mmWave relays in various scenarios and results reveal the deployment of UAV is of great importance in improving capacity gain especially in clustered deployments. In Ref. [25], the system's capacity is maximized by optimizing both the deployment of the mmWave AeBS and the beamforming design.

Early works adopt heuristic algorithms to solve the deployment problem of AeBSs. Reference [26] proposes a proximal stochastic gradient descent-based alternating algorithm to optimize locations of AeBSs, aiming to maximize fair coverage and minimize energy consumption while satisfying backhaul constraints. Reference [27] presents an iterative solution that jointly optimizes the locations of AeBSs and the partially overlapped channel assignment scheme to maximize the throughput of a multi-AeBS system. However, in a dynamic environment where the network topology undergoes changes, the heuristic algorithm requires reinitialization and execution for the new topology, resulting in significant computational complexity for the system. To solve this problem, researchers begin to introduce DRL into AeBS deployment optimization problem. In Ref. [28], a 3D deployment algorithm for AeBSs based on deep Q-network (DQN) is proposed. Researchers found that deploying 3D locations of AeBSs using DRL can significantly increase system capacity. In Ref. [29], a DRL-based AeBS deployment algorithm is proposed, and simulation results demonstrate its significant superiority over the rewards-based greedy algorithm. Nevertheless, these methods follow a centralized paradigm, leading to limitations in scalability and flexibility. As the number of AeBSs to be controlled increases, the performance of the centralized DRL algorithm will degrade significantly due to the complexity caused by the large action space. Then researchers begin to use multi-agent deep reinforcement learning (MADRL) algorithms to solve the deployment problem of multi-AeBS. In Refs. [30, 31], a popular MADRL algorithm, multi-agent deep deterministic policy gradient (MADDPG), is adopted to decide the location planning of AeBSs, and results show that the MADDPG-based algorithm is more efficient than centralized DRL algorithms in obtaining the solution.

However, in the multi-agent framework mentioned above, it is necessary for agents to interact with information such as states and actions, which can result in frequent communication overhead and privacy issues between agents. Federated learning, a decentralized

machine learning method, promises to address these challenges by conducting distributed local model training and transmitting only encrypted model parameters. Reference [32] illustrates the potential benefits of introducing federated learning into aerial networks, and authors in Ref. [33] explore the air-ground integrated federated learning system, aiming to optimize location and resource allocation of the AeBS to achieve energy-efficient and low-latency training for terrestrial users. Considering the benefits brought by federated learning, several works propose to combine MADRL with federated learning framework to fix the abovementioned problems recently. In Ref. [34], researchers apply federated deep Q network (Fed-DQN) to the scenario of the internet of vehicles and concluded that the biggest advantage of introducing federated reinforcement learning is that it can achieve better results faster when new agent nodes are added. In Ref. [35], researchers develop a federated DRL-based cooperative edge caching approach, facilitating base stations to collaboratively learn a shared model and address the intricate and dynamic control challenges involved.

Motivation

As mentioned in the literatures above, facing the high data rate demand of future services, capacity enhancement is crucial for AeBS-assisted cloud computing networks. To achieve this goal, mmWave band can be used for air-to-ground links to enhance system capacity. Furthermore, the locations of AeBSs can be adjusted with the goal of maximizing system capacity, in order to leverage the advantages of flexible deployment of AeBSs and adapt to network dynamic requirements. However, traditional multi-AeBS deployment using MADRL presents challenges related to communication overhead and privacy. Introducing federated learning into MADRL brings several potential benefits: 1) Ensuring data privacy by avoiding raw data leakage; 2) Accelerating DRL model convergence, especially in time-sensitive scenarios; 3) Improving system scalability through parameter communication and generalization; 4) Addressing the data island problem arising from the limitation of each agent's observations. Therefore, in this work, we incorporate federated learning into MADRL to effectively address the deployment optimization problem of multi-AeBSs.

Our contributions

This paper proposes a federated DRL-based AeBS deployment optimization algorithm with an aim to maximize the capacity of AeBSs to serve ground users. The main contributions can be concluded as follows:

1. To improve the capacity while fully exploit the adaptive mobility of the multi-AeBS system, the 3D deployment optimization problem of multi-AeBS is investigated. AeBSs serve ground users using mmWave band to achieve higher capacity gain and a federated deep deterministic policy gradient (Fed-DDPG) algorithm is designed for the deployment decision of AeBSs.
2. A HAP-assisted multi-AeBS deployment architecture is designed, in which low altitude AeBS act as the local nodes to train its own deployment decision model, while the HAP acts as the global node to aggregate the weights of local models. In this architecture, different AeBSs do not exchange raw data, and the data privacy concern is addressed and the communication overhead of the whole system is reduced.
3. Extensive simulations are conducted to compare the performance of the proposed scheme with other MADRL algorithms, including centralized and fully decentralized training approaches. Results indicate that the Fed-DDPG algorithm outperforms fully distributed algorithms and closely approximates the performance of MADDPG, which requires global information during training, but with a reduced training time.

Organization

The rest of the paper is organized as follows. The system model is presented in Section System model and the proposed Fed-DDPG algorithm for 3D deployment of AeBSs is specified in Section Federated deep deterministic policy gradient algorithm for AeBSs. Section Simulation results and discussions presents the discussions of simulation results. Finally, Section Conclusion concludes the paper.

System model

We consider a scenario where multiple low-altitude mmWave AeBSs serve ground users in various regions, as depicted in Fig. 1. These AeBSs may belong to different operators, so for privacy reasons, they are unwilling to share information between each other, such as user location information. To address this challenge, we propose a HAP-assisted multi-AeBS deployment architecture. In this architecture, each AeBS trains its local deployment decision model and periodically uploads the model weights to a HAP. The HAP then aggregates the model weights from all AeBSs, maintaining a global model. This global model is subsequently distributed back to each AeBS for further training. Through this federated learning framework, AeBSs can benefit from the global

Fig. 1 HAP-assisted multi-AeBS deployment architecture

knowledge while avoiding direct data sharing, ensuring data privacy and promoting efficient collaboration among heterogeneous agents.

Air-to-Ground (A2G) channel model

AeBSs and UEs transmit data over an A2G channel, and the signal sent by AeBS suffers from both free space path loss and excessive path loss. The A2G mean path loss between UE u and AeBS m can be modeled as:

$$L_{LoS}^{m,u} = L_{FS}^{m,u} + \eta_{LoS}, \tag{1}$$

$$L_{NLoS}^{m,u} = L_{FS}^{m,u} + \eta_{NLoS}, \tag{2}$$

where $L_{FS}^{m,u}$ represents the free space path loss and $L_{FS}^{m,u} = 20log(4\pi f_c d_{m,u}/c)$, where c is the speed of light, f_c is the carrier frequency, and $d_{m,u}$ is the distance between AeBS m and UE u. η_{LoS} and η_{NLoS} refer to the mean value of the excessive path loss under the line-of-sight (LoS) environment and non-line-of-sight (NLoS) environment, respectively.

