



**QUEEN'S
UNIVERSITY
BELFAST**

UAV-assisted Emergency Communications in Social IoT: A Dynamic Hypergraph Coloring Approach

Wang, B., Sun, Y., Sun, Z., Duong, T. Q., & Nguyen, L. D. (2020). UAV-assisted Emergency Communications in Social IoT: A Dynamic Hypergraph Coloring Approach. *IEEE Internet of Things Journal*.
<https://doi.org/10.1109/JIOT.2020.2988445>

Published in:
IEEE Internet of Things Journal

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights
© 2020 IEEE.

This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

UAV-assisted Emergency Communications in Social IoT: A Dynamic Hypergraph Coloring Approach

Bowen Wang, Yanjing Sun, *Member, IEEE*, Zhi Sun, *Senior Member, IEEE*, Long D. Nguyen, Trung Q. Duong, *Senior Member, IEEE*

Abstract—In this paper, we address social-awareness property and unmanned aerial vehicle (UAV) assisted information diffusion in emergency scenarios, where the UAVs can disseminate alert messages to a set of terrestrial users within their coverage, and then these users can continuously disseminate the received data packets to their socially connected users in a device-to-device (D2D) multicast manner. In this regard, we have to solve both the dynamic cluster formation and spectrum sharing problems in stochastic environments, since both the UAVs and terrestrial users may arrive or depart suddenly. For the cluster formation problem, considering that the data rate of a multicast cluster is determined by the member with worst link condition, we formulate it as a many-to-one matching game and adopt the rotation-swap algorithm to maximize the expected number of users receiving the alerting messages in each time slot. For the dynamic spectrum sharing problem, aiming at eliminating the interference while minimizing the channel switching cost, we propose a dynamic hypergraph coloring approach to model the cumulative interference and maintain the mutual interference at a low level by exploring a small number of vertices, when the graph is dynamically updated, i.e., the insertion/deletion of vertex/edge. Moreover, we prove some crucial properties including global stability, convergence, and complexity. Finally, simulation results shows that our proposed approach can achieve a better trade-off among the information diffusion speed, channel switch cost, and complexity.

Index Terms—Emergency communications, Social Internet of Things, unmanned aerial vehicle, D2D multicast, matching theory, graph theory.

I. INTRODUCTION

AS a promising paradigm to change the way human beings interact with and perceive the physical world,

This work was supported in part by the Key Research & Development Project for Science and Technology of Xuzhou, China (KC18105), in part by the National Natural Science Foundation of China (51734009, 61771417, 51804304). The work of T. Q. Duong was supported in part by the Royal Academy of Engineering (RAEng) under the RAEng Research Fellowships scheme Grant (RF1415\14\22) and in part by a Researcher Links grant (527612186), under the Newton Fund partnership. The grant is funded by the UK Department for Business, Energy and Industrial Strategy and delivered by the British Council, for further information, please visit www.newtonfund.ac.uk.

Bowen Wang and Yanjing Sun are with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China (e-mail: {bowenwang, yjsun}@cumt.edu.cn).

Bowen Wang and Trung Q. Duong are with the School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, BT7 1NN, UK (e-mail: trung.q.duong@qub.ac.uk).

Zhi Sun is with the Department of Electrical Engineering, The State University of New York at Buffalo, Buffalo, NY 14260, USA (e-mail: zhisun@buffalo.edu).

Long D. Nguyen is with the Duy Tan University, Vietnam (e-mail: dinh-longhcmut@gmail.com).

Internet of Things (IoT) is expected to achieve the goal of "Everything Connected", where smart objects with a high level of awareness, such as home appliances, wearable devices, phones, and vehicles are able to "talk" with each other. Recent years has witnessed the maturity of IoT, especially the acquirement of its social flavor, which promotes the emergence of Social IoT (SIoT) [1]. According to the SIoT paradigm, the IoT devices can build their inter-device social networks, and exchange information socially. There are some existing works investigating the application of SIoT into practical scenarios [2]. However, few works have studied the application of SIoT into emergency scenarios, where affected users and rescue teams need to be promptly informed about the environmental conditions, alert messages, and refuge positions. In [3], the authors investigated the joint cluster formation and route selection problem in disaster relief networks. In [4], the authors proposed a social-aware D2D-enhanced scheme to minimize the information diffusion time in emergency scenarios. However, these works only considered the case where the base stations (BSs) can manage the resource allocation and thus the performance may be degraded if the BSs are damaged during an unexpected disaster.

