

Satellite-Aerial Integrated Computing in Disasters: User Association and Offloading Decision

Long Zhang ¹, Hongliang Zhang ¹, Chao Guo ¹, Haitao Xu ¹, Lingyang Song ¹, and Zhu Han ¹

¹Affiliation not available

November 8, 2023

Abstract

In this paper, a satellite-aerial integrated computing (SAIC) architecture in disasters is proposed, where the computation tasks from two-tier users, i.e., ground/aerial user equipments, are either locally executed at the high-altitude platforms (HAPs), or offloaded to and computed by the Low Earth Orbit (LEO) satellite. With the SAIC architecture, we study the problem of joint two-tier user association and offloading decision aiming at the maximization of the sum rate. The problem is formulated as a 0-1 integer linear programming problem which is NP-complete. A weighted 3-uniform hypergraph model is obtained to solve this problem by capturing the 3D mapping relation for two-tier users, HAPs, and the LEO satellite. Then, a 3D hypergraph matching algorithm using the local search is developed to find a maximum-weight subset of vertex-disjoint hyperedges. Simulation results show that the proposed algorithm has improved the sum rate when compared with the conventional greedy algorithm.

Satellite-Aerial Integrated Computing in Disasters: User Association and Offloading Decision

Long Zhang, Hongliang Zhang, Chao Guo, Haitao Xu, Lingyang Song, and Zhu Han

Abstract—In this paper, a satellite-aerial integrated computing (SAIC) architecture in disasters is proposed, where the computation tasks from two-tier users, i.e., ground/aerial user equipments, are either locally executed at the high-altitude platforms (HAPs), or offloaded to and computed by the Low Earth Orbit (LEO) satellite. With the SAIC architecture, we study the problem of joint two-tier user association and offloading decision aiming at the maximization of the sum rate. The problem is formulated as a 0-1 integer linear programming problem which is NP-complete. A weighted 3-uniform hypergraph model is obtained to solve this problem by capturing the 3D mapping relation for two-tier users, HAPs, and the LEO satellite. Then, a 3D hypergraph matching algorithm using the local search is developed to find a maximum-weight subset of vertex-disjoint hyperedges. Simulation results show that the proposed algorithm has improved the sum rate when compared with the conventional greedy algorithm.

I. INTRODUCTION

During disasters, one of the most immediate impacts is the collapse of communications infrastructure, e.g., destroyed base stations (BSs) in cellular wireless systems. In emergencies of this magnitude, the public and disaster response crews will be isolated from the outside due to the sudden interruption of wireless services. For the public, they need to broadcast their locations and send disaster information to outside community, e.g., data-rich, contextual multimedia via social platforms on user equipments (UEs). Meanwhile, disaster response crews are deployed to the affected region to perform a number of relief tasks by help of emergency equipments. For instance, unmanned aerial vehicles (UAVs) are placed as aerial devices for Internet of Things to participate in disaster sensing missions, e.g., situational awareness, reconnaissance, mapping, etc.

However, both ground UEs (GUEs) for the public and aerial UEs (AUEs) for rescue efforts cannot obtain wireless access due to breakdown of terrestrial BSs. Particularly, huge amount of data from these tasks running at GUEs and AUEs typically require sustainable computation resources. Therefore, it is imperative to implement a reliable and resilient wireless communications system for disaster response. Moreover, mobile edge computing (MEC) should be integrated in emergency networks

to provide efficient and flexible computing services for two-tier users. Owing to the inherent advantages of flexibility and mobility, UAVs has been utilized as flying BSs to provide cost-effective solutions for emergency wireless communications in disasters [1]. Nevertheless, UAVs mounted with MEC servers actually cannot improve computation performance due to their limited payloads and short flight time.

Compared with UAVs, high-altitude platforms (HAPs) have potentials to carry heavy 5G infrastructure payloads, to hover over the sky, and to provide longer endurance and larger area coverage [2]. These advantages make HAPs very suitable in emergency situations to offload computation intensive tasks from two-tier users via mounting MEC servers. In addition, for the ubiquitous 3D super-connectivity in global scale, satellite communications can be integrated with the HAP-aided computation offloading to boost the prospect of implementing the satellite-aerial integrated computing (SAIC). This integration not only provides satellite backhauling for global access, but also achieves enhanced computing capability. As the emerging architecture, the space-air-ground integrated networks have been recognized as a key enabler for 6G systems of 2030 [3].

Many recent works are devoted to the integration of satellite systems, aerial networks, and terrestrial communications [4]–[6]. In [4], Shi *et al.* used a greedy method to derive an optimal set of gateways in aerial network for cross-layer data delivery. In [5], Hu *et al.* formulated the joint resource allocation and user association problem as a competitive market model. In [6], Zhang *et al.* proposed a network slicing based architecture for space-air-ground integrated vehicular networks to achieve network agility. There are rare few efforts for computation offloading applied in scenarios with the combination of satellite, aerial, and terrestrial network layers. In [7], Cheng *et al.* proposed a space-air-ground integrated computing architecture wherein MEC and cloud computing are performed at the UAVs and the Low Earth Orbit (LEO) satellites, respectively. The computing offloading problem was modeled as a Markov decision process, and a machine learning method was used to optimize the offloading decision. In [8], Zhang *et al.* developed a dynamic network virtualization scheme to integrate the physical resources using a dynamic resource monitor in satellite MEC for ground UEs. A cooperative computation offloading model was proposed to achieve the parallel computation.

