A Scheme for Flexible-Hybrid Subtask Offloading in a Two-Tier UAV- Assisted MEC Network

Qiuji Luan
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing 100876, China
luanqiuji@bupt.edu.cn

Hongyan Cui
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing 100876, China
cuihy@bupt.edu.cn

Lifeng Zhang
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing 100876, China
zhanglf@bupt.edu.cn

ROY E. WELSCH
Sloan School of Business,
Massachusetts Institute of Technology
Cambridge, MA 02142, USA
rwelsch@mit.edu

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838442

Abstract—**As a technique to buffer the conflict between computation-intensive tasks and capability-limited devices, unmanned aerial vehicle-enabled mobile edge computing (UAV-MEC) has been witnessed as a promising approach. In this paper, we consider an optimization of dependent and independent subtasks hybrid offloading to maximize the utility, which is decided by the weighted sum of makespan gain and energy consumption gain. Firstly, we design a two-tier UAV-assisted MEC offloading architecture to obtain efficient and cooperative communication between the ground command centers and the UAVs. Secondly, we develop a flexible-hybrid subtask offloading scheme(F-HSO) to obtain the optimal solution of the problem in a parallel and sequential manner. Numerical results show that our algorithm can significantly outperform the other representative benchmarks in utility.**

Keywords—UAV-MEC, aerial network, flexible-hybrid offloading, dependent subtask.

I. INTRODUCTION

In aerial communications, due to flexible mobilities, fast deployment, low cost, and strong line-of-sight (LOS) link, unmanned aerial vehicles (UAVs) are regarded as a key solution to improve communication quality, have received extensive attention from academia and industry [1]. UAVs can quickly provide ubiquitous connections, can serve as relay nodes, and can also disseminate information [2]. Mobile edge computing (MEC) can provide computing and communication capabilities at the edge of the network, edge nodes and UAVs have recently emerged as edge computing nodes [3]. The UAV- assisted MEC network can relieve the pressure on ground communication and computing, especially when the ground infrastructures are destroyed, the computing load increases suddenly.

Wireless resource allocation are necessary means to improve transmission performance. To reduce packet collision and improve unstable channel conditions, a joint problem about the access point selection and the UAV path planning was formulated to maximize the UAV utility based on a packet loss rate [4]. The adaptive deployment of UAVs was studied to cater to instantaneous wireless traffic in a territory [5]. A resource allocation problem was investigated to limit interference over the millimeter-wave frequency band, aiming to maximize the sum rate while ensuring a minimum rate guarantee for each user [6]. These efforts have achieved performance improvements through the formulation of effective schemes. However, these effort works are focused on improving the resource utilization of UAVs, and did not fully explore the functions of multiple computing and communication devices in the UAV-assisted communication network, that is, they did not maximize the contribution of each device to the network. Therefore, we designed a two-tier UAV-assisted MEC architecture, which makes full use of the data of the load equipment carried by the UAV, the distributed computing capability of the UAV, and the data processing capability of the ground command center. Devices process data and transmit control information cooperatively, and tasks are executed efficiently.

Complex computing applications require massive computing capabilities, such as mapping, surveillance, photography, etc. However, due to size and cost constraints, the energy and computing capabilities of computing devices do not support resource-intensive and time-sensitive tasks. This is an important challenge for a UAV- assisted MEC networks. A cooperative UAVs computation task offloading scheme based on vehicular fog computing system was developed [7]. A stable matching algorithm is proposed to resolve the transmission competitions. Nevertheless, this work did not consider the dependency constraint among subtasks in the optimization problem. That is, the subtask depends on the output of one or more other subtasks. Since it has a significant impact on the task offloading and executing performance, cannot be ignored, a dependency-aware offloading scheme in MEC with edge-cloud cooperation was proposed under task dependency constraints, aiming at minimizing task finishing time [8]. Considering task dependency between the devices, an optimal task offloading and resource allocation policy was studied to minimize the weighted sum of the devices' energy consumption and task execution time [9]. A task offloading method based on meta reinforcement learning was proposed, offloaded tasks could run in parallel on all cores[10]. However, the waiting delay will increase when the subtasks are offloaded sequentially, and the limited resources will reduce the gain caused by offloading when the subtasks are offloaded in parallel. Hence, in order to improve the offloading efficiency, it is necessary to consider both sequential and parallel offloading simultaneously. Moreover, the priority of multiple subtask offloading needs to be considered. Different from these works, we evaluate multi-dimensional flexible constraints, such as density, resource status, etc., and formulate a fitness function to obtain parallel and sequential offloading priority. Furthermore, considering dependencies between subtasks, we develop a hybrid subtask offloading scheme to maximize the utility.