As a novel and attractive research area, unmanned aerial vehicles (UAVs) have the potential to provide some degrees of freedom in both time (available on demand) and space (controllable mobility) domains, which means that the wide deployment of UAVs can be useful for traffic offloading [5] and coverage extension [6]. In particular, the strong line-of-sight (LoS) links of UAV mounted BSs can be utilized to quickly recover the communication services in post-disaster scenarios [7]–[12]. In [9], the authors proposed a UAV placement scheme to maximize the cellular connectivity while minimizing the energy consumption in post-disaster networks. Considering the limited transmission distance in large-scale UAV communications, the authors in [10] proposed a UAV-assisted multi-hop transmission scheme to extend the wireless coverage in emergency communications. In [11], the authors investigated how to accommodate the air-to-ground (A2G) communications and ground-to-ground (G2G) communications concurrently by utilizing the non-orthogonal multiple access (NOMA) scheme. In [12], the authors studied the UAV-assisted machine-to-machine (M2M) communications in disaster relief networks. However, the application of social awareness into UAV-assisted emergency communications has not been addressed in these works. In practice, the UAVs will first disseminate the data packets in a multicast manner to ensure that as many surviving devices as possible can receive

alert messages [13]. After receiving these alert messages, the survivors will directly share with their socially friendly users, such as family members and colleagues. Considering the limited battery capacity of UAVs, if each UAV gives higher priority to those users with high social centrality, the information diffusion speed will be intuitively improved [4].

Herein, we consider both the UAV-enabled multicast and terrestrial device-to-device (D2D) multicast in information diffusion process. The A2G links are dominated by LoS and their channel gains are much higher than the channel gains of G2G links for terrestrial communication. However, the strong LoS links also cause severe interference to G2G links and thus an efficient interference management method is needed. In [13], the authors investigated the trajectory design to minimize the task completion time for UAV-enabled multicast in a single UAV scenario. For multi-UAV multicast scenarios, the cluster formation and spectrum sharing problems should be addressed. In [14], a matching theory based cluster formation scheme was proposed to optimize the social-aware rate. However, this paper just considered the static and deterministic scenario. In emergency communications, the dynamics and stochasticity have to be addressed since some information may appear or change, and cannot be known a priori. For example, a surviving device who received the alert message as a cluster member in current time slot may become the cluster head in the potential time slot, while some newcomers may arrive in the LoS region of the UAVs. Thus, the cluster structure will change dynamically, and as a result, the spectrum sharing strategy should be updated dynamically, which brings new challenges for resource allocation in such a dynamic topology.

A. Related Works

Recently, there are some existing works investigating the channel allocation problem in various scenarios. In [15], the authors utilized the centralized optimization technique to solve the resource allocation problems in NOMA-enhanced D2D-capable cellular networks. In [16], a Huffman tree based algorithm was proposed to solve the cooperative channel allocation problem in wireless mesh networks with multi-interface. In [17], [18], the authors utilized deep learning approach to solve the traffic load prediction based channel allocation problem and partially overlapping channel assignment problem in software defined network (SDN)-IoT, respectively. However, these works did not couple users' mobility with channel allocation problems. In [19], the authors proposed an anti-coordination game based scheme to solve the dynamic spectrum sharing problem in the scenario where UAVs and D2D users can share the same spectrum. In [20], the authors proposed a potential game based online channel selection scheme for multicluster Flying Ad-Hoc Network (FANET). However, the computational complexity of the proposed schemes in both papers is too high in dynamic and stochastic environment, since each user has to be traversed more than once when the state changes. If the response time cannot adapt to the dynamic topology change, the resource allocation decision will be invalidated and degrade the system performance, which is the main challenge of resource allocation in stochastic

and dynamic environments. As a powerful tool for resource allocation in dynamic network topology, graph theory can solve the channel allocation problem with lower complexity, using bipartite graph matching [21] or graph coloring [22]. In a word, we abstract each device as a vertex, the interference relationship between two devices as an edge, and different channels as different colors. The graph coloring applied into channel allocation problem is to assign appropriate colors (channels) to vertexes (devices) and no two adjacent vertexes share the same color [22].