The issues with existing studies of computation offloading at space/air-assisted computing [7] and space-aided computing [8] are highlighted as (a) the setting of computing offloading, (b) the modeling of offloading decision, and (c) the allocation of computing resources via network functions virtualization.

L. Zhang is with the Department of Communications Engineering, Hebei University of Engineering, Handan 056038, China. H. Zhang and Z. Han are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA, and Z. Han is also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea. C. Guo is with the Beijing Electronic Science and Technology Institute, Beijing, 100070, China. H. Xu is with the Department of Communications Engineering, University of Science and Technology Beijing, Beijing 100083, China. L. Song is with the Department of Electronics Engineering, Peking University, Beijing 100871, China.

This work was supported in part by NSFC (61971032), Hebei Program for Top-notch Young Talents (BJ2017037), and NSF of Hebei (F2019402206).

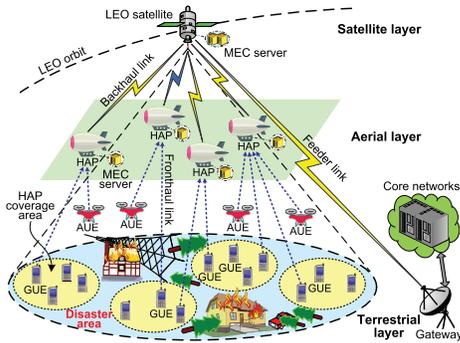


Fig. 1. An architecture of the SAIC in disaster scenario.

The above related works apply only to the scenario of ground users and do not consider the aerial users that also require computing resources. Besides, existing methods in [7] typically assume an ideal condition that the computing tasks are offloaded to the UAV-mounted MEC servers, which are however not practical in realistic scenarios due to UAVs' limited payloads. The specific user association relationship between users and computing servers is also not captured in [7] and [8].

Against this background, by help of HAPs, in this paper we propose a SAIC architecture to provide wireless coverage for two-tier users and offload computation tasks from them. Particularly, we focus on the uplink communications integrating both the fronthaul and backhaul links while designing the joint two-tier user association and satellite-aerial integrated offloading scheme in disasters. Main contributions of our work include:

- The sum rate maximization problem of the SAIC system is formulated as a 0-1 integer linear programming problem by jointly optimizing binary variables for two-tier user association and offloading decision.
- By designing a 3D mapping relation for two-tier users, HAPs, and the LEO satellite, we transform this problem into a weighted 3-uniform hypergraph model, in which the weight of hyperedge is calculated as the sum of the fronthaul uplink rate and computing rate.
- The optimization problem is thus converted to find a 3D hypergraph matching with maximum total weight. A local search based algorithm is proposed to enhance the overall searching performance.

The rest of the paper is organized as follows. We introduce the system model and formulate the problem in Section II. Section III presents the problem transformation and designs the local search based algorithm. Finally, we provide the simulation results in Section IV and conclude the paper in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. Network Model

Consider a disaster scenario in which there is no cellular coverage due to the destroyed terrestrial BSs for a finite time horizon T , as shown in Fig. 1. This scenario consists of two-tier users, i.e., N randomly distributed GUEs on the ground to disseminate disaster information to outside community, and M AUEs with fixed flying trajectories at a given altitude L_A to participate in disaster sensing missions. The proposed SAIC architecture is composed of three layers, i.e., terrestrial layer,

aerial layer, and satellite layer. Specifically, the disaster area in the terrestrial layer includes two-tier users with very limited computation capabilities, i.e., computing speed (in cycles/s) of processor cores, resulting in a large number of consumed time and energy to complete the execution of computation tasks. We represent the sets of GUEs and AUEs with $\mathcal{N} = \{1, 2, \dots, N\}$ and $\mathcal{M} = \{1, 2, \dots, M\}$, respectively. In the aerial layer, J HAPs with MEC servers, denoted by a set $\mathcal{J} = \{1, 2, \dots, J\}$, are deployed to execute the whole computation tasks offloaded from GUEs and AUEs. Each HAP is configured with multiple receive antennas and is quasi-static in the stratosphere at a fixed altitude L_H , for $L_A < L_H$. Lastly, an LEO satellite with an enhanced MEC server is placed at an orbital altitude L_S above the disaster area in the satellite layer. So it has the potential to execute the computation tasks uploaded from HAPs. The LEO satellite is equipped with multiple receive antennas operating in the Ka band, and is also connected to a terrestrial gateway via feeder link, providing access to core networks [9]. Here, a proper minimum elevation angle of communications for the LEO satellite is adopted to guarantee the entire coverage of all the HAPs during time horizon T .

