UAV-Assisted MEC for Disaster Response: Stackelberg Game-Based Resource Optimization

Yafei Guo[1], Ziye Jia[1], Lei Zhang[1], Jia He[1], Yu Zhang[2], and Qihui Wu[1]

[1]The Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 211106, China

[2]Department of Electronic and Electrical Engineering, University College London, London WC1E 6BT, United Kingdom

{guoyafei, jiaziye, Zhang_lei, 071940128hejia, wuqihui}@nuaa.edu.cn, uceezhd@ucl.ac.uk

***Abstract*—The unmanned aerial vehicle assisted multi-access edge computing (UAV-MEC) technology has been widely applied in the sixth-generation era. However, due to the limitations of energy and computing resources in disaster areas, how to efficiently offload the tasks of damaged user equipments (UEs) to UAVs is a key issue. In this work, we consider a multiple UAV-MECs assisted task offloading scenario, which is deployed inside the three-dimensional corridors and provide computation services for UEs. In detail, a ground UAV controller acts as the central decision-making unit for deploying the UAV-MECs and allocates the computational resources. Then, we model the relationship between the UAV controller and UEs based on the Stackelberg game. The problem is formulated to maximize the utility of both the UAV controller and UEs. To tackle the problem, we design a K-means based UAV localization and availability response mechanism to pre-deploy the UAV-MECs. Then, a chess-like particle swarm optimization probability based strategy selection learning optimization algorithm is proposed to deal with the resource allocation. Finally, extensive simulation results verify that the proposed scheme can significantly improve the utility of the UAV controller and UEs in various scenarios compared with baseline schemes.**

***Index Terms*—UAV-assisted MEC, Stackelberg game, improved particle swarm optimization algorithm.**

I. Introduction

With the rapid development of sixth generation communications, multi-access edge computing (MEC) technique is raised to provide computing services for various user equipments (UEs) [1].However, for the UEs in the disaster-stricken area, the ground base stations (BSs) with MEC devices are damaged and unable to provide timely services. Due to the high flexibility and mobility, UAV-MEC can serve as aerial BSs to provide timely and efficient services [2], [3]. However, there exist security issues and resource limitations for the aerial deployment of UAV-MEC.

There are several recent works related to UAVs that address the above issues. In [4], the optimizing of three-dimensional (3D) trajectory of UAV-MEC is considered to ensure the safety. The 3D air corridor defines the flight path of UAVs, reducing collision risks and adding integrations with local airspace management systems [5]. The remote identification (Remote ID) is a new communication system for ensure flight safety by obtaining the information of the UAV [6]. In [7], a cooperative cognitive dynamic system is proposed to optimize the management of UAVs. In addition, for the energy consumption of UAV-MEC, since UAV-MEC has a competitive relationship with UEs, the game theory is an effective method to solve such problems [8]. For instance, authors in [9] utilize the Stackelberg game to maximize the profit of the UAV considering the delay, energy consumption and urgency. The Potential game is proposed to solve the optimization problem for UAV trajectory planning, resource management and task offloading strategy in [10]. However, these works lack unified considerations of the safety and efficiency of UAV-MEC.

In this work, we consider a task offloading scenario, where the UAV-MECs are deployed in the 3D corridor and provide computation services for damaged UEs. A ground UAV controller serves as the central decision-making unit, implementing UAV-MEC deployment and resource allocation strategies through the Remote ID. To maximize the utilities, the UEs and the UAV controller are formulated as a Stackelberg game problem. Then, we design a UAV localization and availability response mechanism (ULAR) based on K-means to pre-deploy the UAVs and enable more efficient use of computing resources. Further, we propose a chess-like particle swarm optimiza-tion probability-based strategy selection learning optimization algorithm (CPPO) to deal with the resource allocation. Finally, we conduct extensive simulations to verify the effectiveness of the proposed methods.

This paper is organized as follows. In Section II, we present the system model and problem formulation. Section III analyzes the optimization problem. Algorithms are designed in Section IV. Section V conducts simulations and analyzes the results. Finally, conclusions are drawn in Section VI.