In emergency scenarios, the abstracted graph is highly dynamic, i.e., the vertexes or edges will be potentially inserted or deleted, which poses a challenge to the algorithm design for the graph coloring problem. In [23], the authors proposed a DC-Local algorithm to solve the dynamic graph coloring problem. However, the local update strategy not only has bad coloring consistency but also results in high complexity consumption since it is affected by the insertion/deletion order of the edges/vertexes. Coloring consistency, which means that the coloring strategy is independent of the updating orders, is of great importance in wireless communications scenarios. For instance, power consumption is a critical issue for the availability of IoT devices, and most of it is caused by channel switch [24]. Coloring consistency can save the channel switch cost by avoiding unnecessary channel allocation triggered by the dynamics [25]. In [26], the authors proposed a color-propagation based algorithm by exploring a small part of vertexes in each update, and thus the complexity is further reduced. In [27], the authors introduced the dynamic graph coloring into D2D communications. However, the conventional graph can only model the interference between two users, which is not accurate in practice. As an effective tool to model the interrelationships between multiple users, hypergraph can be integrated in 5G standard to solve the resource allocation problems with tractable complexity [28]–[30]. In [30], the authors proposed a hypergraph-based 3D matching method to solve the user-relay-channel matching problem in D2D-Assisted machine-type communication networks. In [31], the authors introduced the hypergraph into D2D communications, and proposed a static hypergraph coloring approach to model the cumulative interference. However, the dynamic hypergraph coloring problem has not been well addressed yet.

B. Main Contributions

Based on the above discussion, the contributions of this paper are summarized as follows:

- *Modeling*: We design a novel framework by exploring the inter-device social ties to improve the information diffusion speed in UAV-assisted emergency communication scenarios. In particular, the UAVs will disseminate the emergency messages to terrestrial devices, who then continuously disseminate the messages to their social friendly users in a D2D multicast manner. Specifically, the proposed scheme comprehensively solves the dynamic cluster formation and dynamic spectrum sharing problems with security assured.
- *Algorithm*: For the cluster formation problem, considering that the data rate of a multicast cluster is determined by

the user with worst link condition, we formulate it as a many-to-one matching game with externality and adopt the rotation-swap algorithm to maximize the information diffusion speed in each time slot. For the dynamic spectrum sharing problem, we depict this problem with the framework of hypergraph and simplify the hypergraph structure to a directed graph. Based on the technique for dynamic graph coloring in [26], we propose a dynamic hypergraph coloring approach to maintain the mutual interference level with the lower channel switching cost guaranteed, when the graph is dynamically updated.

- **Validations:** We comprehensively prove the global stability, convergence, and complexity properties of our proposed approach. Last but not least, under various scenarios, simulation results demonstrate that our proposed approach can achieve a better trade-off between the information diffusion speed and complexity compared with existing methods.

Note that the main idea of the technique for dynamic graph coloring comes from [26], but there are some differences between [26] and our work. Firstly, the authors in [26] solved the dynamic coloring problem in conventional graph, in which one edge can only connect two nodes. Considering that the conventional graph cannot accurately model the cumulative mutual interferences from more than one interference source, we conceive the dynamic network topology with hypergraph, in which one edge can connect more than two nodes. In this way, we can model the cumulative mutual interferences more accurately. Since the structure of hypergraph is more complex than conventional graph, coloring the hypergraph is more difficult than coloring the conventional graph. In detail, the chromatic number constraint in conventional graph is each node's degree while the chromatic number constraint in hypergraph is each node's monodegree. Besides, one node will change its color if one of its neighbor chooses the same color in conventional graph whilst one node will change its color only if all of its neighbors choose the same color in conventional graph. Secondly, in order to adapt to the property of hypergraph, we slightly modify the color propagation strategy proposed by [26] in graph construction and assign color stages. To further bound the complexity, we briefly introduce how to migrate some tricks from [26] to our proposed scheme.