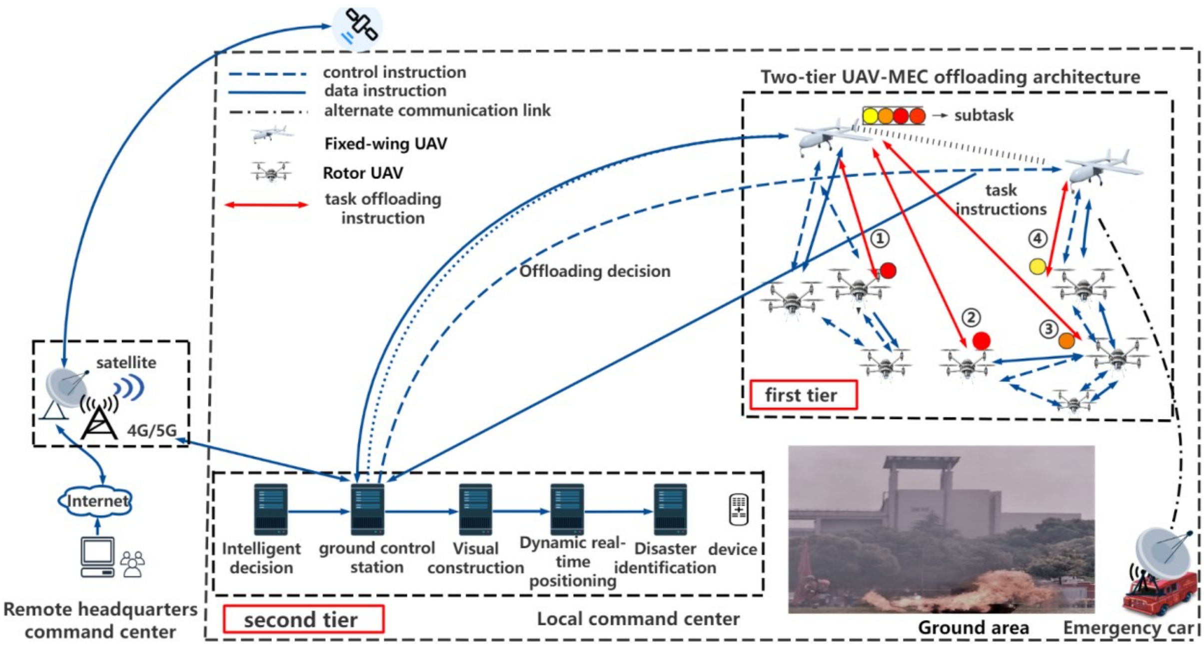

Fig.1 An illustration of a two-tier UAV-assisted MEC network

II. SYSTEM MODEL

A. A Two-Tier UAV-MEC System Architecture

We consider a two-tier UAV-assisted MEC offloading architecture, as shown in Fig.1. The architecture includes two types of UAVs: fixed-wing UAVs and rotor UAVs, both are regarded as flying base stations (FBSs), each FBS is equipped with a MEC sever to extend ground devices with communication and computing capabilities. Fixed-wing UAVs have decision-making capabilities, and coordinate with the command centers to offload subtasks. According to the control instructions from fixed-wing UAVs, rotor UAVs execute subtasks in a distributed manner.

In the first tier, the fixed-wing UAVs are controllers, which obtain the global information of all FBSs within the communication range through cooperation with the ground command centers. Rotor UAVs are actuators that can carry a variety of loads, such as equipment used in positioning, obstacle avoidance, information collection, life detection, etc., these loads generate massive amounts of data. In the second tier, in order to support the execution of subtasks such as virtualization construction, dynamic real-time positioning, and disaster identification, FBS transmits pre-processed data to the second-tier ground command centers. The processed data is transmitted back to the first tier, and then FBS's computing and communication capabilities are used to achieve task offloading. However, the data is heterogeneous, multi-sourced, UAVs have limited computing ability and cannot process massive amounts of data in real time to execute tasks. Therefore, the proposed two-tier UAV-assisted MEC offloading architecture can improve the computing and communication efficiency of aerial communications.

For the task that contains four subtasks in Fig.1, subtask 1 and subtask 2 are dependent, and the others are independent. Hence, in the first step, subtask 1, subtask 3, and subtask 4 are offloaded in parallel. In the second step, subtask 2 gets the computing result of subtask 1 and starts execution. Therefore, the hybrid task offloading strategy that considers dependency improves computing efficiency and makes full use of computing resources.

We assume a quasi-static UAV-assisted MEC network, which containing M UAVs and N ground devices, which are denoted as $\mathcal{M}=\{1,2,\ldots,M\}$, $\mathcal{N}=\{1,2,\ldots,N\}$, respectively. $\mathcal{K}=\mathcal{M}\cup\mathcal{N}$. We let $\mathcal{K}$ be the set of communication nodes, including UAVs and ground devices. For each node $m\in\mathcal{K}$, the transmission power is p_m , $p_m \le p^{\max}$, where $p^{\max}$ is maximum power. For each pair of nodes (m,n) , the communication indicator is $e_{m,n}\in\{0,1\}$, which denotes information can be transmitted between node m and n . The transmission rate and allocated bandwidth between node m and n are denoted as $r_{m,n}$ and $B_{m,n}$, respectively. $B_{m,n}=\sum_{l=1}^{|L|}\alpha b_{m,n,l}$, where $b_{m,n,l}\in\{0,1\}$, $b_{m,n,l}=1$ means sub-channel l is allocated to node i for communication, $l\in\{1,2,\cdots,|L|\}$, α is the bandwidth of each sub-channel. We assumed that the optimal bandwidth between nodes has been allocated in this paper.