II. System Model And Problem Formulation

A. Network Model

As shown in Fig. 1, we consider a UAV-MEC assisted task offloading scenario, where UAVs are deployed at the 3D

This work was supported in part by National Natural Science Foundation of China under Grant 62301251, in part by the Natural Science Foundation on Frontier Leading Technology Basic Research Project of Jiangsu under Grant BK20222001, in part by the Aeronautical Science Foundation of China 2023Z071052007, and in part by the Young Elite Scientists Sponsorship Program by CAST 2023QNRC001.

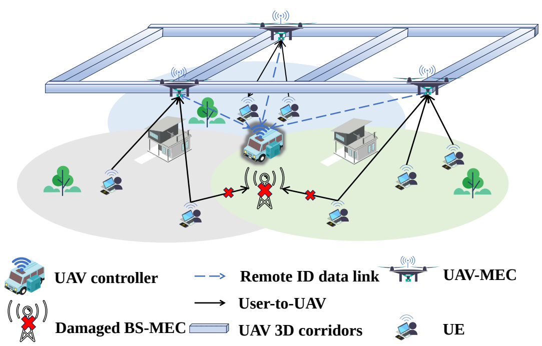


Fig. 1. Multiple UAV-MECs assisted task offloading network.

corridors [11], [12], and provide computing services for UEs in place of the Damaged BS-MEC that has lost communication and computing power. In order to achieve the efficient collaborative operation of UAVs and MEC, the UAV controller deploys UAV-MEC through the precise Remote ID communication technology. In detail, all J UAVs denoted as $\mathcal{J} = \{1, 2, \dots, J\}$ are dispatched by the UAV controller. Further, I UEs are distributed in the disaster area denoted as $\mathcal{I} = \{1, 2, \dots, I\}$, and the computation task generated by UE i is represented as a two-tuple, $T_i = \{G_i, g_i\}$. Wherein, G_i and g_i are the size of all data as well as the size of data offloaded to the UAV-MEC, respectively. Let $\{p_i, \varepsilon_i\}$ denote the local computing power and unit energy consumption of UE i, respectively. The characteristics of UAV j are described as $\{f_j, P_j^{\text{comp}}\}$, where f_j represents the total computation resources owned by UAV j, and P_j^{comp} represents the CPU power of the UAV-MEC server.

The 3D Cartesian coordinates are utilized to represent the positions of UAVs and UEs. The location of UAV j and UE i are denoted as $z_j = (x_j, y_j, h_j)$ and $z_i = (x_i, y_i, 0)$, respectively. Thus, the distance between UAV j and UE i is claculated as $d_{i,j} = \sqrt{\|z_i - z_j\|^2}$.

B. Communication Model

Considering that the UAVs are deployed in the 3D corridors with a constant height of H, we assume that the task offloading link between UEs and UAVs can be modeled as a line-of-sight (LoS). The uplink model from UE i to UAV j is expressed as:

$$r_{i,j} = B \log_2 \left(1 + \frac{q_i h_{i,j}}{\sigma^2}\right), \tag{1}$$

where B is the bandwidth allocated by UE i to UAV j, σ^2 represents the noise power and q_i is the transmission power of UE i. The channel gain can be expressed as $h_{i,j} = d_{i,j}^{-\rho}$, where ρ denotes the path loss factor between UEs and UAVs.

C. Utility of UE and UAV Controller

1) Utility of UE: The uplink transmission time from UE i to UAV j can be calculated as $t_{i,j}^{\text{trans}} = \frac{g_i}{r_{i,j}}$. Then, the uplink transmission energy is $E_{i,j}^{\text{trans}} = p_i t_{i,j}^{\text{trans}}$. The local computation energy consumption of UE i is $E_{\text{local}}^{\text{comp}} = \varepsilon_i (G_i - g_i)$. Therefore, the utility of UE i is

$$U_i = S(g_i)\delta_i - E_{i,j}^{\text{trans}} - E_{\text{local}}^{\text{comp}} - \lambda_i g_i, \tag{2}$$

where $S(g_i) = \ln(1 + g_i)$ is the satisfaction function that reflects the satisfaction degree of UE. δ_i is the control coefficient to assess the impact of UE satisfaction on its utility. λ_i represents the price of each unit of data offloaded by the UAV-MEC server for each UE.