For convenience, a 3D Cartesian coordinate system is used to describe the location of each entity in the SAIC architecture. The time horizon T is equally divided into K time slots with length $\frac{T}{K}$. Thus, the orbital location of the LEO satellite and the location of AUE $m \in \mathcal{M}$ mapped onto the horizontal plane coordinate at slot k are denoted by $\mathbf{q}[k] = (x[k], y[k])$ and $\mathbf{q}_m[k] = (x_m[k], y_m[k])$, respectively. Furthermore, the horizontal locations of HAP $j \in \mathcal{J}$ and GUE $n \in \mathcal{N}$ are defined as ψ_j and φ_n at each slot, respectively.

B. Communication Model

1) *Fronthaul Uplink*: For the fronthaul uplink, N GUEs and M AUEs with computation requirements in the terrestrial layer obtain wireless access and then transmit their computation tasks to J HAPs in the aerial layer. We consider that all the HAPs operate in the Ka band sharing the same resource block according to the ITU-R spectrum regulation [2]. To simplify analysis, it will be assumed that all of the GUEs and AUEs are assigned to a different sub-channel over the resource block to establish the fronthaul uplinks. In other words, co-channel interference during uplink communications can be fully avoided, and the channel allocation problem will not be considered in this work. We further assume that each GUE can only associate with at most one HAP, but a HAP can serve at most N_H GUEs at the same time. Let us introduce a binary variable as follows to represent the association relationship between GUE n and HAP j at slot k :

$$a_{n,j}[k] = \begin{cases} 1, & \text{if GUE } n \text{ is associated with HAP } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Note that $\sum_{j=1}^J a_{n,j}[k] \leq 1, \forall n$, and $\sum_{n=1}^N a_{n,j}[k] \leq N_H, \forall j$. Due to the damaged infrastructure in disasters, it will be possible to provide the ground-to-air (G2A) LOS communication link between GUE and HAP. Therefore, the G2A channel gain between GUE n and HAP j at slot k can be obtained as:

$$g_{n,j}[k] = \eta_0 d_{n,j}^{-\zeta_1}[k] = \frac{\eta_0}{\left(L_H^2 + \|\boldsymbol{\psi}_j - \boldsymbol{\varphi}_n\|^2\right)^{0.5\zeta_1}}, \quad (2)$$

where η_0 is the channel gain at a reference distance, $d_{n,j}[k]$ is the distance between GUE n and HAP j at slot k , and $\zeta_1 \geq 2$ is the path loss exponent. Correspondingly, the achievable rate (in bps/Hz) of GUE n for the fronthaul uplink at slot k can be modeled by:

$$R_n[k] = \sum_{j=1}^J a_{n,j}[k] \log_2 \left(1 + \frac{p_{n,j}[k] g_{n,j}[k]}{\sigma_j^2} \right), \quad (3)$$

where $p_{n,j}[k]$ is the transmit power of GUE n to HAP j at slot k , σ_j^2 is the variance of the AWGN at HAP j .

As for the fronthaul uplink from AUE to HAP, we suppose that each AUE can only associate with at most one HAP, but a HAP can serve at most M_H AUEs simultaneously. As such, to indicate the association relationship between AUE m and HAP j at slot k , a binary variable is also introduced, which can be given as:

$$b_{m,j}[k] = \begin{cases} 1, & \text{if AUE } m \text{ is associated with HAP } j, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Note that $\sum_{j=1}^J b_{m,j}[k] \leq 1, \forall m$, and $\sum_{m=1}^M b_{m,j}[k] \leq M_H, \forall j$. For the air-to-air (A2A) LOS communication link between AUE and HAP, we adopt a channel fading model by integrating free space path loss and miscellaneous atmospheric loss. With this channel model, the path loss between AUE m and HAP j at slot k can be specified by:

$$\begin{aligned} l_{m,j}[k] &= \left(\frac{4\pi f_C d_{m,j}[k]}{c} \right)^{\zeta_2} + l_A \\ &= \left(\frac{4\pi f_C \sqrt{(L_H - L_A)^2 + \|\boldsymbol{\psi}_j - \mathbf{q}_m[k]\|^2}}{c} \right)^{\zeta_2} + l_A, \end{aligned} \quad (5)$$

where f_C is the carrier frequency, $d_{m,j}[k]$ is the distance from AUE m to HAP j at slot k , c is the speed of light, $\zeta_2 \geq 2$ is the path loss exponent, and l_A is the atmospheric loss associated with the effect of oxygen and water vapour on fronthaul link. Thus, the A2A channel gain between AUE m and HAP j at slot k is denoted by $g_{m,j}[k] = \frac{1}{l_{m,j}[k]}$. Thereby, the achievable rate (in bps/Hz) of AUE m for the fronthaul uplink at slot k can be calculated as:

$$R_m[k] = \sum_{j=1}^J b_{m,j}[k] \log_2 \left(1 + \frac{p_{m,j}[k] g_{m,j}[k]}{\sigma_j^2} \right), \quad (6)$$

where $p_{m,j}[k]$ is the transmit power of AUE m to HAP j at slot k .