The rest of this paper is outlined as follows. The architecture for UAV-assisted emergency communications is presented and then the optimization problem is elaborated in Section II. The optimization problem is decoupled into two sub-problems and then transformed in Section III. The dynamic hypergraph coloring approach is described in Section IV. Simulation results are provided in Section V, followed by conclusion in Section VI. The main parameters and variables are summarized in Table I for ease of reading.

II. SYSTEM MODEL AND PROBLEM FORMATION

As shown in Fig. 1, a sudden disaster struck the gray region and the BSs are damaged due to power cut. To recover communication services, a set of UAVs are employed as flying BSs to deliver the emergency information file with size F to

TABLE I: Main Parameters and Variables

Symbol	Description
$U(t), S(t), \mathcal{L}$	Set of UAVs, surviving devices, and channels
$g_{i,j}(t), R_{i,j,l}(t), E_j^{hov}$	Power gain, data rate, hovering energy
$CH(t), CM(t)$	The set of cluster heads and members
$B_i, R_{C_j}(t)$	Social centrality and data rate of cluster C_j
ω, β	Cluster formation and channel allocation variable
y, ϕ	Interference state and indicator function
CI, IC	Cumulative interference and capacity
$G_s(t), G_H(t), G_o(t)$	Social graph, hypergraph, and OCG
$nb(v), nb^+(v), nb^-(v)$	Set of neighbors, out-neighbors, and in-neighbors
$id(v), f, \mathcal{C}, v.color$	ID, coloring function, color set, and color
$mdeg(v), deg(v)$	monodegree in hypergraph, and degree in OCG
$deg^+(v), deg^-(v)$	Out-degree and in-degree in OCG
$\Sigma(G_o(t)), \chi(G_o(t))$	Global oriented coloring and property
$C_{cnt}^u(c), R_{can}^u$	Color Counts and recolor Candidates

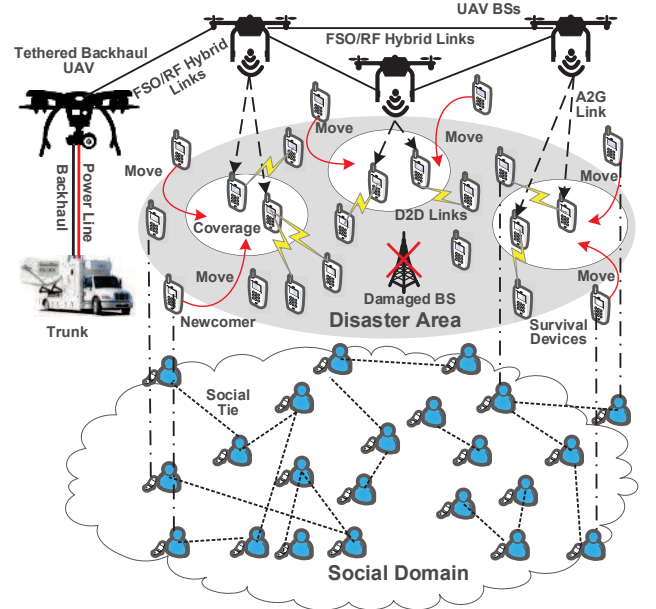


Fig. 1: UAV-assisted emergency scenarios in SIoT.

the surviving cellular users in a multicast manner [32]. Then these users can relay the received data packets to their socially connected users in a D2D multicast way. Generally, in the case that the whole cellular infrastructure is destroyed, only the satellite can provide the available backhaul connection. However, the UAV platform cannot be equipped with satellite transceiver equipment [9]. Inspired by [9], we use a post-disaster tethered backhaul UAV which can connect the core network and charge using a cable. Meanwhile, the other end of the cable is connected to a truck that is equipped with satellite transceiver equipment. In this way, the UAVs can be connected to the core network. Thus, we assume that the UAVs are connected to each other using hybrid free-space optics/radio frequency (FSO/RF) links and one of the UAVs acquires the backhaul connection from a tethered backhaul UAV installed hundreds of meters from the disaster area [9]. The tethered backhaul UAV provides connectivity to the core network to the whole flying network via hybrid FSO/RF links. It also

acts as a central controller to manage the resource allocation.