2) Utility of UAV Controller: In face of simultaneous offloading of multiple UEs, the UAV-MEC server adopts an equal allocation strategy of computing power [13]. We assume that there are M_j UEs offloading tasks to the UAV j simultaneously. The computing time of UAV j for the offloading data of UE i can be calculated as $t_{i,j}^{\text{comp}} = \frac{\alpha g_i}{f_{ij}}$, where $f_{ij} = f_j / M_j$ represents UE i obtains the computing resource from UAV j, and α is a coefficient related to data encoding.

Therefore, the computing energy consumption of UAV-MEC server j for UE i is $E_{i,j}^{\text{comp}} = P_j^{\text{comp}} t_{i,j}^{\text{comp}}$. The energy consumption equation of UAV j is $E_j^{\text{comp}} = \sum_{i=1}^{N} X_{i,j} E_{i,j}^{\text{comp}}$, where variable $X_{i,j} \in \{0, 1\}$ represents the link status between UE i and UAV j. When $X_{i,j} = 1$, it indicates that UE i has establishes a connection with UAV j, enabling the allocation of computing resources. Conversely, $X_{i,j} = 0$ means there exists no connection between UE i and UAV j. The hovering energy consumption is denoted by $E_j^{\text{hov}} = P_j^{\text{hov}} / \eta$, where P_j^{hov} indicates the minimum power for hovering, and η represents the power efficiency [14]. Therefore, the utility of the UAV controller can be calculated as

$$U_{con} = \sum_{i=1}^{N} \lambda_i g_i - \sum_{j=1}^{J} E_j^{\text{comp}} - \sum_{j=1}^{J} E_j^{\text{hov}}. \tag{3}$$

D. Problem Formulation

The optimization goal is to maximize the utility of both the UAV controller and UEs. The optimization problem for UE i is

$$\begin{aligned}
\mathbf{P0}\colon \max_{g_i, \lambda_i, X_{i,j}} \quad & U_i, \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} X_{i,j} \leq 1, \forall i \in \mathcal{N}, && (4)\\
& g_i \in [0, G_i], \forall g_i \in \boldsymbol{g}, && (5)\\
& \lambda_i \in [\lambda_i^{min}, \lambda_i^{max}], \forall \lambda_i \in \boldsymbol{\lambda}, && (6)\\
& X_{i,j} \in \{0, 1\}, \forall i \in \mathcal{N}, \forall j \in \mathcal{J}, && (7)
\end{aligned}$$

where constraint (4) denotes that UE i can only select one UAV from set $\mathcal{J}$ to offload tasks. Constraint (5) indicates that the data amount of tasks offloaded by UE i is no more than the data amount of the entire task T_i. Constraint (6) indicates the resource price range of the UAV controller, where λ_i^{min} and λ_i^{max} are the minimum and maximum prices, respectively. Constraint (7) indicates that UE i has the option to decide whether or not to offload the task to UAV j.

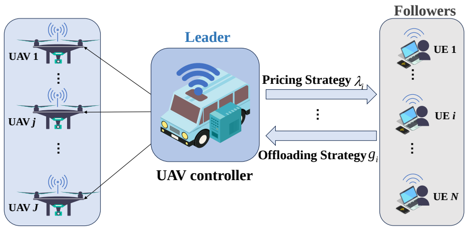


Fig. 2. The Stackelberg game procedure.