2) *Backhaul Uplink*: The computation tasks of HAPs may be also offloaded to the LEO satellite via the backhaul uplink due to the placement of enhanced MEC server at the LEO satellite. Different from the G2A and A2A LOS communication links, the air-to-space (A2S) backhaul uplink is determined by a Rician fading channel model with the AWGN [10], [11]. Specifically, the channel fading coefficient between HAP j and the LEO satellite is modeled as a circularly symmetric complex Gaussian random variable, i.e., $h_j = X_1 + X_2i$, where $X_1 \sim \mathcal{N}(\mu_1, \frac{\sigma^2}{2})$ and $X_2 \sim \mathcal{N}(\mu_2, \frac{\sigma^2}{2})$. For simplicity, each HAP is assumed to be assigned to a different sub-channel with equal bandwidth of W at the Ka band for accessing into the LEO

satellite. Thus, the achievable rate (in bps/Hz) of HAP j for the backhaul uplink at slot k is expressed as:

$$\begin{aligned} R_j^B[k] &= \log_2 \left(1 + \frac{p_j[k] |h_j|^2 d_j^{-\zeta_3}[k]}{\sigma_S^2} \right) \\ &= \log_2 \left(1 + \frac{p_j[k] |h_j|^2}{\sigma_S^2 ((L_S - L_H)^2 + \|\mathbf{q}[k] - \boldsymbol{\psi}_j\|^2)^{0.5\zeta_3}} \right), \end{aligned} \quad (7)$$

where $p_j[k]$ is the transmit power of HAP j to the LEO satellite at slot k , $|h_j|^2$ is the magnitude A2S channel gain square, $d_j[k]$ is the distance from HAP j to the LEO satellite at slot k , $\zeta_3 \geq 2$ is the path loss exponent, and σ_S^2 is the variance of the AWGN at the LEO satellite.

C. Computing Model

Under the SAIC architecture, the computation tasks of two-tier users can either be locally executed at the HAPs, or be offloaded through the backhaul uplink to and executed by the LEO satellite. To reach this goal, a decision on computation offloading should be made by each HAP. Let us assume that the LEO satellite can serve at most J_S HAPs for offloading. We then design an offloading decision binary variable to depict whether to fully offload for HAP j at slot k :

$$c_j[k] = \begin{cases} 1, & \text{if HAP } j \text{ decides to fully offload task,} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Note that $\sum_{j=1}^J c_j[k] \leq J_S$. When $c_j[k] = 0$, the local execution mode is applied for HAP j . However, when $c_j[k] = 1$, HAP j needs to offload its task to the LEO satellite. Here, we adopt a full computation offloading strategy for each HAP. That is, the whole computation task of HAP j is fully offloaded and processed by the LEO satellite when it decides to offload. For tractability, we use the size of computation data (in bits) composed of program codes and input parameters to describe the HAP's computation task. Hence, HAP j is assumed to own a computation task $\{I_j[k], O_j[k]\}$ at slot k to accomplish, where $I_j[k]$ and $O_j[k]$ are the data size of computation task executed by itself and by the LEO satellite, respectively.

1) *Local Execution Mode*: For local computing, we denote ϖ_j (in cycles/bit) as the number of processor cycles required to complete one bit information at HAP j . Thus, the total number of processor cycles required to execute the computation data $I_j[k]$ locally at slot k is $\varpi_j I_j[k]$. We further use f_j to denote the local computation capability of HAP j . Then, the total number of processor cycles required to compute the local computation task of HAP j during slot k can be written as $\frac{T}{K} f_j$. So, we can easily have $I_j[k] = \frac{T f_j}{K \varpi_j}$. The computing rate (in bps/Hz) with unit bandwidth for local execution at HAP j at slot k is accordingly given by:

$$R_j^{(1)}[k] = f_j / \varpi_j. \quad (9)$$

2) *Full Offloading Mode*: For full offloading, we concentrate on the uplink communications from HAPs to the LEO satellite, and neglect the computation time consumed at the LEO satellite thanks to its powerful computing capability. By considering the communication overhead (e.g., encryption and packer header) denoted by δ_j for HAP j , the actual data size

of computation task for HAP j to be offloaded at slot k is given by $\delta_j O_j[k] = \frac{WT}{K} R_j^B[k]$. Therefore, the offloading rate (in bps/Hz) for HAP j at slot k is expressed as:

$$R_j^{(2)}[k] = \frac{1}{\delta_j} \log_2 \left(1 + \frac{p_j[k] |h_j|^2 d_j^{-\zeta_3}[k]}{\sigma_S^2} \right). \quad (10)$$

By integrating the offloading decision, the actual computing rate for HAP j at slot k can be achieved by:

$$R_j[k] = (1 - c_j[k]) R_j^{(1)}[k] + c_j[k] R_j^{(2)}[k]. \quad (11)$$

To simplify notations, we set $\wp_{n,j}[k] = \log_2 \left(1 + \frac{p_{n,j}[k] g_{n,j}[k]}{\sigma_j^2} \right)$ and $\wp_{m,j}[k] = \log_2 \left(1 + \frac{p_{m,j}[k] g_{m,j}[k]}{\sigma_j^2} \right)$. Combining (3), (6), and (11), the sum rate of the system at slot k can be obtained as:

$$\begin{aligned} & R(a_{n,j}[k], b_{m,j}[k], c_j[k]) \\ &= \sum_{n=1}^N R_n[k] + \sum_{m=1}^M R_m[k] + \sum_{j=1}^J R_j[k] \\ &= \sum_{j=1}^J \left\{ \sum_{n=1}^N a_{n,j}[k] \wp_{n,j}[k] + \sum_{m=1}^M b_{m,j}[k] \wp_{m,j}[k] \right. \\ & \quad \left. + (1 - c_j[k]) \frac{f_j}{\varpi_j} + c_j[k] \frac{1}{\delta_j} R_j^B[k] \right\}. \end{aligned} \quad (12)$$