In this paper, we consider a time-slotted scenario containing $N(t)$ UAVs and $M(t)$ surviving devices at time slot t . The UAV set and surviving device set are given by $U(t) = \{u_j\}_{j=1}^{N(t)}$ and $S(t) = \{s_i\}_{i=1}^{M(t)}$, respectively. The time period \mathcal{T} is discretized into T slots with duration Δ , i.e., $\mathcal{T} = T\Delta$, and the set of cumulative slots is denoted by $\{1, \dots, t, \dots, T\}$. In each time slot, both UAVs and terrestrial devices are assumed to have fixed positions. We divide the disaster region into $N(1)$ subregions based on the distribution of surviving devices and each UAV provides cellular coverage to one subregion, which can be formulated as a N-way graph partition problem [33]. For simplicity, we invoke the multilevel graph partition algorithm in [33] to partition the graph into $N(1)$ sub-graphs, where the vertex number of each sub-graph is roughly equal. For a given region, each UAV will determine its real-time location based on the K-means methods [7]. In practice, the network topology changes in consecutive slots. The first reason is that UAVs have limited battery capacity, which means that UAVs will depart for recharge when their battery capacity is below a threshold, and then be airborne again. Similar to [34], we assume that batteries having to be recharged is quantified by the air-time ratio $\epsilon = \text{air time} / (\text{air time} + \text{charge time})$. The second reason lies in the dynamics of terrestrial devices. For example, the rescuers and survivors arrive stochastically in the disaster region and their mobility should also be considered. Herein, we assume that the user arrival process follows the Poisson distribution with expectation λ .

In this paper, we consider the orthogonal frequency division multiple access (OFDMA) system, where heterogeneous devices can share the same spectrum for communication. The available channel set is denoted as $\mathcal{L} = \{l\}_{l=1}^L$ and the unit bandwidth of one subchannel is B . The channel state is regarded as quasi-static in each time slot. Without loss of generality, we consider the 3D Cartesian coordinate model. For surviving device s_i and UAV u_j , their real-time coordinates are denoted as $C_{s_i}(t) = (x_{s_i}(t), y_{s_i}(t), 0)$, and $C_{u_j} = (x_{u_j}(t), y_{u_j}(t), h_{u_j}(t))$, respectively. The distance $d_{i,j}$ between any two devices u_j and s_i can be represented by the Euclidean distance between their coordinates, i.e.,

$$d_{i,j}(t) = \sqrt{(x_{s_i}(t) - x_{u_j}(t))^2 + (y_{s_i}(t) - y_{u_j}(t))^2 + h_{u_j}^2(t)}. \quad (1)$$

Based on the practical measurement results in [35], the A2G communications can be approximated by LoS channel model very well. Hence, the A2G channel from the UAV to the terrestrial device is assumed to be dominant by LoS links [36], [37]. The power gain from the UAV u_j to a terrestrial device s_i is expressed as

$$g_{i,j}(t) = \eta_0 d_{i,j}^{-\alpha_A}(t), \quad (2)$$

where α_A represents the path loss exponents for A2G link and η_0 represents the power gain of a LoS channel with unit reference distance, i.e., $d_0 = 1$ m.

Different from the assumption that all G2G links are assumed to be NLoS [11], we use the practical Rayleigh fading channel to model the regular D2D link [38]. The power gain

from the terrestrial device d_i to another device d_k is expressed as

$$g_{i,k}(t) = d_{i,k}^{-\alpha_G}(t) |h_{i,k}|^2, \quad (3)$$

where α_G denotes the path loss exponents for G2G link and $h_{i,k}$ is the complex Gaussian channel coefficient which follows the distribution of $\mathcal{CN}(0, 1)$.