The LoS probability is related to the environment constants α, β and elevation angle $\theta_{m,u}$ between AeBS m and UE u, which can be expressed as [36]:

$$P_{LoS}^{m,u} = \frac{1}{1 + \alpha \exp\left[-\beta\left(\theta_{m,u} - \alpha\right)\right]}. \tag{3}$$

And the probability of NLoS can be obtained as $P_{NLoS}^{m,u} = 1 - P_{LoS}^{m,u}$.

As a result, the average path loss between UE u and AeBS m is as follows:

$$L_{m,u} = P_{LoS}^{m,u} \times L_{LoS}^{m,u} + P_{NLoS}^{m,u} \times L_{NLoS}^{m,u}. \tag{4}$$

mmWave beam scheduling

In addition to the A2G propagation path loss, the directional mmWave antenna gain significantly impacts the AeBS channel. In this study, we assume that a 3D beam

has a uniform gain G_M within its beamwidth and a small constant sidelobe gain G_S outside the beamwidth. The main lobe gain G_M can be determined as: [37]:

$$G_M = \frac{2 - (2 - (1 - \cos\delta_u))G_S}{1 - \cos\delta_u}. \tag{5}$$

In the procedure of 3D beam alignment, the time cost τ can be derived as [37]:

$$\tau = \frac{1 - \cos\delta_{T,s}}{1 - \cos\delta_{T,u}} \cdot \frac{1 - \cos\delta_{R,s}}{1 - \cos\delta_{R,u}} T_p, \tag{6}$$

where $\delta_{T,s}$ and $\delta_{R,s}$ are the sector width at transmitter and receiver respectively, while $\delta_{T,u}$ and $\delta_{R,u}$ are the beamwidth at transmitter and receiver respectively. Additionally, the time T_p is required for the beam to traverse through the entire sector and send a pilot signal at each position for alignment.

To cover the entire considered region with the narrow mmWave beam's strong directivity, beam scanning is necessary. When the number of associated UEs exceeds the available beams of an AeBS, a round-robin scheme is adopted for mmWave beam scheduling. The approximation of the average ratio of time-frequency resources η_u occupied by UE u is given by:

$$\eta_u = \begin{cases} 1, N_b \le N_u, \\ \frac{N_b}{N_u}, N_b > N_u, \end{cases} \tag{7}$$

where N_b is the number of mmWave beams of the AeBS and N_u is the number of UEs. In the following part, we take beam alignment time cost and resource utilization ratio into consideration when modeling the capacity.

Capacity model

In this work, each UE is associated with the unique AeBS that provides the strongest received signal. If AeBS m and UE u are associated, the SNR ξ_u of the signal received at UE u can be given as:

$$\xi_u = \frac{P_m G_{T,M} G_{R,M} L_{m,u}^{-1}}{\sigma^2}, \tag{8}$$

where P_m is the transmit power of AeBS m, σ^2 denotes the thermal noise power, $G_{T,M}$ and $G_{R,M}$ are the main lobe antenna gain of transmitter and receiver, which can be calculated according to Eq. (5).

Then the capacity of UE u is:

$$\varrho_u = \eta_u \left(1 - \frac{\tau}{T}\right) B \log_2(1 + \xi_u), \tag{9}$$

where B is the bandwidth, τ is the beam alignment time in Eq. (6), T is the time slot and τ must be less than one time slot T to ensure sufficient time for data transmission.

The capacity of the whole system can be calculated as:

$$\varrho = \sum_{m=1}^{M} \left[\sum_{u \in \mathcal{U}_m} \varrho_u \right], \tag{10}$$

where $\mathcal{U}_m$ represents the UE set associated with AeBS m and M is the total number of AeBSs in the system.

Problem formulation

In this work, our objective is to maximize the capacity of the entire system by optimizing the 3D locations of the AeBSs. The formulation of the optimization problem is as follows:

$$\begin{aligned} &\max_{\{x_m, y_m, h_m\}} \varrho \\ \text{s.t.} \quad &\text{C1}: x_m \in [x_{\min}, x_{\max}], \forall m \in \boldsymbol{M} \\ &\text{C2}: y_m \in [y_{\min}, y_{\max}], \forall m \in \boldsymbol{M} \\ &\text{C3}: h_m \in [h_{\min}, h_{\max}], \forall m \in \boldsymbol{M} \\ &\text{C4}: (x_m, y_m, h_m) \neq (x_l, y_l, h_l), \forall m, l \in \boldsymbol{M}, m \neq l, \end{aligned} \tag{11}$$

where $\boldsymbol{M}$ is the set of AeBSs in the considered system and ϱ is the system capacity in Eq. (10). Constraints C1-C3 prevent AeBSs from flying beyond the boundaries of the considered region, while C4 imposes restrictions to avoid collisions between AeBSs.

Federated deep deterministic policy gradient algorithm for AeBSs

In this section, we propose a Fed-DDPG algorithm to decide the 3D positions of AeBSs to achieve better system capacity. First of all, in our Fed-DDPG algorithm, the agent is each AeBS whose observation, action, and reward are defined as follows:

Observation: Each agent's observation is the current 3D location of each AeBS, $o_m = (h_m, x_m, y_m)$.

Action: Each agent's action is the movement distance in vertical and horizontal directions, $a_m = \{(\Delta h_m, \Delta x_m, \Delta y_m)\}$.

Reward: The reward is set as the system capacity calculated in Eq. (10).

In the DDPG algorithm, each agent has two modules: actor and critic. The critic is trained using neural networks to approximate the action-value function Q(s, a), while the actor is trained to output deterministic actions based on the current state, aiming to approximate the optimal deterministic policy.

The actor determines the selection probability of the action based on the current observation. In the environment, there are M agents and the deterministic policies for all agents $\mu = \{\mu_1, ..., \mu_M\}$ are parameterized by $\theta = \{\theta_1, ..., \theta_M\}$. The actor is updated by minimizing the gradient of the expected return for agent m, given by:

$$\nabla_{\theta_m} J = \mathbb{E}\left[\nabla_{\theta_m}\mu_m\left(o_m^j\right) \times \nabla_{a_m} Q_m^{\mu}\left(o^j, a_1^j, ..., a_M^j\right)\Big|_{a_m=\mu_m\left(o_m^j\right)}\right]. \tag{12}$$

The critic approximates the value function of the observation-action to evaluate the actor's selected action. The critic network is updated by minimizing the subsequent loss function:

$$L(\theta_m) = \mathbb{E}\left[y^j - Q_m^{\mu}\left(o^j, a_1^j, \ldots, a_M^j\right)\right]^2, \tag{13}$$

where y^j is the target value and can be estimated as:

$$y^j = r_m^j + \gamma \cdot Q_m^{\mu'}\left(o^j{}_{\text{new}}, a_1', ..., a_M'\right)\Big|_{a_m'=\mu_m'\left(s_m^j\right)}, \tag{14}$$

where γ is the discount factor and μ_m' is the target policy with parameter θ_m'.