D. Problem Formulation

Under the above setup, we aim to maximize the sum rate of the system during time horizon T by jointly optimizing binary variables for two-tier user association and offloading decision. This optimization problem is mathematically formulated as:

$$\begin{aligned} & \max_{a_{n,j}[k], b_{m,j}[k], c_j[k]} \sum_{k=1}^K R(a_{n,j}[k], b_{m,j}[k], c_j[k]) \quad (13) \\ \text{s.t. C1: } & \sum_{j=1}^J a_{n,j}[k] \wp_{n,j}[k] \geq R_G^{\min}, \forall n, k, \\ \text{C2: } & \sum_{j=1}^J b_{m,j}[k] \wp_{m,j}[k] \geq R_A^{\min}, \forall m, k, \\ \text{C3: } & \sum_{n=1}^N a_{n,j}[k] \wp_{n,j}[k] + \sum_{m=1}^M b_{m,j}[k] \wp_{m,j}[k] \\ & \leq R_j^B[k], \forall j, k, \\ \text{C4: } & \sum_{j=1}^J a_{n,j}[k] \leq 1, \forall n, k, \quad \text{C5: } \sum_{n=1}^N a_{n,j}[k] \leq N_H, \forall j, k, \\ \text{C6: } & \sum_{j=1}^J b_{m,j}[k] \leq 1, \forall m, k, \quad \text{C7: } \sum_{m=1}^M b_{m,j}[k] \leq M_H, \forall j, k, \\ \text{C8: } & \sum_{j=1}^J c_j[k] \leq J_S, \forall k, \\ \text{C9: } & a_{n,j}[k], b_{m,j}[k], c_j[k] \in \{0, 1\}, \forall n, m, j, k, \end{aligned}$$

where R_G^{\min} and R_A^{\min} are the minimum rates of GUE and AUE, respectively. C1 and C2 are used to ensure the minimum rate requirements of GUE and AUE, respectively. C3 shows that the achievable rate of each HAP for the backhaul uplink must be larger than or equal to the sum rates of both GUEs and AUEs for the fronthaul uplink to this HAP. The association

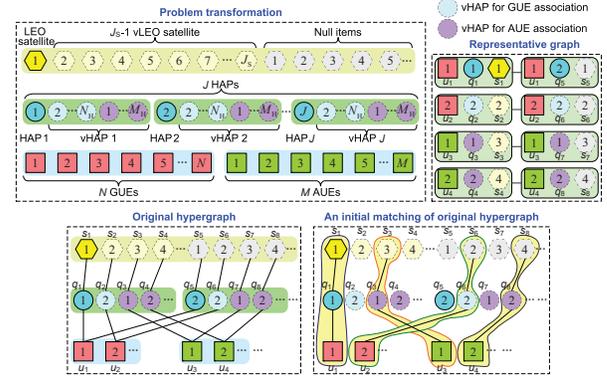


Fig. 2. Illustration of problem transformation, hypergraph construction, and representative graph generation.

constraints between GUE and HAP are demonstrated in C4 and C5, respectively. Besides, C6 and C7 imply that the association constraints between AUE and HAP. In C8, the offloading decision constraint for each HAP is further provided. Finally, C9 reflects the binary constraints for association variables and offloading decision variable.

III. PROBLEM TRANSFORMATION AND ALGORITHM DESIGN FROM HYPERGRAPH PERSPECTIVE

The optimization problem in (13) is a pure 0-1 integer linear programming problem, aiming to find an assignment of either 0 or 1 to the binary variables, such that the objective function is maximized and all the constraints are satisfied. Such kind of problem is NP-complete. When N , M , and J become large, it is very difficult to solve this problem via traditional methods. Considering that the inherent coupling constraints for two-tier user association with HAPs and data offloading to the LEO satellite at each slot, we shall transform the optimization problem in (13) into a weighted 3-uniform hypergraph model by establishing a one-to-one-to-one (3D) mapping relation for two-tier users, the HAPs, and the LEO satellite.

To obtain this mapping relation, let us first introduce some virtual elements for HAPs and the LEO satellite as well as several null items. As exhibited in Fig. 2, we precisely devise $N_H + M_H - 1$ virtual HAPs (vHAPs) for each HAP to associate two-tier users due to constraints C5 and C7. Thereby, we have $J(N_H + M_H)$ HAPs and vHAPs in the aerial layer. Based on constraint C8, $J_S - 1$ virtual LEO (vLEO) satellites are adopted to offload the computation tasks from J HAPs. To adapt to the local execution scenario, $\lambda(N_H + M_H)$ null items are utilized in the satellite layer, where $\lambda \in [1, J]$ is an integer. For the converted 3D mapping, the conventional graph model, generally used to describe pairwise relation between objects, cannot exactly analyze 3D relation among objects. Fortunately, the hypergraph model wherein every hyperedge can be formulated as a subset of the vertex set is very suited to characterize 3D mapping relation among multiple objects.