Based on the channel model above, the transmission rate of the A2G link from the UAV u_j to a terrestrial devices s_i selecting channel l at time slot t is given by

$$R_{i,j,l}(t) = B \log_2 \left(1 + \frac{p_j g_{i,j}(t)}{I_{j',i}^A(t) + I_{i',i}^G(t) + N_0} \right), \quad (4)$$

where p_j is the transmission power of the UAV u_j , $I_{j',i}^A(t) = \sum_{u_{j'} \in U_l(t) \setminus u_j} p_{j'} g_{j',i}(t)$ represents the co-channel interference of A2G links and $U_l(t)$ is a set of UAVs sharing channel l , $I_{i',i}^G(t) = \sum_{s_{i'} \in S_l(t) \setminus s_i} p_{i'} g_{i',i}(t)$ represents the co-channel interference of G2G links and $S_l(t)$ is a set of terrestrial devices sharing channel l . N_0 denotes the variance of Gaussian noise.

Similarly, the transmission rate of the G2G link from the terrestrial devices s_i to s_k selecting channel l at time slot t is given by

$$R_{i,k,l}(t) = B \log_2 \left(1 + \frac{p_i g_{i,k}(t)}{I_{j,k}^A(t) + I_{i',k}^G(t) + N_0} \right). \quad (5)$$

In this paper, we mainly consider the energy consumption caused by hovering, and ignore the energy consumption caused by communication, because this part is much lower than the former [9]. By following the energy consumption model in [9], the hovering energy denoted by E_j^{hov} for UAV u_j is expressed as

$$E_j^{hov} = \sqrt{\frac{(m_{tot}g)^3}{3\pi r_p^2 n_p^2 \rho}} T^{hov}, \quad (6)$$

where m_{tot} denotes the UAV mass, g denotes the earth gravity, and ρ denotes the air density, respectively. r_p and n_p represent the radius and number of propellers, respectively [9].

In social domain, we consider the impact of social relationships on the performance of emergency information diffusion. By analogy to the SIoT paradigm [1], we consider the four types of inter-device social ties: the Ownership Object Relationship (OOR) implies that two objects belong to the same owner, the Co-Location Object Relationship (CLOR) is established when two objects are close in distance; the Co-target Object Relationship (CTOR) is built when two objects collaborate on the same rescue task; the Social Object Relationship (SOR) can measure the interaction level between two objects. Intuitively, if one device receives the emergency information, it will directly forward this information to the devices having any kinds of social ties with it. For example, two devices in the same rescue team will exchange the real-time emergency information since they have CTOR. For simplicity, we introduce the social graph $G_s(t) = (S(t), E_s)$ to model the inter-device social network, where $E_s = \{(i, i') | st_{i,i'} = 1, \forall s_i, s_{i'} \in S(t)\}$. The binary variable $st_{i,i'}$ can determine whether there exists a social tie between s_i and $s_{i'}$. In this

paper, we only consider the undirected edge and the social ties are unchanged in consecutive time slots.

Based on the above discussion, the optimization problem in this paper is to maximize the expected number of surviving devices receiving the emergency messages in information diffusion process, which can be formulated as

$$\max \frac{1}{T} \sum_{t=1}^T |o_t|, \quad (7)$$

where $O = \{o_t\}_{t=1}^T$ represents the information diffusion process and $|o_t|$ is the number of surviving devices receiving the emergency messages successfully in time slot t . It is obvious that we should jointly tackle the cluster formation sub-problem and spectrum sharing sub-problem in each time slot. In the next section, we will describe how to decouple the problem (7) into two sub-problems and how to transform the two sub-problems from the perspective of matching theory and graph theory, respectively.

III. PROBLEM TRANSFORMATION

In this section, we decouple the problem (7) into two sub-problems: cluster formation and spectrum sharing. Given a determined channel allocation decisions, UAVs and those terrestrial users who have received the emergency information act as cluster heads and then form clusters. Then, they will forward the received information to those their cluster members. Given determined clustering structures, cluster heads will dynamically change their spectrum sharing strategy. We transform the cluster formation sub-problem into a many-to-one matching problem with externality and the spectrum sharing problem into a hypergraph coloring problem, respectively.