Upload the model parameters to the global server for model aggregation after the dispersed training, the aggregation process for actor networks is as follows:

$$\theta_{g,a} = \frac{1}{n}\sum_{m=1}^{n}\theta_{l,a}^m, \tag{15}$$

where $\theta_{g,a}$ is the weight of the global actor network, $\theta_{l,a}^m$ is the weight of the local actor network, and n is the total number of selected AeBSs.

Similarly, the aggregation process for critic networks is as follows:

$$\theta_{g,c} = \frac{1}{n}\sum_{m=1}^{n}\theta_{l,c}^m, \tag{16}$$

where $\theta_{g,c}$ is the weight of the global critic network and $\theta_{l,c}^m$ is the weight of the local critic network.

Figure 2 shows our proposed Fed-DDPG algorithm structure. In the proposed algorithm, each AeBS agent has a DDPG model for distributed training. Each AeBS agent interacts with the environment independently and updates its own model based on its current state. Periodically, AeBS agents upload their local model weights to the HAP and the HAP receives and aggregates the weights, and then constructs an updated global model by using federated averaging. The federated learning function is conducted at HAP as HAP can hover at a high altitude and provide reliable, high LoS links with low-altitude AeBSs. Through this mechanism, the AeBS agents can exchange local model weights without exposing raw data, thereby addressing privacy concerns and reducing communication overhead. By maintaining and sharing a global model, the training efficiency is improved, as AeBS agents can leverage the collective experience and knowledge, overcoming the limitations of their individual experiences.

Input:
Training episodes $max_episode$
Max steps of each episode max_step
Aggregation frequency of federated learning f_a
Output:
Optimal 3D locations of each AeBS
Initialize:
Online actor network Q_a and target actor network Q_a' for AeBS m
Online critic network Q_c and target critic network Q_c' forAeBS m
Replay memory size N
Noise N_n for action exploration
for $episode = 1 : max_episode$ **do**
  Initialize the locations of AeBSs
  Initialize the locations of UEs
  for $step = 1 : max_step$ **do**
    Each AeBS m obtains its observation o_m
    Each AeBS selects the action based on its observation
    Each AeBS observe reward r_m
    Execute action a_m and update observation o_m
    Store transition (o_m, a_m, r_m, o_{m+1}) in D
    Sample random minibatch of transitions and training
    Soft update the actor and critic network
    if $episode \ mod \ f_a$=0 **then**
      AeBSs upload the model weights to the HAP for model aggregation according to Eq. (15, 16)
      The HAP transmits the aggregated global model weight to the AeBSs, and each AeBS updates its model based on the received global model weight.
    end if
  end for
end for

Algorithm 1 3D locations of AeBSs with federated deep deterministic policy gradient algorithmThe specific procedure of the Fed-DDPG algorithm is shown in Algorithm 1. To begin with, each AeBS initializes its deep neural networks with random weights. The experience memory, enabling agents to retain and reuse past experiences, is initialized. Afterward, the noise level for random action exploration and the frequency of federated learning are configured. During the training process, the agent generates experience through interactions with the environment, storing it in the experience memory buffer. Subsequently, the experience memory buffer is sampled, and both the actor and critic networks are trained using the sampled experiences. Each AeBS periodically uploads network parameters to the HAP and obtains the aggregated parameters of the HAP. To enhance learning stability, the target network parameters are updated gradually through soft updates.

Then we analyze the computational complexity of the algorithm. In the current scenario, the difference in the computational complexity of different DRL algorithms mainly depends on two parts, the number of network inputs and the communication overhead. Fed-DDPG belongs to the Fed-DRL algorithm and MADDPG belongs to the MADRL algorithm.

1) The number of network inputs: In the MADRL algorithm, all observations and actions of the agents are considered as inputs. The action of AeBS is defined as A, the observation of AeBS is defined as S, and the number of AeBS is defined as M. In the network of each AeBS, the number of network inputs for the MADRL algorithm is $O(M * (A + S))$ and the number of network inputs for the Fed-DRL algorithm is $O(A + S)$.