A. Hypergraph Construction

Definition 1. A hypergraph \mathcal{H} is an ordered pair $(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is a finite set (the vertex set), and $\mathcal{E} \subset 2^{\mathcal{V}}$ is a family of non-empty subsets of \mathcal{V} (the hyperedge set). A weighted

hypergraph is a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, w)$ that has a positive number $w(E_\ell)$ associated with every hyperedge $E_\ell \in \mathcal{E}$, called the *weight* of hyperedge E_ℓ , for $\ell = 1, 2, \dots, |\mathcal{E}|$.

For ease of exposition, let us use $\mathcal{U} = \mathcal{N} \cup \mathcal{M}$, \mathcal{Q} , and \mathcal{S} to denote the set of two-tier users, the set of $J(N_H + M_H)$ HAPs and vHAPs, and the set of a LEO satellite, $J_S - 1$ vLEO satellites, and $\lambda(N_H + M_H)$ null items, respectively. Given this scenario, every element in \mathcal{U} , \mathcal{Q} , and \mathcal{S} refers to a vertex of \mathcal{H} , and the union of \mathcal{U} , \mathcal{Q} , and \mathcal{S} stands for the vertex set of \mathcal{H} , i.e., $\mathcal{V} = \mathcal{U} \cup \mathcal{Q} \cup \mathcal{S}$. Due to the mobility of AUEs and LEO satellite at each slot, 3D mapping relation is also determined by dynamic time-variant feature. We thus use a slot-based combination $(u, q, s)^k$ to denote a hyperedge of \mathcal{H} at slot k , i.e., $E_\ell[k] = (u, q, s)^k$, while constraints C1-C3 are satisfied simultaneously, for $u \in \mathcal{U}$, $q \in \mathcal{Q}$, $s \in \mathcal{S}$. Then, the weight of hyperedge $E_\ell[k]$ is defined as the sum of the fronthaul uplink rate and the actual computing rate for $(u, q, s)^k$ at slot k , i.e., $w(E_\ell[k]) = R_u[k] + R_j[k]$, where $u = n$, for GUE n , or $u = m$, for AUE m . Thereby, the optimization problem is transformed into a weighted 3-uniform hypergraph model, as shown in Fig. 2. Based on original hypergraph \mathcal{H} , our target for the sum rate maximization problem during time horizon T is to find a 3D hypergraph matching that is a collection of $|\mathcal{U}|$ vertex-disjoint hyperedges with the maximum total weight at each slot.

B. Algorithm Design

For a 3D hypergraph matching, seeking a maximum-weight subset of vertex-disjoint hyperedges is NP-hard. Inspired by the local search idea with an increasing approximation ratio in polynomial time [12], we then adopt local search to design the hypergraph matching algorithm to find a suboptimal solution.

Definition 2. A *representative graph* G of original hypergraph \mathcal{H} is an ordinary graph, where every vertex $z_\ell[k] \in \mathcal{Z}$ of G stands for every hyperedge $E_\ell[k] \in \mathcal{E}$ of \mathcal{H} at slot k , and every edge of G corresponds to the adjacent hyperedges if they intersect with each other at least one vertex of \mathcal{H} . The *weight* of vertex $z_\ell[k]$ of G is equal to $w(E_\ell[k])$ of associated hyperedge $E_\ell[k]$ of \mathcal{H} at slot k .

As depicted in Fig. 2, vertex (u_1, q_1, s_1) of representative graph G refers to hyperedge $\{u_1, q_1, s_1\}$ of \mathcal{H} , and the edge between vertices (u_1, q_1, s_1) and (u_1, q_5, s_5) of representative graph G indicates that hyperedges $\{u_1, q_1, s_1\}$ and $\{u_1, q_5, s_5\}$ intersect one vertex, i.e., u_1 , in \mathcal{H} .

Definition 3. Given a representative graph G , a ϵ -claw is an induced subgraph C_ϵ whose center vertex connects to ϵ independent vertices, called *talons*, which forms an independent set \mathcal{I}_{C_ϵ} with ϵ vertices.

From Fig. 2, we find that there exists only 1-claw for center vertex (u_1, q_1, s_1) of G . Note that we can also search for ϵ -claw ($2 \leq \epsilon \leq \epsilon^*$) for center vertex (u_1, q_1, s_1) with the growing size of vertex set \mathcal{V} of \mathcal{H} . Let $\mathcal{X}[k]$ and $\Upsilon(\mathcal{X}[k])$ be an initial independent set of G and the sum of the weight of all the vertices in $\mathcal{X}[k]$ at slot k , respectively. Here, an initial independent set refers to an initial matching of original hypergraph (see Fig. 2). To obtain an initial independent set $\mathcal{X}[k]$ from G with vertex set \mathcal{Z} at slot k , a greedy algorithm is

Algorithm 1 Greedy Algorithm

Input: Original weighted hypergraph \mathcal{H}
Output: Initial independent set $\mathcal{X}[k]$ of representative graph G