The optimization problem for the UAV controller is:

$$\begin{aligned} \mathbf{P1}: \ &\max_{\boldsymbol{g},\boldsymbol{\lambda},X_{i,j}} \quad U_{con}, \\ &\text{s.t. } E_j^{\text{comp}} + E_j^{\text{hov}} \leq \varepsilon, \forall j \in \mathcal{J}, \\ &(5), (6), \end{aligned} \tag{8}$$

where constraint (8) indicates that the energy consumption can not exceed the UAV battery budget.

III. GAME THEROY BASED PROBLEM ANALYSIS

A. Stackelberg Game Model

Since the UAV controller has a competitive relationship with UEs, the interaction between UEs and the UAV controller is modeled as a Stackelberg game with a single leader and multiple followers, as depicted in Fig. 2. The game is described in two stages. In the first stage, each UE submits the task information and the UAV controller sets the resource prices $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_I\}$ based on the information provided by the UEs. In the second stage, each UE determines the data offloading strategy $\boldsymbol{g} = \{g_1, g_2, \ldots, g_I\}$, according to the specified pricing scheme.

B. Optimization of UE

We apply a backward induction method to deal with the game-theoretic problem. The proof of the existence and uniqueness of Nash equilibrium is given as follows.

Definition 1. *There exists Nash equilibrium among EUs with $\boldsymbol{g^*} = \{g_1^*, g_2^*, ..., g_N^*\}$. At this point, there is a utility function $U_i\left(g_i^*, g_{-i}^*\right) > U_i\left(g_i, g_{-i}^*\right)$, where g_{-i}^* is the best strategy for other UEs excluding EU i.*

Theorem 1. *In the Stackelberg game, there exists a unique Nash equilibrium point when the utility function of UE i adheres to Eq. (2). In this case, the optimal offloading strategy of UE i is denoted by*

$$g_i^* = \frac{\delta_i}{\frac{p_i}{r_{i,j}} - \varepsilon_i + \lambda_i} - 1. \tag{9}$$

Proof. *The first and second partial derivatives of the utility function U_i with respect to g_i can be obtained as follows*

$$\frac{\partial U_i}{\partial g_i} = \frac{\delta_i}{1+g_i} - \frac{p_i}{r_{i,j}} + \varepsilon_i - \lambda_i, \tag{10}$$

and

$$\frac{\partial U_i^2}{\partial g_i^2} = -\frac{\delta_i}{\left(1+g_i\right)^2}. \tag{11}$$

Since $\delta_i > 0$ and $1 + g_i > 0$, we can obtain that the second derivative is less than zero. The utility function $U_i(g_i)$ of UE i is strictly concave. Therefore, the Nash equilibrium solution exists. When $\partial U_i/\partial g_i = 0$, we obtain the optimal data offloading strategy for UE. □

Since $g_i \in [0, G_i]$, we can obtain the threshold for the pricing strategy of the UAV controller as: $\lambda_i^{min} = \frac{\delta_i}{(1+G_i)} - \frac{p_i}{r_{i,j}} + \varepsilon_i$, and $\lambda_i^{max} = \delta_i + \varepsilon_i - \frac{p_i}{r_{i,j}}$. Therefore, the optimal offloading strategy of UE i can be expressed as:

$$g_i^* = \begin{cases} G_i, & \lambda_i \leq \lambda_i^{min}, \\ \frac{\delta_i}{\frac{p_i}{r_{i,j}} - \varepsilon_i + \lambda_i} - 1, & \lambda_i^{min} < \lambda_i < \lambda_i^{max}, \\ 0, & \lambda_i \geq \lambda_i^{max}. \end{cases} \tag{12}$$

C. Optimization of the UAV Controller Server

Definition 2. *If $U_{con}(\lambda_i^*, g_i^*) > U^{edge}(\lambda_i, g_i^*)$, a unique Stackelberg equilibrium is proposed between UEs and the UAV controller.*