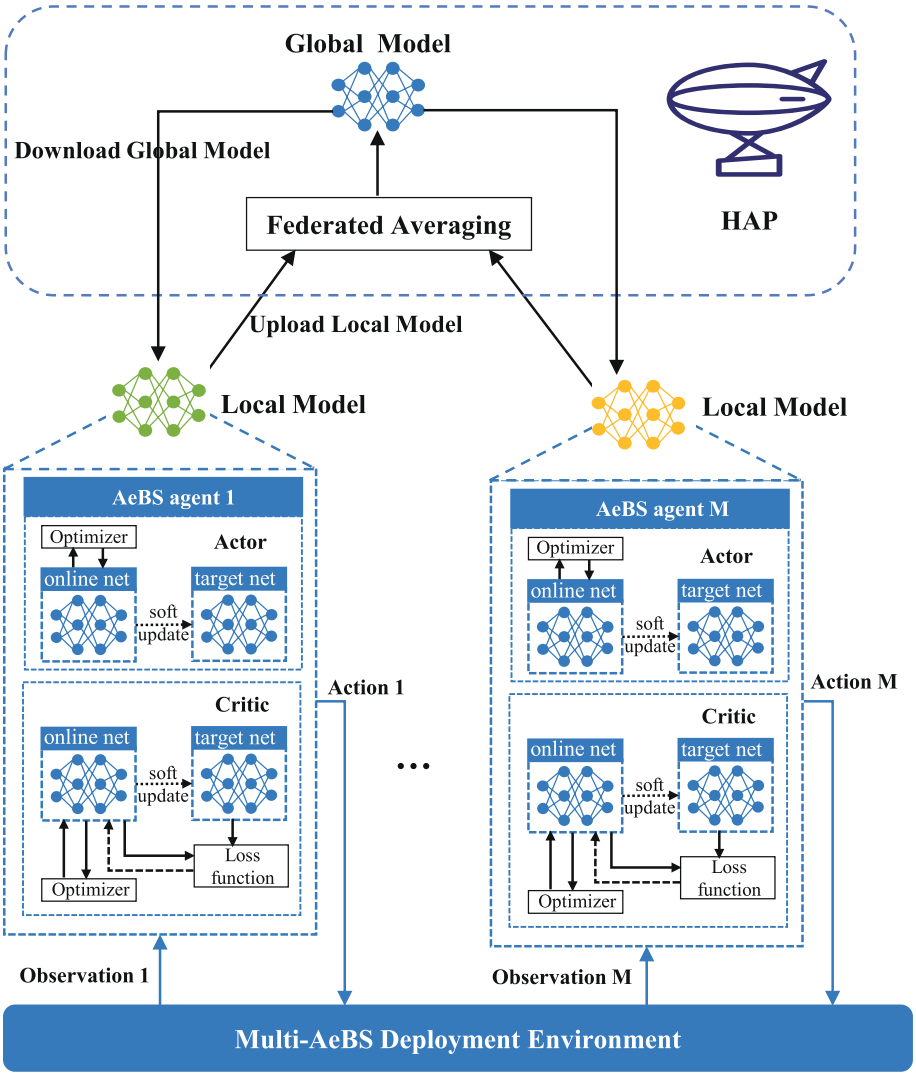

Fig. 2 The Fed-DDPG framework for HAP-assisted multi-AeBS deployment

2) The communication overhead: In the MARL framework, each agent needs to communicate state information to all other agents. This leads to MARL frameworks relying heavily on agents' communication, and federated learning can solve this problem. The observation and action of AeBS are defined as O, the number of bits to store a unit of data is defined as C, and the number of AeBS is defined as M. The overhead in the MADRL algorithm is equal to $(M * O * C)$, and the overhead in the Fed-DRL algorithm is equal to $(O * C)$.

Simulation results and discussions

Simulation is carried out in a 6 km × 6 km urban environment, and the vertical flight range of AeBSs is from 10 m to 200 m. The simulations are conducted in a Python 3.8 environment with torch 1.7.1. The environment's simulation parameters are detailed in Table 1 and the hyperparameters of the Fed-DDPG are listed in Table 2.

We compare our proposed Fed-DDPG with the other four MADRL algorithms: MADDPG, distributed DDPG (Dis-DDPG), MADQN, and Fed-DQN. Among them, Fed-DDPG and Fed-DQN belong to the Fed-DRL algorithm, MADDPG and MADQN belong to the MADRL algorithm. The details of the five algorithms are as follows:

1 Fed-DDPG: Fed-DDPG is an algorithm that introduces federated learning into MADRL. Each AeBS can only observe its own observation and uploads its deep deterministic policy gradient network parameters to Hap for aggregation.

Table 1 The simulation parameters of the environment

Simulation Parameters	Values
Environmental constants α, β	9.61, 0.16
Excessive path loss η_{LoS}, η_{NLoS}	1, 20
Speed of light c	3×10^8 m/s
Transmit power of AeBS P_m	50 dBm
Channel bandwidth B	150 MHz
Carrier frequency	30 GHz
Sector width $\delta_{T,s}, \delta_{R,s}$	$\pi/2, \pi/2$
Beamwidth $\delta_{T,u}, \delta_{R,u}$	$\pi/6, \pi/6$
Pilot duration ratio T_p/T	2×10^{-4}
Side lobe gain $G_{T,S}, G_{R,S}$	-10 dB, -10 dB
Thermal noise power density	-174dBm/Hz
Number of mmWave beams N_b	32

Table 2 The hyperparameters of Fed-DDPG

Network hyperparameters	Values
Layer type of actor	Fully connected
Neurons of hidden layers for actor	[64,64]
Layer type of critic	Fully connected
Neurons of hidden layers for critic	[64,64]
Optimizer	Adam
Activation Function	Leaky ReLU
ϵ in ϵ-greedy	0.1
Memory size	10000
Batch size	512
Discount factor	0.9

2 MADDPG [31]: MADDPG is a MADRL algorithm that is centrally trained and distributively executed. In the current scenario, each AeBS has a DDPG network and the critic networks can obtain the observation of all other agents.
3 Dis-DDPG [38]: Dis-DDPG is a fully distributed algorithm. And each AeBS makes decisions based on its own observation.
4 MADQN [31]: MADQN is a MADRL algorithm. In the current scenario, each AeBS has a deep Q network and obtain the observation of all other agents.
5 Fed-DQN [39]: Fed-DQN is an algorithm that introduces federated learning into MADRL. Each AeBS can only observe its own observation and uploads its deep Q network parameters to HAP for aggregation.

First of all, we depict the learning curve of the proposed Fed-DDPG under different aggregation frequencies of federated learning in Fig. 3. It can be found that different frequencies of federated learning affect the convergence process of Fed-DDPG, with every 10 episodes having the fastest convergence rate and every 100 episodes having the slowest convergence rate. A higher aggregation frequency means that the global model is updated more frequently, which may lead to faster convergence of the model, as the global model can reflect updates from all participants in a more timely manner. On the other hand, a lower aggregation frequency may cause significant fluctuations in the global model between each aggregation step, resulting in less stable training processes and reduced convergence performance. However, it is worth noting that high-frequency aggregation may introduce more training time, as more frequent model uploads and downloads consume more computing resources. This will be discussed in our subsequent simulation.

To assess the impact of different aggregation frequencies of federated learning on the total training time, we conducted a comparative analysis in Fig. 4. The time required for training the model can be divided into two parts: computation time and synchronization time. The computation time represents the duration needed for each AeBS to train the neural network, while the synchronization time accounts for the time taken when each AeBS uploads its parameters to the HAP, followed by HAP performing parameter aggregation and distributing them back to each AeBS. We assumed a fixed time for each upload and download of the model. As illustrated in Fig. 4, the frequency of aggregating different model parameters has a significant impact on the total training time, suggesting the importance of selecting an appropriate communication frequency. When the communication frequency is excessively high, such as communicating every 10 episodes, the synchronization time increases significantly, leading to a substantial increase in the total time. Conversely, when the communication frequency is too low, such as communicating every 100 episodes, the computation time becomes dominant and occupies the majority of the running time. Upon analysis, we find that a communication frequency of every 20 episodes is a suitable choice for our scene. This frequency allows federated learning to effectively expedite the model convergence without incurring excessively high synchronization costs.

Then we analyze the impact of numbers of deep neural network layers on the learning performance, as shown in Fig. 5. It can be found that as the number of layers increases, the performance of the algorithm slightly improves. However, it can also be seen that when the number of layers of the neural network increases, the convergence efficiency of the algorithm is reduced. This is because more layers of deep neural networks usually require more parameters and