B. Task Model

We assume that the task $\mathcal{T}_i$ consists of multiple dependent and independent subtasks, are denoted as $\mathcal{T}_i=\{\{t_{i,1},t_{i,2},\cdots,t_{i,j-1}\},t_{i,j},\{t_{i,j+1}\cdots,t_{i,T}\}\}$, where $t_{i,j}$ is the j th subtask of the i th task. T is the number of subtasks. In subset $\{t_{i,1},t_{i,2},\cdots,t_{i,j-1}\}$ and $\{t_{i,j+1}\cdots,t_{i,T}\}$, subtasks are dependent. The subtask $t_{i,j}$ is independent. The task $\mathcal{T}_i$ is represented as a directed call graph $G^T(\mathcal{T}_i,\mathcal{V}_i)$, where the set of edges $\mathcal{V}_i$ represents the dependency between the subtasks. If there is an edge $v_{i,j,k}\in\mathcal{V}_i$, that is, $v_{i,j,k}=1$, means subtask $t_{i,j}$ must be completed and transfer its computing result to subtask $t_{i,k}$ before subtask $t_{i,k}$ starts execution. Hence, we define the subtask $t_{i,j}$ is a "parent" subtask of the subtask $t_{i,k}$, the subtask $t_{i,k}$ is a "child" of the subtask $t_{i,j}$. In Fig.1, subtask 1 is the "parent" subtask of subtask 2.

III. PROBLEM FORMULATION

A. Data Transmission Rate

1) Data transmission rate of UAV-to-UAV(U2U)

Since the high altitude of the UAV, we assume the U2U channel is the LOS, the path loss $\eta^U_{m,m'}$ between mth UAV and m'th UAV is $\eta^U_{m,m'}$. The channel gain $g^U_{m,m'}$ for U2U communication contains path loss and small-scale Rayleigh fading is calculated by $g^U_{m,m'} = \beta^U_{m,m'} 10^{-\eta^U_{m,m'}}$, where $\beta^U_{m,m'}$ is the small-scale fading coefficient. The data transmission rate from the mth UAV to the m'th UAV is given by

$$r^U_{m,m'} = B_{m,m'} \log(1 + \frac{p_m g^U_{m,m'}}{\sigma^2 + \sum_{m''\in\mathcal{M}\setminus\{m,m'\}} p_{m''} g^U_{m'',m'}}) \quad (1)$$

where σ^2 is the power of additive white Gaussian noise, $B_{m,m'}$ represents the allocated bandwidth. The co-channel interference from m''th UAV if it uses the same channel is $\sum_{m''\in\mathcal{M}\setminus\{m,m'\}} p_{m''} g^U_{m'',m'}$.

2) Data transmission rate of UAV-to-user(U2E)

The channel of U2E communication includes large-scale fading, which can result in a Non- Line of Sight (NLOS) channel. Therefore, U2E channel gain between mth UAV and nth user is modeled as an opportunistic LOS, which is given as

$$g^{U,E}_{m,n} = \eta^0_{m,n} + \xi_{m,n}\eta^{LoS}_{m,n} + (1-\xi_{m,n})\eta^{NLoS}_{m,n} \quad (2)$$

where $\eta^0_{m,n}$ is the distance average path loss. $\xi_{m,n}$ is the probability of LOS, can be estimated by [11]. $\eta^{LoS}_{m,n}$,$\eta^{NLoS}_{m,n}$ are the average path loss for LOS and NLOS. The data rate is expressed as

$$r^{U,E}_{m,n} = B^{U,E}_{m,n} \log(1 + \frac{p_n g^{U,E}_{m,n}}{\sigma^2 + I^U_{m,m'} + I^{U,E}_{m,n'}}) \quad (3)$$

where the co-channel interference from m'th UAV and n'th user that reuses sub-channel with n th user are $I^U_{m,m'} = \sum_{m'\in\mathcal{M}\setminus\{m\}} p_{m'} g^U_{m,m'}$, $I^{U,E}_{m,n'} = \sum_{n'\in\mathcal{N}\setminus\{n\}} p_{n'} g^{U,E}_{m,n'}$.

B. Subtask Offloading Model

1) Local computing

We assume that the node m has a subtask $t_{i,j}$ to be executed, $t_{i,j} \in \mathcal{T}_i$. The computing resources of node m is denoted as ϕ^l_m, the local computation capability is ρ^l_m. Thus, the local computing time of subtask $t_{i,j}$ is given as $\tau^l_{i,j,m} = c_{i,j} / \rho^l_m$. The size of input data, the output computed result, and the number of CPU cycles that are required to execute the subtask $t_{i,j}$ is denoted by $u_{i,j}$, $d_{i,j}$, $c_{i,j}$. The energy consumption is expressed as $W^l_{i,j} = \theta c_{i,j} (\rho^l_m)^2$, θ is set to be 10^{-11} [8].