Theorem 2. *In the Stackelberg game, there exists a unique Nash equilibrium point when the utility function of UAV controller adheres to Eq. (3). Similarly, the pricing strategy optimal for the UAV is calculated by*

$$\lambda_i^* = \frac{\sqrt{r_{i,j}\delta_i(p_i - r_{i,j}\varepsilon_i + r_{i,j}\frac{\alpha \cdot M_j P_j^{comp}}{f_j}X_{i,j})} - p_i + r_{i,j}\varepsilon_i}{r_{i,j}}. \tag{13}$$

Proof. *By putting g_i^* into U_{con}, we can obtain $U_{con}(\lambda_i, g_i^*)$ as*

$$\begin{aligned} U_{con}(\lambda_i, g_i^*) &= \sum_{i=1}^{N} \lambda_i (\frac{\delta_i r_{i,j}}{p_i - \varepsilon_i r_{i,j} + \lambda_i r_{i,j}} - 1) \\ &- \sum_{j=1}^{J}\sum_{i=1}^{N} \frac{\alpha \cdot M_j}{f_j} (\frac{\delta_i r_{i,j}}{p_i - \varepsilon_i r_{i,j} + \lambda_c r_{i,j}} - 1) X_{i,j}. \end{aligned} \tag{14}$$

The first and second partial derivatives of the utility function U_{con} with respect to λ_i can be obtained as

$$\frac{\partial U_{con}}{\partial \lambda_i} = \frac{r_{i,j}\delta_i(p_i - r_{i,j}\varepsilon_i + r_{i,j}\frac{\alpha \cdot M_j P_j^{comp}}{f_j}X_{i,j})}{(p_i - r_{i,j}\varepsilon_i + r_{i,j}\lambda_i)^2} - 1, \tag{15}$$

and

$$\frac{\partial U_{con}^2}{\partial \lambda_i^2} = -\frac{2r_{i,j}^2\delta_i(p_i - r_{i,j}\varepsilon_i + r_{i,j}\frac{\alpha \cdot M_j P_j^{comp}}{f_j}X_{i,j})}{(p_i - r_{i,j}\varepsilon_i + r_{i,j}\lambda_i)^3}. \tag{16}$$

Since g_i^* is no less than 0, we can obtain $\frac{p_i}{r_{i,j}} - \varepsilon_i + \lambda_i > 0$. Therefore the second derivative is less than zero. When $\partial U_{con}/\partial \lambda_i = 0$, we obtain the optimal pricing strategy. □

Algorithm 1 ULAR Mechanism

Input: $\{z_1, z_2 \cdots, z_I\}$, $\boldsymbol{g}$, and J.
Output: $\{C_1 \cdots C_J\}$, $\{z_1, z_2, \cdots z_J\}$, and $X_{i,j}$.
1: **Initialize** set cluster number $\{C_1 \cdots C_J\}$, the maximum iterations N, and maximum UAV load D_j.
2: Select J random samples from $\{z_1, z_2 \cdots, z_I\}$ as the initial centroid $(\mu_1 \cdots \mu_J)$.
3: **for** $i = 1$ to N **do**
4:   **for** $j = 1$ to J **do**
5:     Calculate the initial distance d_i^*.
6:     Allocate z_i to C_j with minimum d_i^j.
7:     Update μ according to μ_1.
8:   **end for**
9:   **if** $\mu_{j(n)} = \mu_{j(n-1)}$ **then**
10:     End loop.
11:   **end if**
12: **end for**
13: **while** $\sum_{i \in C_j} g_i > D_j$ **do**
14:   Removal z_i with $max(d_i^j)$ from C_j.
15:   Add z_i to $C_\iota \neq C_j$ with minimum d_i^j.
16: **end while**