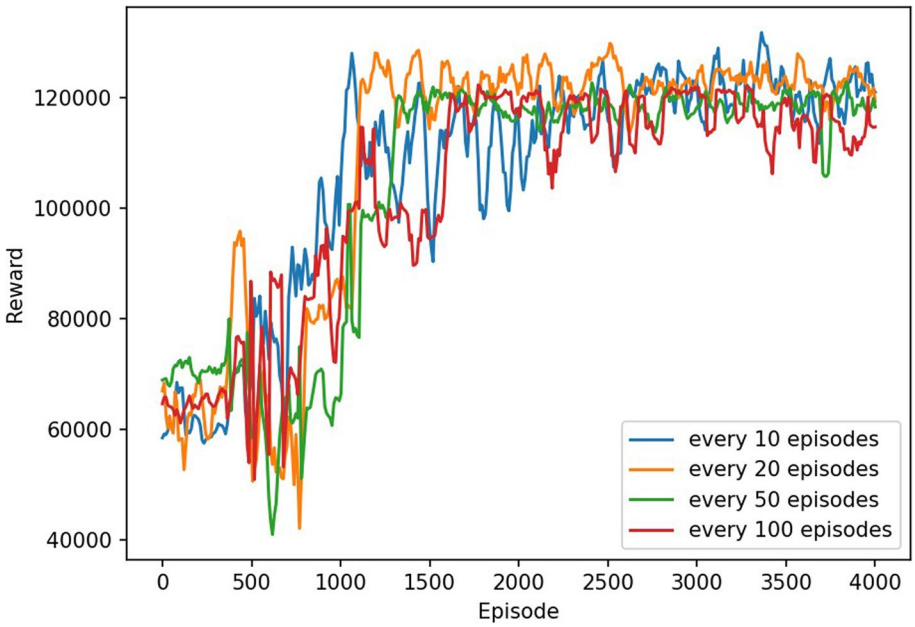

Fig. 3 The reward of Fed-DDPG at different aggregation frequencies of federated learning

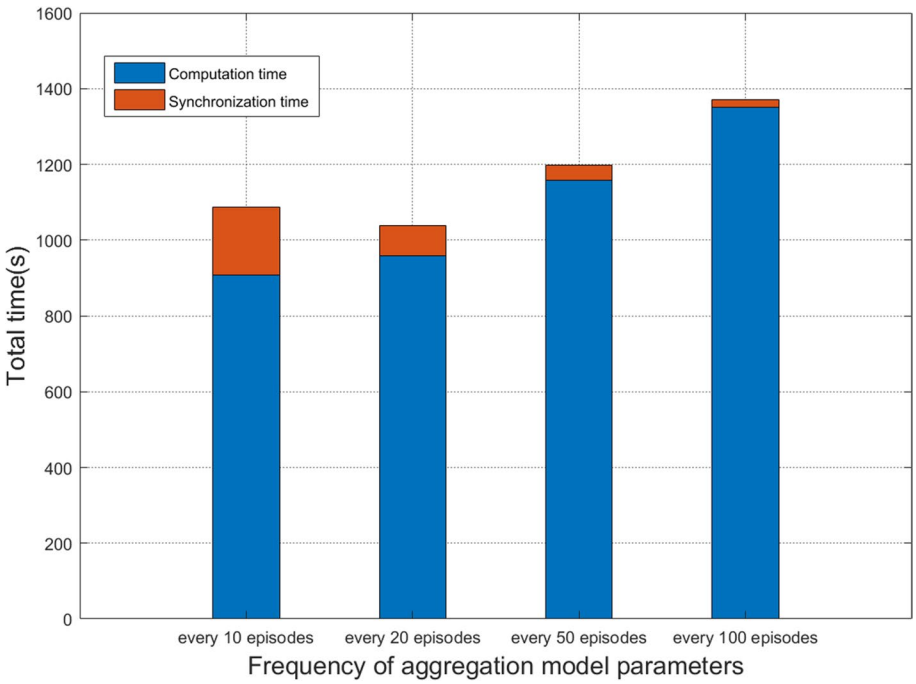

Fig. 4 The total time of training model at different frequencies of federated learning

computational resources, so training time and computational costs may be relatively high, which is not suitable for resource-constrained environments. Therefore, in our scene, it is recommended to set the number of layers of the model to 3 or 4, which can achieve a satisfactory convergence at a faster rate of convergence.

Figure 6 illustrates the influence of the number of AeBSs on the system capacity. Our proposed scheme is compared against four other MADRL algorithms. By comparing MADDPG and Fed-DDPG, MADQN and Fed-DQN, it can be found that federated learning can approach the performance of centralized training

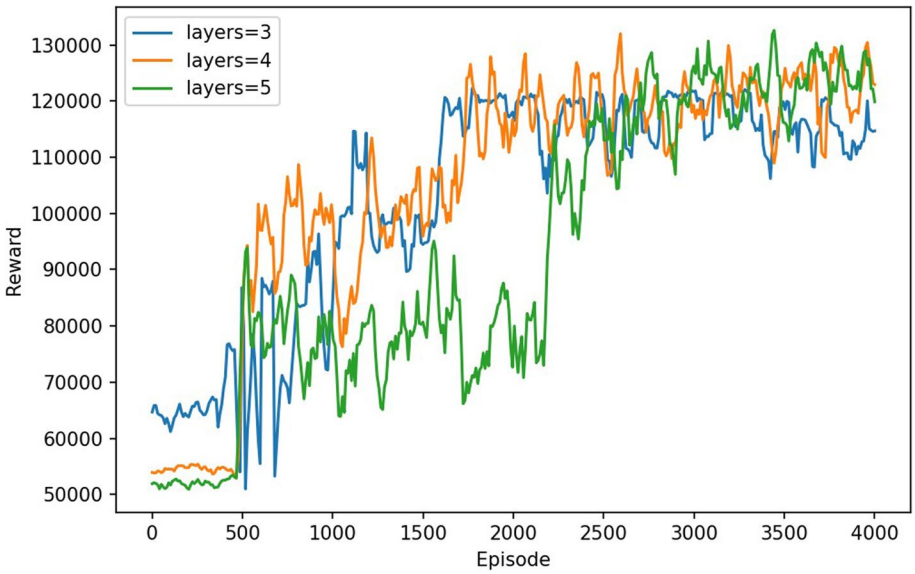

Fig. 5 The reward of Fed-DDPG with different numbers of deep neural network layers