- 1: Generate representative graph G with vertex set \mathcal{Z} based on \mathcal{H} .
- 2: Set $\mathcal{X}[k] = \emptyset$.
- 3: **while** $\mathcal{Z} \neq \emptyset$ **do**
- 4: Select a vertex $z^* \in \mathcal{Z}$ with maximum-weight of vertex in G .
- 5: Set $\Phi_{z^*}[k] = \emptyset$. Find all the adjacent vertices of z^* , and let $\Phi_{z^*}[k]$ be the set of adjacent vertices of z^* .
- 6: Set $\mathcal{X}[k] = \mathcal{X}[k] \cup \{z^*\}$ and $\mathcal{Z} = \mathcal{Z} - \{z^*\} \cup \Phi_{z^*}[k]$.
- 7: **end while**

Algorithm 2 3D Hypergraph Matching Algorithm

Input: Original weighted hypergraph \mathcal{H}
Output: 3D hypergraph matching \mathcal{E}^*

- 1: Generate representative graph G via original hypergraph \mathcal{H} .
- 2: Obtain an initial independent set $\mathcal{X}[k]$ of G via **Algorithm 1**.
- 3: **repeat**
- 4: Sort all the vertices of $\mathcal{X}[k]$ in an ascending order based on the weight of every vertex. Set $l = 1$.
- 5: **repeat**
- 6: Find all the adjacent vertices of the l -th vertex $\mathcal{X}(l) \in \mathcal{X}[k]$, and let $\Phi_{\mathcal{X}(l)}[k]$ be the set of adjacent vertices of $\mathcal{X}(l)$.
- 7: Sort all the vertices of $\Phi_{\mathcal{X}(l)}[k]$ in a descending order based on the weight of every vertex.
- 8: Set l -th vertex $\mathcal{X}(l)$ as a center vertex of ϵ -claw C_ϵ .
- 9: **for** $\epsilon = 1 \rightarrow 3$ **do**
- 10: Search for ϵ -claw C_ϵ from the set $\Phi_{\mathcal{X}(l)}[k]$ in G .
- 11: **if** there exists ϵ -claw C_ϵ satisfying $\Upsilon((\mathcal{X}[k] - \Phi(\mathcal{I}_{C_\epsilon}, \mathcal{X}[k])) \cup \mathcal{I}_{C_\epsilon}) > \Upsilon(\mathcal{X}[k])$ **then**
- 12: $\mathcal{X}[k] = (\mathcal{X}[k] - \Phi(\mathcal{I}_{C_\epsilon}, \mathcal{X}[k])) \cup \mathcal{I}_{C_\epsilon}$, go to step 3.
- 13: **end if**
- 14: **end for**
- 15: $l = l + 1$.
- 16: **until** $l \geq |\mathcal{X}[k]|$
- 17: **until** $|\mathcal{X}[k]| = |\mathcal{U}|$
- 18: **return** $\mathcal{E}^* = \{E_1^*[k], E_2^*[k], \dots, E_{|\mathcal{U}|}^*[k]\}$.

adopted in Algorithm 1 to approximately achieve a maximum-weight subset of edge-disjoint vertices in G . To improve the overall performance of achieving a 3D hypergraph matching, searching for ϵ -claw needs to be efficiently conducted via local search. By searching for ϵ -claw repeatedly, we propose a 3D hypergraph matching algorithm via local search to obtain a maximum-weight collection of $|\mathcal{U}|$ vertex-disjoint hyperedges. For tractability, let us use $\Phi(\mathcal{I}_{C_\epsilon}, \mathcal{X}[k])$ to denote the set of the adjacent vertices of an independent set \mathcal{I}_{C_ϵ} with ϵ vertices in an initial independent set $\mathcal{X}[k]$. The procedure of the proposed algorithm is summarized in Algorithm 2. Based on the output of Algorithm 2, we can obtain the maximum-weight subset \mathcal{E}^* of $|\mathcal{U}|$ vertex-disjoint hyperedges, which achieves the suboptimal solution to joint two-tier user association and offloading decision. Besides, the maximum sum rate of the system at slot k is derived as $R(a_{n,j}[k], b_{m,j}[k], c_j[k]) = \Upsilon(\mathcal{X}[k])$. So the sum rate of the system during time horizon T is specified by $\sum_{k=1}^K \Upsilon(\mathcal{X}[k])$. We wish to remark that the proposed algorithm by using local search improves the overall searching performance by identifying ϵ -claw C_ϵ in comparison with the traditional greedy algorithm.

IV. SIMULATION RESULTS

In this section, we present simulation results to assess the proposed scheme for the SAIC architecture in disasters. We

TABLE I
SIMULATION PARAMETERS.