Algorithm 2 Chess-like PSOPSSL Optimization Algorithm

Input: $X_{i,j}$, $\{C_1, C_2, \ldots, C_J\}$, $\{z_1, z_2, \ldots, z_J\}$, I, J, and parameters for PSOPSSL.
Output: The optimal pricing strategy $\boldsymbol{\lambda^*}$, the optimal offloading strategy $\boldsymbol{g^*}$, U_{con}^*, and U_i^*.
1: **repeat**
2:   **for** $i = 1$ to I **do**
3:     Update pricing strategy $\lambda_i(j)$ of the the UAV controller by the PSOPSSL algorithm.
4:     Update offloading strategy $g_i(j)$ of the UE i by using (12).
5:   **end for**
6:   Generate the optimal pricing set λ^*=$\{\lambda_1^*, \lambda_2^*, \ldots, \lambda_I^*\}$ and the optimal offloading set $\boldsymbol{g}^*$=$\{g_1^*, g_2^*, \ldots, g_I^*\}$.
7:   **while** $\sum_{i \in C_j} g_i^* > D_j$ **do**
8:     Remove z_i with $max(d_i^j)$ from C_j.
9:     Add z_i to $C_\iota \neq C_j$ with minimum d_i^j.
10:   **end while**
11:   Calculate the current utility of the UAV Controller.
12:   Calculate the current utility of the UE.
13: **until** a Nash equilibrium is obtained.

IV. ALGORITHM DESIGN

In this section, we design a low computing cost mechanism ULAR to find the best location of UAV-MEC, and adjust the unloading destination of UEs according to the maximum data capacity of UAV-MEC. Then, in order to achieve the Nash equilibrium efficiently, we propose the CPPO algorithm.

A. UAV Localization and Availability Response Mechanism

In order to deploy UAVs and allocate computing resources of UAVs more efficiently, we design the ULAR mechanism, as shown in algorithm 1. We adopt the K-means clustering based to select the optimal deployment positions of UAVs. From steps 1 to 12, the UEs are clustered according to the distance and the updated clusters are obtained. Further, when the amount of offloading data exceeds the maximum load of UAV j, the farthest UE in the cluster assigned to the next nearest cluster from steps 13 to 16. In general, with the Algorithm 1, more UEs can be served in the limited resources of UAVs.

B. Chess-like PSOPSSL Optimization Algorithm

We propose the CPPO to obtain the Nash equilibrium in Algorithm 2. The particle swarm optimization probability based strategy selection learning optimization (PSOPSSL) method is used to optimize the pricing strategy and offloading strategy to find the optimal solution (lines 2-5). We use the ULAR mechanism to simulate a scenario similar to the two-player game in chess, with the obtained solution to avoid the situation that the UAV is not available after optimization (lines 7-10). The utility of both sides is calculated to reach Nash equilibrium (lines 11-12). The iteration is carried out until a Nash equilibrium is reached.

Inspired by learning automata theory [15], we propose the PSOPSSL method and the position update formula is

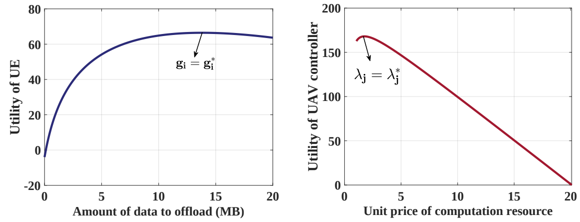
$$\begin{aligned} v_{ij}(t+1) = & w v_{ij}(t) + c_1^\tau r_1(t)\left[p_{ij}(t) - x_{ij}(t)\right] \\ & c_2^\tau r_2(t)\left[p_{ij}(t) - x_{ij}(t)\right], \end{aligned} \tag{17}$$

where c_1^τ is updated as $c_1^\tau = c_1^0 + 0.1 * s_1^\tau$, and s_1^τ is the round whose local extremums are invariant. $c_2^\tau = c_2^0 + 0.1 * s_2^\tau$ and s_2^τ are the global extremum invariant round. w represent the weight that keeps initial velocity, which can be expressed as $w = w - 0.1 * s_1^\tau$. τ is the number of iterations of PSOPSSL algorithm. The PSOPSSL adjusts the size of its parameters based on the number of iterations and the frequency of selecting the same extreme value position. It effectively balances both global and local search capabilities of particles.