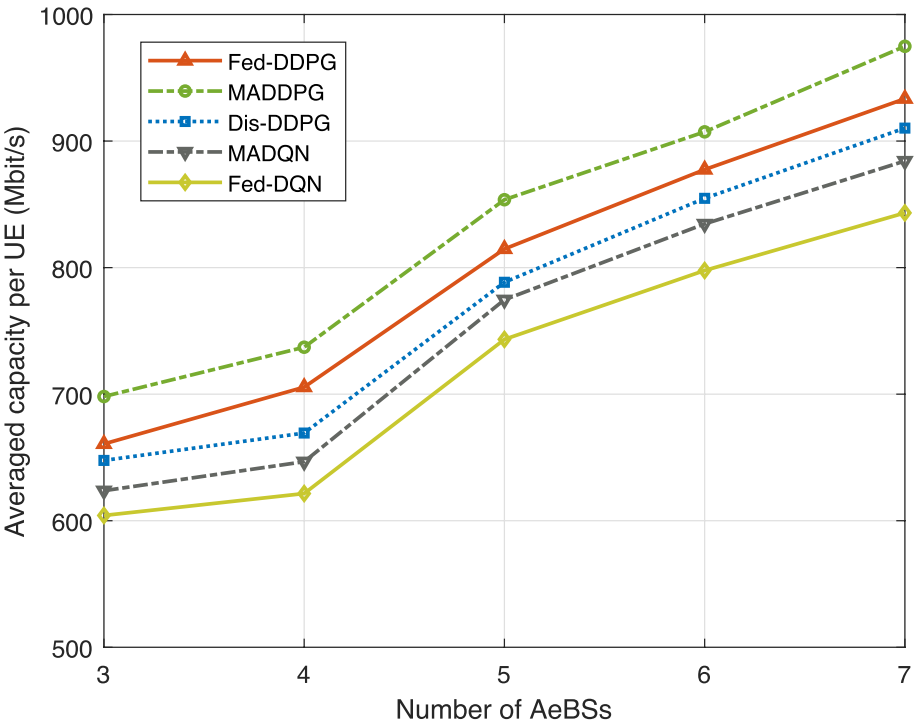

Fig. 6 Averaged capacity for different numbers of AeBSs

algorithms that know the global observations. Under the different number of AeBSs, the Fed-DDPG algorithm can reach at least 94.6% of the MADDPG algorithm's performance, and the highest can reach 96.7% of the MADDPG algorithm's performance. Fed-DDPG outperforms Dis-DDPG which treats each AeBS training as an independent operation and makes decisions based on its own observation. In addition, with the increase of AeBSs, the average capacity increases gradually. This is because when the number of AeBSs increases, the communication resources of the system increase, and the capacity of each UE increases.

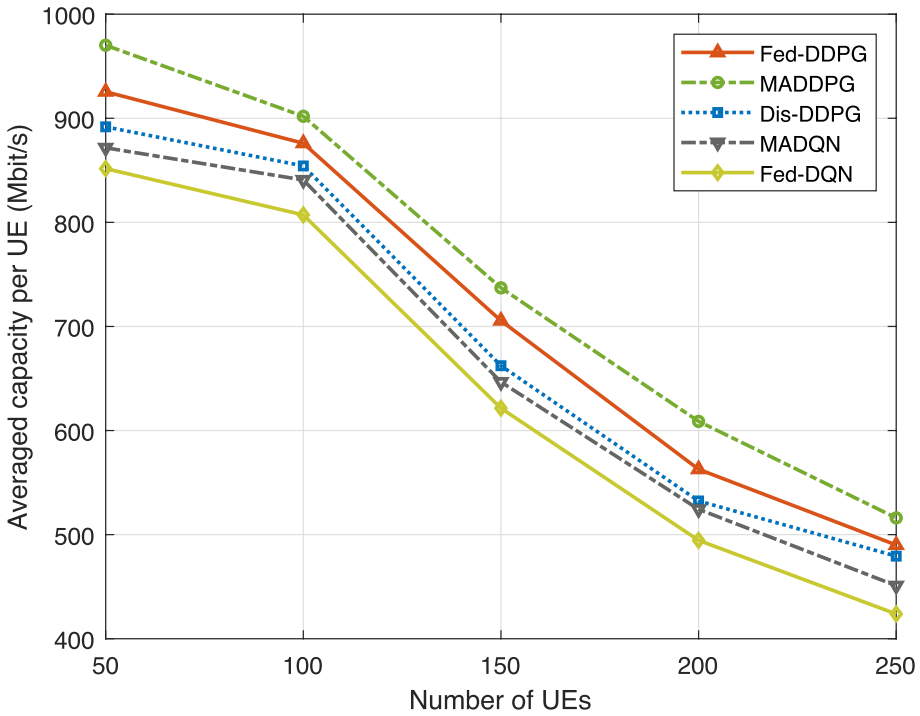

Fig. 7 Averaged capacity for different numbers of UEs

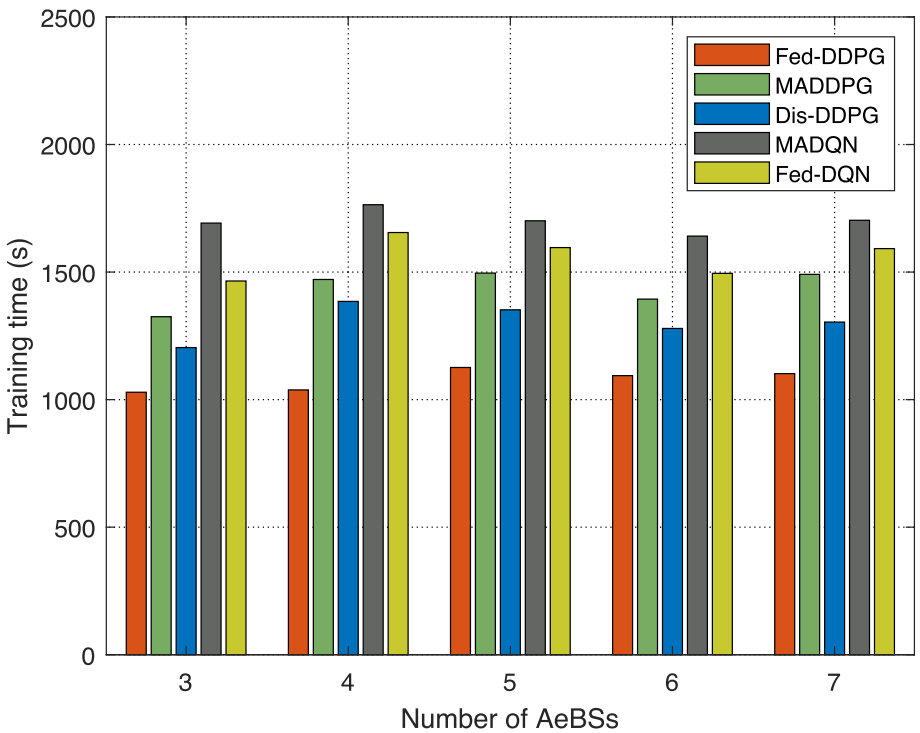

Fig. 8 Training time of different DRL algorithms with different numbers of AeBSs

Figure 7 presents the impact of the number of UEs on the system capacity and UEs represent the total number of users served by AeBSs. Under the different number of users, the Fed-DDPG algorithm can reach at least 92.4% of the MADDPG algorithm's performance, and the highest can reach 97.1% of the MADDPG algorithm's performance. Moreover, the average UE capacity of the Fed-DDPG algorithm is better than that of the completely independent distributed DDPG. From Figs. 6 and 7, it can be found that the performance of the algorithm based on DDPG is better than the algorithm based on DQN. This is because DDPG has the following two advantages compared with DQN in the current scenario: Firstly, the discrete action space is extended to the continuous action space. Secondly, the Actor-Critic framework is introduced to better explore the environment and train the model.