A. Dynamic Cluster formation

For UAV-enabled multicast or D2D multicast system, the transmission rate of one cluster is determined by the cluster member with the worst link condition. For ease of analysis, all UAVs and terrestrial devices are divided into cluster head $CH(t) = \{h_j\}_{j=1}^{N_h(t)}$ and cluster member $CM(t) = \{m_i\}_{i=1}^{N_m(t)}$, which means that there are $\{1, \dots, C_j, \dots, C_{N_h(t)}\}$ clusters in time slot t . The transmission rate of cluster C_j equals to the minimum achievable rate among cluster members in C_j . We use a binary variable $\omega_{i,j,t} = 1$ or 0 to determine whether m_i belongs to cluster C_j in time slot t . Therefore, the transmission rate of C_j is given by

$$R_{C_j}(t) = \min\{R_{i,j,t}(t) | \omega_{i,j,t} = 1, \forall m_i \in CM(t)\}. \quad (8)$$

Besides, we also consider the social centrality in cluster formation process. Intuitively, if the cluster head gives higher priority to those cluster members with higher social centrality, these cluster members received the messages will act as cluster head in the next time slot and forward the emergency messages to their social connected cluster members so that the information diffusion speed will be improved. Herein, we

use the betweenness centrality in [39] to measure the social centrality of m_i , which can be given by

$$B_i = \sum_{j=1}^{N_m(t)} \sum_{k < j} \frac{g_{j,k}(i)}{G_{j,k}}, \quad (9)$$

where $G_{j,k}$ is the number of shortest links between node i and node j based on social graph $G_s(t)$, and $g_{j,k}(i)$ denotes the number of those shortest links that pass node i .

Hence, in cluster formation phase, the transmission rate can guarantee the information diffusion speed in the current time slot while social centrality can guarantee the information diffusion speed in the potential time slot. Note that the optimization object is different for cluster heads with different roles. The UAVs as cluster heads will focus both the link condition and social centrality of cluster members, while terrestrial device as cluster head will focus on the link condition and whether the social tie exists ($st_{i,j} = 1$). Because one terrestrial device only informs those social-connected devices. Based on the above discussion, the optimization object of the UAVs in cluster formation phase at time slot t can be formulated as

$$\mathbf{P1}: \max_{\omega(t)} \sum_{t=t'}^{t'+1} \sum_{i=1}^{N_m(t)} \sum_{j=1}^{N_h(t)} \omega_{i,j,t} B_i R_{C_j}(t), \forall h_j \in U(t), \quad (10a)$$

$$s.t. \quad \Delta R_{C_j}(t) \geq F, \forall h_j \in CH(t); \quad (10b)$$

$$E_j^{hov} \leq E_j^{th}, \forall h_j \in U(t); \quad (10c)$$

$$\sum_{t=t'}^{t'+1} \sum_{i=1}^{N_m(t)} \omega_{i,j,t} \leq q_j, \sum_{t=t'}^{t'+1} \sum_{j=1}^{N_h(t)} \omega_{i,j,t} \leq 1, \quad (10d)$$

where (10b) makes sure that the received data size in each time slot should be larger than the size of emergency information file F . (10c) is to ensure that the energy consumption cannot exceed a given threshold, otherwise this UAV will depart for recharge. (10e) ensures that one cluster head can deliver emergency information to at most q_j survivors per time slot, while one cluster member can receive emergency information from at most one cluster head per time slot.

Similarly, the optimization object of the terrestrial devices as cluster heads at time slot t can be formulated as

$$\mathbf{P2}: \max_{\omega(t)} \sum_{t=t'}^{t'+1} \sum_{i=1}^{N_m(t)} \sum_{j=1}^{N_h(t)} \omega_{i,j,t} st_{i,j} R_{C_j}(t), \forall h_j \in S(t), \quad (11a)$$

$$s.t. \quad C1: \sum_{t=t'}^{t'+1} \sum_{i=1}^{N_m(t)} \omega_{i,j,t} \leq q_j, \sum_{t=t'}^{t'+1} \sum_{j=1}^{N_h(t)} \omega_{i,j,t} \leq 1, \quad (11b)$$

where (11b) ensures that one cluster head can deliver emergency information to at most q_j survivors per time slot, while one cluster member can receive emergency information from at most one cluster head per time slot. As a powerful tool to model the users' selfishness in resource allocation, the matching theory can be utilized to simultaneously optimize multiple objectives for agents with different roles, which can be naturally applied to solve the problem (10) and (11).