V. SIMULATION RESULT

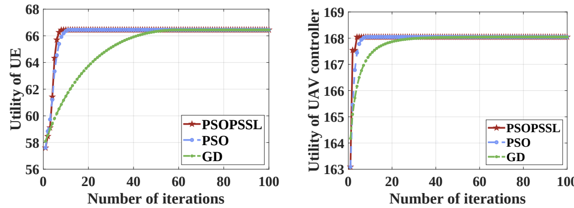
In this section, we carry out the simulations using MATLAB. The energy consumption data per unit task of UEs is set to $[0.2, 0.5]\mathrm{J/MB}$. δ_i is set as 40. For the UAV-MEC, the computing power is set as between [1, 5]$\times 10^9$cycles/s. The CPU power is a random value between [0.1, 0.5]W. We consider a data unit requires a CPU revolution of 1,900 cycles/byte, i.e., $\alpha = 1,900$ cycles/byte. The amount of data for each task is distributed in the range [10, 50]MB.

Fig. 3 depicts the process of Stackelberg equilibrium. Fig. 3(a) is the utility of UE i with a fixed UAV controller price $\lambda_i = \lambda_i^*$. The utility of UE i reaches the maximum when the unloading policy is $g_i = g_i^*$. Fig. 3(b) illustrates the income generated when the UAV controller adopts the fixed unloading strategy $g^* = g_i^*$. The optimal price $\lambda_i = \lambda_i^*$ corresponds to the point of the maximum profit. This analysis demonstrates that both UEs and UAV controller collaborate to achieve a Nash equilibrium.



(a) Optimal offloading of the UE. (b) Optimal price of the UAV.

Fig. 3. The Stackelberg equilibrium.



(a) Utility of UE. (b) Utility of UAV controller.

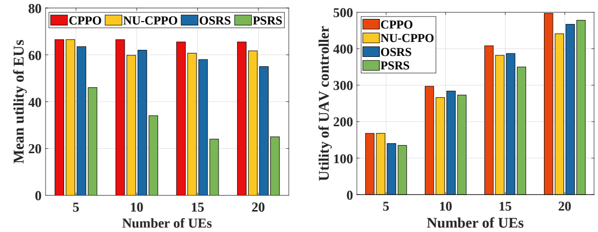
Fig. 4. Iterations of PSOPSSL algorithm.

Fig. 4 illustrates the iterative process of the proposed PSOPSSL. Fig. 4(a) shows the iterative convergence process for the UE. As iterations increases, the utility progressively improves, reaching the stable convergence after 10 iterations. Similarly, as shown in Fig. 4(b), the convergence is achieved after approximately 10 iterations as the number of iterations increases. It is noted that PSOPSSL converges faster than particle swarm optimization (PSO) and gradient descent (GD).

To assess the performance of the proposed CPPO method, we compare it with three alternative scenarios: 1) the CCPO algorithm without the ULAR mechanism (NU-CPPO), 2) uploading strategies determined via game theory (OSRS), and 3) pricing strategies determined via game theory (PSRS). In detail, Fig. 5(a) and 5(b) compare the mean utility of UEs for all users and the utilize of the UAV controller under four different offloading strategies, respectively. The CPPO scheme significantly improves the utility of UEs and the UAV controller compared to the OSRS and PSRS. Compared to the NU-CPPO strategy, the results of the comparison highlight the necessity of implementing the ULAR mechanism.

VI. Conclusions

This paper presents a novel framework in which a ground-based UAV controller coordinates the deployment of UAV-MECs within 3D corridors and efficiently allocates computational resources to support damaged UEs in disaster areas. Considering that there is a competitive relationship between the UAV controller and UEs, it is modeled as a Stackelberg game problem. We design a ULAR mechanism to optimize UAV placement and resource allocation. To deal with the formulated Stackelberg game problem, we propose a CPPO algorithm. Extensive simulations are conducted to assess the performance of the proposed methods, with results showing that the proposed algorithms outperform other baseline approaches.