Figure 8 shows the training time of different MADRL algorithms with different numbers of AeBSs. By comparing Figs. 6, 7, and 8, it can be found that although the performance of the MADDPG algorithm is slightly better than Fed-DDPG, Fed-DDPG can save a lot of training computation time. Compared with the MADDPG algorithm that knows the global information, the Fed-DDPG algorithm can save at least 21% of the training time and up to 29.4% of the training time. Fed-DDPG also has better training speed than completely independent distributed DDPG. In addition, Fed-DQN also has less training time than MADQN. The training time of Fed-DRL is lower than MADRL. This is because Fed-DRL has fewer network inputs and lower communication overhead than MADRL according to the analysis of algorithm computational complexity in Section Federated deep deterministic policy gradient algorithm for AeBSs.

Based on the previous simulation results comparing Fed-DDPG and MADDPG on algorithm performance and training time, we observe that the introduction of federated learning into MADRL can significantly accelerate convergence. Furthermore, the performance of Fed-DDPG can approximate MADDPG algorithms but with less training time. The shorter training time means that AeBSs can be deployed faster, and the reduced information exchange between different AeBS means that AeBS can save communication resources and protect privacy. Considering the goals of rapid multi-AeBS deployment and privacy preservation, our Fed-DDPG algorithm proves to be a more suitable for the scene.

Conclusion

In this article, the 3D deployment of multi-AeBS is modeled as the maximum system capacity problem, and the Fed-DDPG algorithm is designed to solve the problem. Inspired by the federated learning framework, we introduce a HAP-assisted multi-AeBS deployment architecture, where low-altitude AeBSs train their individual local models, and the HAP serves as a global node responsible for model aggregation. Simulation results show that the proposed Fed-DDPG achieves close performances to the MADDPG algorithm which knows the global information with less training time. The optimal aggregation frequency in the algorithm considering the time of uploading and downloading models is also discussed through simulations. However, there is still scope for enhancing the method proposed in this article. For instance, the strategy aggregation of the proposed approach relies on the FedAvg method, which might face challenges when dealing with disparities in computing resources, storage capacity, and data distribution among different AeBSs. In our future work, we plan to develop better-performing aggregation techniques that can cater to the requirements of heterogeneous application scenarios in aerial cloud computing networks.

Authors' contributions

Lei Liu proposed the main idea and principles of the research and sketched the manuscript. Haoran He designed and implemented the algorithms and experiment schemes and drafted the technical part. Fei Qi guided the design of the algorithms and experiment, and prepared the final manuscript for submission. Yikun Zhao helped with setting up experiment environment, including the illustrative figures. Weiliang Xie was responsible for data visualization and prepared the analytical figures. Fanqin Zhou investigated and drafted the background and related research part of the manuscript. Lei Feng refined the whole text of the manuscript, and help with preparing the final manuscript for submission. All the authors reviewed the manuscript.

Funding

This research was funded by National Key R&D Program of China (No. 2020YFB1806700).

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 7 June 2023 Accepted: 16 August 2023
Published online: 29 September 2023

References

1. Pham QV, Ruby R, Fang F, Nguyen DC, Yang Z, Le M, Ding Z, Hwang WJ (2022) Aerial computing: A new computing paradigm, applications, and challenges. IEEE Internet Things J 9(11):8339–8363. https://doi.org/10.1109/JIOT.2022.3160691
2. Jia Y, Liu B, Dou W, Xu X, Zhou X, Qi L, Yan Z (2022) CroApp: A CNN-based resource optimization approach in edge computing environment. IEEE Trans Ind Inf 18(9):6300–6307. https://doi.org/10.1109/TII.2022.3154473