2) Edge computing

The subtask $t_{i,j}$ is offloaded to the edge node n. For edge node n, the computation resources and computation capability are denoted by ϕ^e_n, ρ^e_n, respectively. Let $a_{i,j,n} \in \{0,1\}$ represents the offloading decision vector of subtask $t_{i,j}$. Especially, $a_{i,j,n} = 1$ denotes that the subtask $t_{i,j}$ is offloaded to edge node n, $a_{i,j,n} = 0$ means $t_{i,j}$ is executed locally. The edge execution time of the subtask $t_{i,j}$ on the edge node n is calculated as $\tau^e_{i,j,n} = c_{i,j} / \rho^e_n$. The energy consumption of the edge node n is given by $W^e_{i,j} = (\mu c_{i,j} (\rho^e_n)^v + \vartheta)\tau^e_{i,j,n}$ [8].

3) Dependency Model

The subtask $t_{i,j}$ can be offloaded to only one edge node.

$$\sum_{m=1}^{M+N} a_{i,j,m} * \sum_{n=1}^{M+N} a_{i,j,n} = 0, \forall t_{i,j} \in \mathcal{T}_i, \{m,n\} \in \mathcal{K}, m \neq n \quad (4)$$

The subtask $t_{i,j}$ can only be executed once locally or at the edge node m as represented by (5).

$$\sum_{m=1}^{M+N} a_{i,j,m} \le 1, \quad t_{i,j} \in \mathcal{T}_i, m \in \mathcal{K} \quad (5)$$

We assume that node m is executing subtask $t_{i,j'}$ when the subtask $t_{i,j}$ is ready to be executed on m. For the subtask $t_{i,j}$, the start time $Ts_{i,j,m}$ is larger than the finish time of $t_{i,j'}$, which is represented by (6).

$$Ts_{i,j,m} \ge Tf_{i,j',m}, \quad \forall \{t_{i,j}, t_{i,j'}\} \in \mathcal{T}_i, m \in \mathcal{K} \quad (6)$$

where $Tf_{i,j',m}$ is the makespan of subtask $t_{i,j'}$. If the subtask dependency vector $v_{i,j,j'}$=1, the subtask $t_{i,j'}$ can start after receiving computing results from the preceding subtask $t_{i,j}$, which is represented by (7).

$$Ts_{i,j',n} \ge Tf_{i,j,m}, \quad \forall a_{i,j',n} = 1, a_{i,j,m} = 1, \{t_{i,j}, t_{i,j'}\} \in \mathcal{T}_i, \{m,n\} \in \mathcal{K} \quad (7)$$

The resulting transmission time $Tu_{m,n}$ from node m to node n is given as $Tu_{m,n} = d_{m,n} / r_{m,n}$.

The makespan $Tf_{i,j,m}$ of the subtask $t_{i,j}$ at the edge node m is composed of uplink transmission time $Tr_{i,j,m}$, waiting time $Tw_{i,j}$, execution time $\tau_{i,j,m}$, and downlink result transmission time $Tu_{m,n}$.The subtask dependency model can be one of the following four cases. Fig. 2 is an example DAG diagram of a task.

Case 1: The subtask $t_{i,j}$ is independent(e.g. subtask 5 in Fig.2): when $a_{i,j,m} = 0$, $t_{i,j}$ is executed locally, the makespan is $\tau^l_{i,j,m}$. However, when subtask $t_{i,j}$ is offloaded to the edge node m. The makespan is the sum of the uplink transmission time $Tr_{i,j,m}$ and execution time $\tau^e_{i,j,m}$, Thus, the makespan $Tf_{i,j,m}$ of *case 1* is given by

$$Tf_{i,j,m} = (1 - a_{i,j,m}) * \tau^l_{i,j,m} + a_{i,j,m} * \left(Tr_{i,j,m} + \tau^e_{i,j,m}\right) \quad (8)$$

Case 2: The subtask $t_{i,j}$ has signal-layer parent subtasks (e.g. subtask 3 in Fig.2, its parent subtasks are subtask 1 and 2): We assume that the set of "parent subtasks" of subtask $t_{i,j}$ is expressed as $\mathcal{T}_{i,pre} = \left\{\{t_{i,1}\}, \{t_{i,2}\}, \cdots, \{t_{i,pre}\}, \cdots, \{t_{i,T_{i,pre}}\}\right\}$, $v_{i,pre,j} = 1$, $t_{i,pre} \in \mathcal{T}_{i,pre}$. The parent subtask $t_{i,pre}$ is offloaded to node n, and needs to transmit the result before the "child subtask" $t_{i,j}$ is executed. Thus, the makespan of subtask is calculated by

$$Tf_{i,j,m} = \max\left\{ a_{i,j,m} * Tr_{i,j,m}, \max_{t_{i,pre}\in\mathcal{T}_i\setminus\{t_{i,j}\}} \{v_{i,j,pre} * (Tf_{i,pre,n} + Tu_{n,m})\} \right\} + \tau_{i,j,m} \tag{9}$$

where $\max_{t_{i,pre}\in\mathcal{T}_i\setminus\{t_{i,j}\}} \{v_{i,j,pre} * (Tf_{i,pre,n} + Tu_{n,m})\}$ represents the maximum time for the parent subtasks to be transmitted from node n to m.