Parameter	Value
Association constraint of HAP for GUE, N_H	20
Association constraint of HAP for AUE, M_H	4
Offloading constraint of the LEO satellite, J_S	4
Channel gain at reference distance 1m, η_0	-28dB
Variance of the AWGN at HAP, σ_j^2	-20dBm
Atmospheric loss, l_A	1.2dB
Channel fading coefficient, h_j	-15dB
Variance of the AWGN at the LEO satellite, σ_S^2	-25dBm
Number of cycles required to compute 1bit, ϖ_j	2.7×10^2 cycles/bit
Local computation capability of HAP, f_j	10^7 cycles/s
Communication overhead for HAP, δ_j	200bits

consider a 500m×500m disaster scenario consisting of $N = 100$ randomly distributed GUEs on the ground and $M = 20$ AUEs with fixed flying trajectories at altitude $L_A = 50$ m during time horizon $T = 12$ s. Above the disaster area, $J = 6$ HAPs are deployed in the stratosphere at altitude $L_H = 20$ km, and an LEO satellite is placed at orbital altitude $L_S = 500$ km. Based on the ITU-R spectrum regulation, the HAPs adopt the Ka band to communicate with two-tier users and the LEO satellite. Particularly, the 28GHz frequency bands is specifically selected for the fronthaul uplink, and 120 orthogonal sub-channels with equal bandwidth of 50MHz are allocated to each user. For the backhaul uplink, each HAP is assigned to an orthogonal sub-channel with equal bandwidth of 75MHz at 31GHz bands for accessing into the LEO satellite. For simplicity, the fixed transmit power allocation policy is used for two-tier users and the HAP. The transmit powers of each GUE to the HAP, of AUE to the HAP, and of the HAP to the LEO satellite at each slot are set to 280mW, 550mW, and 860mW, respectively. For the path loss exponent, we set $\zeta_1 = \zeta_2 = \zeta_3 = 2$. The details of other relevant parameters in simulations are listed in Table I.

In Fig. 3, we show the comparison of the system's sum rate between the proposed algorithm and the greedy algorithm under the varying number of time slots within time horizon $T = 12$ s. It is observed that the proposed algorithm provides a higher sum rate of the system than that of the greedy algorithm. This is because by incorporating ϵ -claw C_ϵ , our algorithm works in its best efforts to seek the suboptimal solution achieving the increase of the overall searching performance. As expected, as the number of time slots increases, the system's sum rate increases as well for both the algorithms. It can be concluded that with the increasing number of time slots, the number of two-tier user association and offloading decision also increases in spite of the shortened slot length. That results in the increase of the accumulated rate. This highlights the importance of properly tuning the number of time slots allowing, theoretically, the better sum rate for the system.

V. CONCLUSION

This paper investigated the problem of joint two-tier user association and offloading decision for the SAIC architecture in disasters. The optimization problem of the sum rate maximization was formulated as a 0-1 integer linear programming problem. By deriving a 3D mapping relation for two-tier users, the HAPs, and the LEO satellite, this problem was transformed

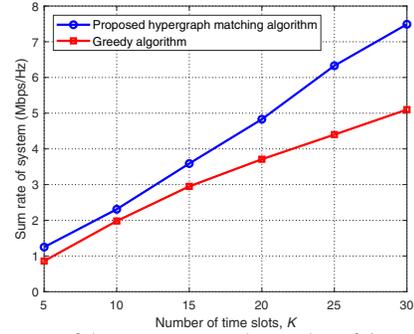


Fig. 3. The sum rate of the system versus the number of time slots comparing the proposed algorithm and the greedy algorithm.

into a weighted 3-uniform hypergraph model. A local search based 3D hypergraph matching algorithm was devised aiming to obtain the suboptimal solution to two-tier user association and offloading decision. Our results show that the proposed algorithm achieves good performance with significant increase on the sum rate, indicating its potential for a practical design.

REFERENCES

- [1] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K.-K. Wong, "UAV-assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.
- [2] A. Mohammed, A. Mehmood, F.-N. Pavlidou, and M. Mohorcic, "The role of high-altitude platforms (HAPs) in the global wireless connectivity," *Proc. IEEE*, vol. 99, no. 11, pp. 1939–1953, Nov. 2011.
- [3] H. Yanikomeroglu, "Integrated terrestrial/non-terrestrial 6G networks for ubiquitous 3D super-connectivity," in *Proc. ACM MSWIM*, Montreal, QC, Canada, Oct. 2018.
- [4] Y. Shi, J. Liu, Z. M. Fadlullah, and N. Kato, "Cross-layer data delivery in satellite-aerial-terrestrial communication," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 138–143, Jun. 2018.
- [5] Y. Hu, M. Chen, and W. Saad, "Competitive market for joint access and backhaul resource allocation in satellite-drone networks," in *Proc. IEEE NTMS*, Canary Islands, Spain, Jun. 2019.
- [6] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.
- [7] X. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.
- [8] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan. 2019.
- [9] B. Li, Z. Fei, C. Zhou, and Y. Zhang, "Physical-layer security in space information networks: A survey," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 33–52, Jan. 2020.
- [10] J. Du, C. Jiang, J. Wang, Y. Ren, S. Yu, and Z. Han, "Resource allocation in space multiaccess systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 2, pp. 598–618, Apr. 2017.
- [11] B. Li, Z. Fei, Z. Chu, F. Zhou, K.-K. Wong, and P. Xiao, "Robust chance-constrained secure transmission for cognitive satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4208–4219, May 2018.
- [12] L. Zhang, H. Zhang, L. Yu, H. Xu, L. Song, and Z. Han, "Virtual resource allocation for mobile edge computing: A hypergraph matching approach," in *Proc. IEEE GLOBECOM*, Waikoloa, HI, USA, Dec. 2019.