3. Yang C, Xu X, Zhou X, Qi L (2022) Deep Q network–driven task offloading for efficient multimedia data analysis in edge computing–assisted IoV. ACM Trans Multimedia Comput Commun Appl 18(2s). https://doi.org/10.1145/3548687
4. Xu X, Fang Z, Qi L, Zhang X, He Q, Zhou X (2021) TripRes: Traffic flow prediction driven resource reservation for multimedia IoV with edge computing 17(2). https://doi.org/10.1145/3401979
5. Qi L, Lin W, Zhang X, Dou W, Xu X, Chen J (2023) A correlation graph based approach for personalized and compatible web APIs recommendation in mobile APP development. IEEE Trans Knowl Data Eng 35(6):5444–5457. https://doi.org/10.1109/TKDE.2022.3168611
6. Wang Y, Qi L, Dou R, Shen S, Hou L, Liu Y, Yang Y, Kong L (2023) An accuracy-enhanced group recommendation approach based on DEMATEL. Pattern Recogn Lett 167:171–180. https://doi.org/10.1016/j.patrec.2023.02.008. https://www.sciencedirect.com/science/article/pii/S0167865523000284
7. Xu Y, Feng Z, Zhou X, Xing M, Wu H, Xue X, Chen S, Wang C, Qi L (2023) Attention-based neural networks for trust evaluation in online social networks. Inf Sci 630:507–522. https://doi.org/10.1016/j.ins.2023.02.045. https://www.sciencedirect.com/science/article/pii/S0020025523002396
8. Wu S, Shen S, Xu X, Chen Y, Zhou X, Liu D, Xue X, Qi L (2023) Popularity-aware and diverse web APIs recommendation based on correlation graph. IEEE Trans Comput Soc Syst 10(2):771–782. https://doi.org/10.1109/TCSS.2022.3168595
9. Li Z, Xu X, Hang T, Xiang H, Cui Y, Qi L, Zhou X (2022) A knowledge-driven anomaly detection framework for social production system. IEEE Trans Comput Soc Syst pp 1–14. https://doi.org/10.1109/TCSS.2022.3217790
10. Xu X, Gu J, Yan H, Liu W, Qi L, Zhou X (2023) Reputation-aware supplier assessment for blockchain-enabled supply chain in Industry 4.0. IEEE Trans Ind Inf 19(4):5485–5494. https://doi.org/10.1109/TII.2022.3190380
11. Jenila C, Jeyachitra R (2021) Green indoor optical wireless communication systems: Pathway towards pervasive deployment. Digit Commun Netw 7(3):410–444. https://doi.org/10.1016/j.dcan.2020.09.004. https://www.sciencedirect.com/science/article/pii/S2352864820302650
12. Santipach W, Jiravanstit K (2022) On selecting transmission mode for D2D transmitter in underlay cellular network with a multi-antenna base station. Digit Commun Netw 8(2):194–207. https://doi.org/10.1016/j.dcan.2021.06.006. https://www.sciencedirect.com/science/article/pii/S2352864821000341
13. Wu J, Li Y, Zhuang H, Pan Z, Wang G, Xian Y (2021) SMDP-based sleep policy for base stations in heterogeneous cellular networks. Digit Commun Netw 7(1):120–130. https://doi.org/10.1016/j.dcan.2020.04.010. https://www.sciencedirect.com/science/article/pii/S2352864819304092
14. Xiao Z, Xia P, Xia XG (2016) Enabling UAV cellular with millimeter-wave communication: Potentials and approaches. IEEE Commun Mag 54(5):66–73
15. Dai H, Yu J, Li M, Wang W, Liu AX, Ma J, Qi L, Chen G (2022) Bloom filter with noisy coding framework for multi-set membership testing. IEEE Trans Knowl Data Eng 1–14. https://doi.org/10.1109/TKDE.2022.3199646
16. He Q, Tan S, Chen F, Xu X, Qi L, Hei X, Zomaya A, Hai J, Yang Y (2023) EDIndex: Enabling fast data queries in edge storage systems. ACM SIGIR
17. Wang Y, Wang J, Zhang W, Zhan Y, Guo S, Zheng Q, Wang X (2022) A survey on deploying mobile deep learning applications: A systemic and technical perspective. Digit Commun Netw 8(1):1–17. https://doi.org/10.1016/j.dcan.2021.06.001. https://www.sciencedirect.com/science/article/pii/S2352864821000298
18. Oubbati OS, Atiquzzaman M, Lakas A, Baz A, Alhakami H, Alhakami W (2021) Multi-UAV-enabled AoI-aware WPCN: A multi-agent reinforcement learning strategy. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, pp 1–6
19. Liu Z, Garg N, Ratnarajah T (2023) Multi-agent federated reinforcement learning strategy for mobile virtual reality delivery networks. IEEE Trans Netw Sci Eng 1–14. https://doi.org/10.1109/TNSE.2023.3292570
20. Zhang S, Wang Z, Zhou Z, Wang Y, Zhang H, Zhang G, Ding H, Mumtaz S, Guizani M (2022) Blockchain and federated deep reinforcement learning based secure cloud-edge-end collaboration in power IoT. IEEE Wirel Commun 29(2):84–91. https://doi.org/10.1109/MWC.010.2100491
21. Xu J, Ota K, Dong M (2022) Aerial edge computing: Flying attitude-aware collaboration for multi-UAV. IEEE Trans Mob Comput 1. https://doi.org/10.1109/TMC.2022.3179399
22. Jeong S, Simeone O, Kang J (2017) Mobile cloud computing with a UAV-mounted cloudlet: optimal bit allocation for communication and computation. IET Commun 11(7):969–974
23. Khan SK, Naseem U, Siraj H, Razzak I, Imran M (2021) The role of unmanned aerial vehicles and mmWave in 5G: Recent advances and challenges. Trans Emerg Telecommun Technol 32(7):4241
24. Gapeyenko M, Petrov V, Moltchanov D, Yeh SP, Himayat N, Andreev S (2020) Comparing capacity gains of static and UAV-based millimeter-wave relays in clustered deployments. In: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), pp 1–7. https://doi.org/10.1109/ICCWorkshops49005.2020.9145216
25. Xiao Z, Dong H, Bai L, Wu DO, Xia XG (2020) Unmanned aerial vehicle base station (UAV-BS) deployment with millimeter-wave beamforming. IEEE Internet Things J 7(2):1336–1349. https://doi.org/10.1109/JIOT.2019.2954620
26. Liu Y, Huangfu W, Zhou H, Zhang H, Liu J, Long K (2022) Fair and energy-efficient coverage optimization for UAV placement problem in the cellular network. IEEE Trans Commun 70(6):4222–4235. https://doi.org/10.1109/TCOMM.2022.3170615
27. Zou C, Li X, Liu X, Zhang M (2021) 3D placement of unmanned aerial vehicles and partially overlapped channel assignment for throughput maximization. Digit Commun Networks 7(2):214–222. https://doi.org/10.1016/j.dcan.2020.07.007. https://www.sciencedirect.com/science/article/pii/S2352864820302479
28. Yu P, Guo J, Huo Y, Shi X, Wu J, Ding Y (2020) Three-dimensional aerial base station location for sudden traffic with deep reinforcement learning in 5G mmWave networks. Int J Distrib Sensor Netw 16(5):1550147720926374. https://doi.org/10.1177/1550147720926374
29. Wang Q, Zhang W, Liu Y, Liu Y (2019) Multi-UAV dynamic wireless networking with deep reinforcement learning. IEEE Commun Lett 23(12):2243–2246. https://doi.org/10.1109/LCOMM.2019.2940191
30. Wang W, Lin Y (2021) Trajectory design and bandwidth assignment for UAVs-enabled communication network with multi-agent deep reinforcement learning. In: 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), IEEE, pp 1–6
31. Zhao Y, Zhou F, Feng L, Li W, Yu P (2022) MADRL-based 3D deployment and user association of cooperative mmWave aerial base stations for capacity enhancement. Chin J Electron 32(2):283–294. https://doi.org/10.23919/CJE.2021.00.327
32. Pham QV, Zeng M, Huynh-The T, Han Z, Hwang WJ (2022) Aerial access networks for federated learning: Applications and challenges. IEEE Netw 36(3):159–166. https://doi.org/10.1109/MNET.013.2100311
33. Jing Y, Qu Y, Dong C, Ren W, Shen Y, Wu Q, Guo S (2023) Exploiting UAV for air-ground integrated federated learning: A joint UAV location and resource optimization approach. IEEE Trans Green Commun Netw 1. https://doi.org/10.1109/TGCN.2023.3242999
34. Zhang X, Peng M, Yan S, Sun Y (2020) Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. IEEE Internet Things J 7(7):6380–6391. https://doi.org/10.1109/JIOT.2019.2962715
35. Wang X, Wang C, Li X, Leung V, Taleb T (2020) Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching. IEEE Internet Things J 7(10):9441–9455. https://doi.org/10.1109/JIOT.2020.2986803
36. Al-Hourani A, Kandeepan S, Lardner S (2014) Optimal LAP altitude for maximum coverage. IEEE Wirel Commun Lett 3(6):569–572. https://doi.org/10.1109/LWC.2014.2342736
37. Zhu L, Zhang J, Xiao Z, Cao X, Wu DO, Xia XG (2019) 3D beamforming for flexible coverage in millimeter-wave UAV communications. IEEE Wirel Commun Lett 8(3):837–840. https://doi.org/10.1109/LWC.2019.2895597
38. Ciftler BS, Alwarafy A, Abdallah M (2022) Distributed DRL-based downlink power allocation for hybrid RF/VLC networks. IEEE Photon J 14(3):1–10. https://doi.org/10.1109/JPHOT.2021.3139678
39. Nie Y, Zhao J, Gao F, Yu FR (2021) Semi-distributed resource management in UAV-aided MEC systems: A multi-agent federated reinforcement learning approach. IEEE Trans Veh Technol 70(12):13:162–13,173. https://doi.org/10.1109/TVT.2021.3118446

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.