Case 3: The subtask $t_{i,j}$ has multi-layer parent subtasks (e.g. subtask 4 in Fig.2, its parent subtasks are subtask 1, 2 and 3, and the parent subtasks of subtask 3 are subtask 1 and 2): We assume the set of *multi-layer parent subtasks* is $\mathcal{T}_{i,p-pre} = \left\{\{t_{i,1}\},\cdots,\{t_{i,p-pre}\},\cdots,\{t_{i,T_{p-pre}}\}\right\}$.The result can not only be transmitted directly but also can be transferred through the parent subtask of the upper layer $\mathcal{T}_{i,pre}$, as shown in the second term of (10). Thus, the makespan of subtask can be calculated by

$$Tf_{i,j,m} = a_{i,j,m} * Tr_{i,j,m} + \min_{t_{i,pre}\in\mathcal{T}_i\setminus\{t_{i,j}\}} \Bigg\{ v_{i,j,pre} * \max\{Tf_{i,pre,n} + Tu_{n,m}\}, \; v_{i,pre,\; p-pre} * \left(Tf_{i,pre,n} + \frac{d_{i,pre} + \sum_{t_{i,p-pre}\in\mathcal{T}_i\setminus\{t_{i,j},t_{i,pre}\}} d_{i,p-pre}}{r_{j,pre}} \right) \Bigg\} + \tau_{i,j} \tag{10}$$

where $t_{i,p-pre} \in \mathcal{T}_{i,p-pre}$ is the parent subtask of $t_{i,pre}$.

Case 4: *The edge node m is occupied when the subtask $t_{i,j}$ arrive*s*:* since an edge node can execute at most one subtask at the same time, the subtask $t_{i,j}$ can be executed only after the edge node m is released. Hence, the makespan of subtask $t_{i,j}$ can be expressed as

$$Tf_{i,j,m} = a_{i,j,m} * Tr_{i,j,m} + \max_{t_{i,j'}\in\mathcal{T}_i}\{a_{i,j',m} * (Tf_{i,j',m} - Tr_{i,j,m}), 0\} + \tau_{i,j} \tag{11}$$

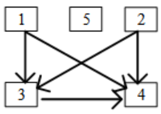

Fig.2 An example DAG diagram of a task

4) Energy Consumption Model

The energy consumption of computing subtask $t_{i,j}$ is the sum of communication energy consumption and executing energy consumption, is expressed by

$$W_{i,j} = (1-a_{i,j,m})W_{i,j}^{l} + a_{i,j,m}\left[W_{i,j}^{e} + \sum_{n=1}^{|Neig|} e_{m,n}\alpha_0 p_m \right] \tag{12}$$

where $(1-a_{i,j,m})W_{i,j}^{l} + a_{i,j,m}W_{i,j}^{e}$ represents the energy consumption of executing subtask $t_{i,j}$, $\sum_{n=1}^{|Neig|} e_{m,n}\alpha_0 p_m$ is the energy consumption of exchanging information with $|Neig|$ neighboring nodes, $\alpha_0 > 0$ denotes the unit energy consumption of the power.

C. Objective Function

The goal of the problem is to maximize the utility $U_{i,j}$ of executing subtask $t_{i,j}$ of $\mathcal{T}_i$,which is defined as the weighted sum of makespan gain and energy consumption gain, can be expressed as

$$U_{i,j} = \gamma_1 \frac{Tf_i^{\max} - Tf_{i,j,m}}{Tf_i^{\max}} + \gamma_2 \frac{W_i^{\max} - W_{i,j}}{W_i^{\max}} \tag{13}$$

where γ_1 , γ_2 are positive weights. $Tf_i^{\max}$, $W_i^{\max}$ are maximum tolerable makespan and energy consumption of task $\mathcal{T}_i$, respectively. Therefore, the objective function is

$$\begin{aligned} &\max_{\boldsymbol{a},\boldsymbol{e},p_m} \frac{1}{T}\sum_{j=1}^{T} U_{i,j} \\ &s.t. \quad C1:(4), C2:(5), C3:(6), C4:(7) \\ &\quad C5: e_{m,n} \in \{0,1\}, \quad \forall m,n \in \mathcal{K} \\ &\quad C6: a_{i,j,m} \in \{0,1\}, \forall t_{i,j} \in \mathcal{T}_i, m \in \mathcal{K} \\ &\quad C7: 0 \le p_m \le p_m^{\max}, \forall m \in \mathcal{K} \\ &\quad C8: Tf_{i,j,m} \le Tf_i^{\max}, \forall t_{i,j} \in \mathcal{T}_i \\ &\quad C9: W_{i,j} \le W_i^{\max}, \forall t_{i,j} \in \mathcal{T}_i \end{aligned} \tag{14}$$

Constraints $C1$ and $C2$ guarantee that each subtask is offloaded to only one edge node, and is executed at most once, respectively; Constraint $C3$ ensures that an edge node can perform at most one subtask at the same time; Constraint $C4$ is the dependency constraint; The binary constraints are presented in $C5$ and $C6$; $C7$ is the power constraint; $C8$, $C9$ limit the maximum tolerable makespan and energy consumption.