(a) Mean utility of UEs. (b) Utility of UAV controller.

Fig. 5. Comparison of different algorithms.

References

[1] J. You, Z. Jia, C. Dong, L. He, Y. Cao, and Q. Wu, "Computation Offloading for Uncertain Marine Tasks by Cooperation of UAVs and Vessels," in *ICC - IEEE International Conference on Communications*, Rome, Italy, Oct. 2023.

[2] Z. Ning, Y. Yang, X. Wang, L. Guo, X. Gao, S. Guo, and G. Wang, "Dynamic Computation Offloading and Server Deployment for UAV-Enabled Multi-Access Edge Computing," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2628–2644, Nov. 2023.

[3] Z. Jia, Y. Cao, L. He, Q. Wu, Q. Zhu, D. Niyato, and Z. Han, "Service Function Chain Dynamic Scheduling in Space-Air-Ground Integrated Networks," *IEEE Trans. Veh. Technol.*, pp. 1–15, Feb. 2025.

[4] Y. He, K. Xiang, X. Cao, and M. Guizani, "Task Scheduling and Trajectory Optimization Based on Fairness and Communication Security for Multi-UAV-MEC System," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 30 510–30 523, Jun. 2024.

[5] S. Karimi-Bidhendi, G. Geraci, and H. Jafarkhani, "Optimizing Cellular Networks for UAV Corridors via Quantization Theory," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 14 924–14 939, Jul. 2024.

[6] N. Ruseno, C.-Y. Lin, and S.-C. Chang, "UAS Traffic Management Communications: The Legacy of ADS-B, New Establishment of Remote ID, or Leverage of ADS-B-Like Systems?" *Drones*, vol. 6, no. 3, p. 57, Feb. 2022.

[7] Z. Jia, J. You, C. Dong, Q. Wu, F. Zhou, D. Niyato, and Z. Han, "Co-operative Cognitive Dynamic System in UAV Swarms: Reconfigurable Mechanism and Framework," *IEEE Veh. Technol. Mag.*, vol. 19, no. 3, pp. 90–101, Jul. 2024.

[8] T. Fang, F. Yuan, L. Ao, and J. Chen, "Joint Task Offloading, D2D Pairing, and Resource Allocation in Device-Enhanced MEC: A Potential Game Approach," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3226–3237, Jul. 2021.

[9] L. Geng, H. Zhao, and C. Zou, "Stackelberg Game-Based Computation Offloading and Pricing in UAV Assisted Vehicular Networks," *IEEE Trans. Reliab.*, pp. 1–15, May 2024.

[10] Y. Zhang, X. Hou, H. Du, L. Zhang, J. Du, and W. Men, "Joint Trajectory and Resource Optimization for UAV and D2D-Enabled Heterogeneous Edge Computing Networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13 816–13 827, May 2024.

[11] N. Cherif, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, "3D Aerial Highway: The Key Enabler of the Retail Industry Transformation," *IEEE Commun. Mag.*, vol. 59, no. 9, pp. 65–71, Oct. 2021.

[12] D. Chao, Y. Zhang, J. Ziye, L. Yiyang, L. Zhang, and W. Qihui, "Three-dimension Collision-free Trajectory Planning of UAVs Based on ADS-B Information in Low-altitude Urban Airspace," *Chin. J. Aeronaut.*, vol. 38, no. 2, p. 103170, Aug. 2024.

[13] M. Wang, L. Zhang, P. Gao, X. Yang, K. Wang, and K. Yang, "Stackelberg-Game-Based Intelligent Offloading Incentive Mechanism for a Multi-UAV-Assisted Mobile-Edge Computing System," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15 679–15 689, Apr. 2023.

[14] K. Zhang, X. Gui, D. Ren, and D. Li, "Energy–Latency Tradeoff for Computation Offloading in UAV-Assisted Multiaccess Edge Computing System," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6709–6719, Jun. 2021.

[15] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications*. Elsevier, 2014.