IV. ALGORITHMS FOR SUBTASK OFFLOADING

The problem (14) is a mixed-integer and non-convex optimization problem, and cannot be obtained the optimal solution in polynomial time. Therefore, we decouple the problem into a topology reconstruction subproblem about $\{p_m, \boldsymbol{e}\}$, which is solved by [12], and a subtask offloading subproblem about $\{\boldsymbol{a}\}$, which is solved by our proposed F-HSO algorithm. Based on the optimal solution $\{p_m^*, \boldsymbol{e}^*\}$, F-HSO obtains the offloading priority through the designed fitness firstly, and then offloads subtasks in a hybrid way.

A. Offloading Priority with Multiple Flexible Constraints

When the task offloading decision is made, we not only consider fixed constraints, such as UAV's density, and connection fairness, but also heterogeneous parameters such as task requirements, resource status, and the UAV's computing status. Therefore, we formulate a subtask offloading fitness function based on multiple flexible constraints, and obtain the task offloading priority.

The connection fairness between the m th node and n th neighboring node $F_m(p_m)$ is given by

$$F_m(p_m) = \left(\sum_{n=1}^{|Neig|} \hat{e}_{m,n} r_{m,n}(p_n)\right)^2 \Big/ |Neig| \sum_{n=1}^{|Neig|} \left(r_{m,n}(p_m)\right)^2 \tag{15}$$

where $\hat{e}_{m,n}$ is the slack variable of edge $e_{m,n}$, $0 \le \hat{e}_{m,n} \le 1$. $|Neig|$ is the number of neighboring nodes. We assume that the bandwidth $B_{m,n}$ is constant, p_m is a linear function of $r_{m,n}$, so $F_m(p_m)$ can be rewritten as

$$F_m(p_m) = \left(\sum_{n=1}^{|Neig|} \hat{e}_{m,n} p_n\right)^2 \Big/ |Neig| \sum_{n=1}^{|Neig|} (p_m)^2 \tag{16}$$

The link quality $q_{m,n}$ is determined by the data transmission rate, which is given by

$$q_{m,n} = 1 \big/ {r_{m,n}}^2 \tag{17}$$

The granularity $g_{i,j}$ of a subtask $t_{i,j}$ is defined as the ratio of computational workload to communication workload [13], and is expressed as

$$g_{i,j} = \left(\alpha_1 d_{i,j} + \alpha_2 u_{i,j}\right) \big/ r_{m,n} \tag{18}$$

where α_1, α_2 are weights, satisfying $\alpha_1, \alpha_2 \in [0,1]$, $\alpha_1 + \alpha_2 = 1$. Therefore, the fitness function fit_n of the candidate UAV n is given by

$$fit_n = \beta_0 * F_m + \beta_1 * q_{m,n} + \beta_2 * g_{i,j} \tag{19}$$

where β_0, β_1, β_2 are positive weights, satisfying $\beta_0, \beta_1, \beta_2 \in [0,1]$, $\beta_0 + \beta_1 + \beta_2 = 1$.

Considering the dependency of subtasks (which are given) and the fitness function, the offloading priority SR is formed, can be used as the basis of hybrid offloading.

B. Hybrid Subtask Offloading Scheme

We propose a solution to achieve hybrid subtask offloading and find the optimal $\{\boldsymbol{a}^*\}$ to maximize the utility. The details of the proposed solution are given in Algorithm 1. Step 1 gets the flexible priority SR, the subtask that inter-SR's subsets are independent, intra-SR's subsets are dependent. Dependent subtasks are offloaded sequentially (step 3-step 9), the time complexity is $O(|SR_m^{\max}| * |\mathcal{K}|) \ll O(T * |\mathcal{K}|)$; Independent subtasks are offloaded in parallel (step 10-step 13), the time complexity is $O(|\mathcal{K}|)$); Step 14- step 22 get the most profitable decision (time complexity is $O(T)$). Hence, the time cost of Algorithm 1 is $O(T * |\mathcal{K}|)$.

V. PERFORMANCE EVALUATION

In this section, we compare the performance of F-HSO with other benchmark solutions. The UAVs and UEs are randomly distributed in an area with a radius of 2Km. The limited height of the UAV is [20-300]m. The number of UAVs is 20% of the total number of nodes. The sub-carrier frequency b^c is set to 2GHz, and the channel bandwidth α is 5MHz. The maximum power of UAVs and UEs is 0.1W, the maximum power of infrastructure is 40W. Other channel parameters are set as in [14]. The power of noise is $\sigma^2 = 1$. We assume that a node (UAV or UE) generates a subtask at each time slot, the data size of a subtask $u_{i,j}$ is [2-8]Mbit [15]. The workload $d_{i,j}$ is gained by $1.2 u_{i,j}$. The computing ability of UAVs is 3.6×10^9Hz, the computing ability of UEs is 1.0×10^8Hz [8]. To highlight the advantage of our proposed methods, we compare performances of our approaches with some solutions in existing work described as follows:

One-climb Policy based Gibbs Sampling (OCP-GS) [9]: The subtasks of each user are offloaded sequentially, and the subtasks between users are offloaded in parallel. However, we consider the offloading decision of multiple subtasks of a user. Hence, in OCP-GS, subtasks are offloaded sequentially.

Local execution (LE): Subtasks are executed locally according to the release time.

Algorithm 1: F-HSO

Input: network topology $G^{n^*}(\mathcal{K}, \mathcal{E}^*)$; the number of subtasks T, the number of nodes $|\mathcal{K}|$.

Output: optimal $\{\boldsymbol{a}_{i,j,m}{}^*\}$, $\{U_{i,j}{}^*\}$

1. compute the offloading priority SR of task $\mathcal{T}_i$ as (19)
2. **for** $k = 1$ to $|SR|$, $k \in SR$ **do**
3.   **if** $|SR_k| > 1$ **do**
4.   **for** $j = 1$ to $|SR_k^{\max}|$, $t_{i,j} \in SR_k$ **do**
5.   **for** $m = 1$ to $|\mathcal{K}| - 1$ **do**
6.   utility $U_{i,j}$ as (13)← compute makespan $Tf_{i,j,m}$, energy consumption $W_{i,j}$ of subtask $t_{i,j}$ as (8)- (11), (12)
7.   **end for**
8.   **end for**
9.   **else do**
10.   **for** $m = 1$ to $|\mathcal{K}| - 1$ **do**
11.   utility $U_{i,j}$ as (13)← compute makespan $Tf_{i,j,m}$, energy consumption $W_{i,j}$ of subtask $t_{i,j}$ as (8)- (11), (12)
12.   **end for**
13.   **end if**
14.   **for** $j = 1$ to T **do**
15.   utility of local execution $U_{i,j}^l$ as (13)←compute the local makespan $\tau_{i,j,m}^l$, energy consumption $W_{i,j}^l$
16. $U_{i,j}' = \arg\max_{j \in T, m \in |\mathcal{K}|-1} \{U_{i,j}\}$
17.   **if** $U_{i,j}^l < U_{i,j}'$ **do**
18. $a_{i,j,m} = 1$, $U_{i,j}{}^* = U_{i,j}'$
19.   **else do**
20. $a_{i,j,m} = 0$, $U_{i,j}{}^* = U_{i,j}^l$
21.   **end if**
22.   **end for**
23. **end for**

Edge execution (RE): According to the release time of each subtask, the RE solution is obtained by greedily offloading the subtasks to the edge node with the maximum utility.

Fig.3 plots the average utility versus the number of nodes. There is an increase in performance difference between F-HSO and OCP-GS from 20.4% at 20 nodes to 33% at 100 nodes. It embodies the advantages of power control and topology reconstruction of F-HSO. OCP-GS ignores the priority, and when the number of nodes increases, the advantages of the F-HSO are more obvious due to the more complex environment. LE solution has a long waiting delay, this is the reason that the utility of F-HSO outperforms LE by about 92% on average. RE sacrifices power to offload all the subtasks to the edge node, and causes additional consumption of energy, so the maximum utility is 86% lower than F-HSO.

In Fig.4, the utility is evaluated under different data sizes when the number of nodes is 100. We can observe that the performance has shown linear growth with the increasing number of data sizes. The utility of LE is the least, which is

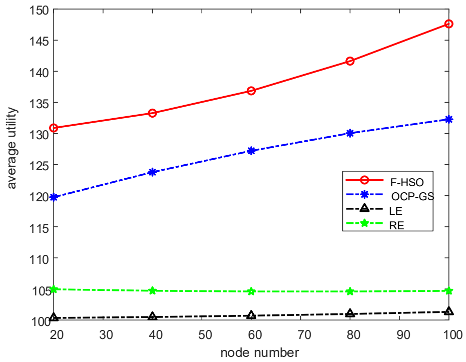

Fig.3 utility with the number of nodes

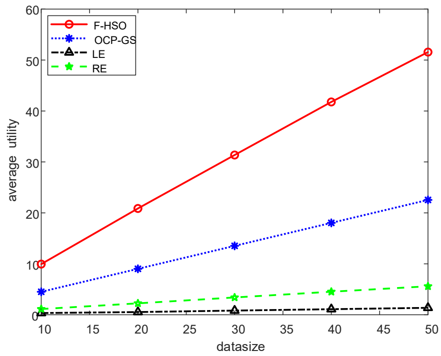

Fig.4 utility with different data sizes

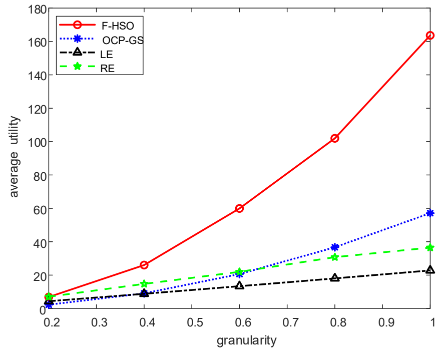

Fig.5 utility with different granularity

caused by limited local computing ability and local computing resources. The performance gap between F-HSO and RE increases from 16% to 78% as the data size increases from 10Kb to 50Kb. This can be explained as dependency constraints and resource competition extend the waiting time of RE. We can see that the average utility gap between OCP-GS and F-HSO increases from 9.9% at 10Kb to 16.5% at 50Kb. This can be explained as OCP-GS interact with all neighboring nodes, resulting in a waste of energy. This disadvantage is more obvious when the data size becomes larger,

Fig.5 demonstrate the changes of the average utility with the variation of granularity when the number of nodes is 100. Because the execution of computation-intensive takes more time than communication-intensive tasks, the average utility of subtasks increases as the granularity increases. Almost all subtasks are executed locally when the granularity is small. LE is only related to the computational workload, the maximum utility gap is 78% with F-HSO. For RE, energy consumption reduces the utility brought by offloading subtasks, which caused the difference between F-HSO and RE to be increases from 1.5% at granularity=0.2 to 71% at granularity=1. Complex information interaction reduces the utility of OCP-GS, the maximum utility is 59% lower than F-HSO, also shows the superiority of hybrid offloading.

VI. CONCLUSION

In this paper, we have studied the subtask offloading problem in UAV-assisted MEC network. First, in order to support task offloading efficiently, we have designed a two-tier UAV-assisted MEC architecture to make full use of the resource of the ground control center and UAVs. Second, we have proposed an F-HSO algorithm to maximize the average utility. By evaluating multi-dimensional flexible constraints, we have formulated the fitness function to obtain parallel and sequential hybrid offloading priority. Considering the power, the connection and the dependency constraints between nodes, we have investigated a hybrid subtask offloading scheme to obtain the optimal subtask offloading decision. Simulation results have validated the efficiency by varying different simulation parameters, including the number of nodes, the number of data sizes, granularity. Performance comparison shows that the proposed F-HSO has higher utilities than other benchmark solutions.

ACKNOWLEDGEMENT

This work was in part supported by the National Natural Science Foundation of China under Grant 62171049, the National Key Research and Development Program of China under Grant 2020YFB1807805.

REFERENCES

[1] X. Liu et al., "Transceiver Design and Multihop D2D for UAV IoT Coverage in Disasters," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1803-1815, Apr. 2019.

[2] Y. Zeng, R. Zhang and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36-42, May 2016.

[3] S. Wan, J. Lu, P. Fan and K. B. Letaief, "Toward Big Data Processing in IoT: Path Planning and Resource Management of UAV Base Stations in Mobile-Edge Computing System," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5995-6009, Jul. 2020.

[4] S. Zhu, L. Gui, N. Cheng, F. Sun and Q. Zhang, "Joint Design of Access Point Selection and Path Planning for UAV-Assisted Cellular Networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 220-233, Jan. 2020.

[5] Z. Wang, L. Duan and R. Zhang, "Adaptive Deployment for UAV-Aided Communication Networks," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 9, pp. 4531-4543, Sep. 2019.

[6] S. Kumar, S. Suman and S. De, "Dynamic Resource Allocation in UAV-Enabled mmWave Communication Networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9920-9933, Jun. 2021.

[7] W. Chen, Z. Su, Q. Xu, T. H. Luan and R. Li, "VFC-Based Cooperative UAV Computation Task Offloading for Post-disaster Rescue," in Proc. *IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Toronto, Canada, Jul. 2020, pp. 228-236.

[8] L. Chen, J. Wu, J. Zhang, H. -N. Dai, X. Long and M. Yao, "Dependency-Aware Computation Offloading for Mobile Edge Computing with Edge-Cloud Cooperation," *IEEE Trans. Cloud. Comput*., early access, doi: 10.1109/TCC.2020.3037306.

[9] J. Yan, S. Bi, Y. J. Zhang and M. Tao, "Optimal Task Offloading and Resource Allocation in Mobile-Edge Computing With Inter-User Task Dependency," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 1, pp. 235-250, Jan. 2020.

[10] H. Djigal, J. Feng, J. Lu and J. Ge, "IPPTS: An Efficient Algorithm for Scientific Workflow Scheduling in Heterogeneous Computing Systems,"*IEEE Trans. Parallel. Distrib. Syst.*, vol. 32, no. 5, pp. 1057-1071, May 2021.

[11] T. Do-Duy, L. D. Nguyen, T. Q. Duong, S. Khosravirad and H. Claussen, "Joint Optimisation of Real-time Deployment and Resource Allocation for UAV-Aided Disaster Emergency Communications," *IEEE J. Sel. Areas Commun.*, early access.

[12] Y. Du, J. Xia, J. Gong, X. Hu, “An Energy-Efficient and Fault-Tolerant Topology Control Game Algorithm for Wireless Sensor Network,” *Electronics*, vol. 8, no. 9, pp. 1009,2019.

[13] Y. Liu, S. Xie and Y. Zhang, "Cooperative Offloading and Resource Management for UAV-Enabled Mobile Edge Computing in Power IoT System," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12229-12239, Oct. 2020.

[14] S. Zhang, H. Zhang, B. Di and L. Song, "Resource Allocation and Trajectory Design for Cellular UAV-to-X Communication Networks in 5G," *in Proc. IEEE Global Communications Conference (IEEE GLOBECOM)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-6.

[15] Y. Sahni, J. Cao, L. Yang and Y. Ji, "Multi-Hop Multi-Task Partial Computation Offloading in Collaborative Edge Computing," *IEEE Trans. Parallel. Distrib. Syst.*, vol. 32, no. 5, pp. 1133-1145, May 2021.