It is obvious that the problem (10) and (11) belong to the category of many-to-one matching problem since one cluster head can serve more than one cluster members while one

cluster member can only select one cluster head. However, different from the conventional many-to-one matching problem such as classic hospital problem, the preference will dynamically change in the matching process, due to the fact that transmission rate of one cluster may be affected by other members' addition. This interaction is known as *peer effect* or *externality* in matching theory [2]. The utility function U_i^m for m_i is measured by the transmission rate over each CR. And the utility function U_j^h for h_j is measured by both the transmission rate and social centrality. This setting can guarantee that the optimization problem in user pairing keeps the same as problem (10). Based on the above discussion, the many-to-one matching with peer effect is defined as

Definition 1: The many-to-one matching Ω_t with peer effect is defined as a function mapping from $CH(t) \cup CM(t) \cup \emptyset$ into $CH(t) \cup CM(t) \cup \emptyset$ such that for $m_i \in CM(t)$ and $h_j \in CH(t)$:

- 1) $|\Omega_t(m_i)| \leq 1, \forall m_i \in CM(t)$ and $\Omega_t(m_i) = \emptyset$, if $\Omega_t(m_i) \notin CH(t)$;
- 2) $|\Omega_t(h_j)| \leq q_j, \forall h_j \in CH(t)$, and $\Omega_t(h_j) = \emptyset$, if $\Omega_t(h_j) \notin CM(t)$;
- 3) $\Omega_t(m_i) = h_j$, only if $h_j \in \Omega_t(m_i)$.

We write $(h_j, \Omega_t) \succ_{m_i} (h_{j'}, \Omega'_t)$ to indicate that m_i prefers h_j in Ω_t to $h_{j'}$ in Ω'_t , when $U_m^i(h_j, \Omega_t) > U_m^i(h_{j'}, \Omega'_t)$. It is obvious that the utility function is not only influenced by its matching partner, but also by other users reusing the same spectrum.

Definition 2 ([2]): As a subset of $CM(t)$, the rotation-swap blocking group $\mathcal{BG} = \{m_1, \dots, m_i, \dots, m_l\}$, $1 \leq l \leq N_m(t)$ satisfies the following condition: for each r_i , $1 \leq i \leq l$ ($i-1 = l$ when $i=1$), $\Omega_t(r_{i-1}) \succeq_{r_i} \Omega_t(r_i)$.

The rotation-swap blocking group reorganizes the swap intention of users in a circular manner. In this way, not only the direct blocking relation but also the undirected relation can be satisfied. In short, the direct blocking relation is "You prefer my partner while I prefer your partner". The undirected blocking relation is "I prefer your partner, you prefer his partner, and finally he prefers my partner". In addition, the swap-operation should ensure that the utilities of users within rotation-swap blocking group increase while the utilities of the outside users are at least not worse off. Finally, the definition of rotation-swap stability is given by

Definition 3: A matching Ω_t is called rotation-exchange stable matching, if there exists no rotation-swap blocking group in Ω_t .

In this paper, we invoke the rotation-swap matching algorithm in [2] to obtain the stable cluster formation structure. Since cluster formation is not our main contribution, we only provide a sketch of this algorithm, and interested readers can refer to [2] for more details.

B. Dynamic Spectrum Sharing

Now, we will solve the dynamic spectrum sharing problem from the perspective of graph theory. We first depict the cumulative interference with the framework of hypergraph. In this stage, we mainly focus on the spectrum sharing among different clusters. For ease of analysis, we abstract the

clusters as nodes, and their mutual interference relationships as hyperedge. In each time slot t , the cluster set can be denoted as $\{1, \dots, C_j, \dots, C_{n(t)}\}$.

Definition 4 ([31], [40]): A hypergraph $G_H(t) = (V(t), E(t))$ on a finite vertex set $V(t) = \{v_1, \dots, v_{n(t)}\}$ is a family $E(t) = \{e_1, \dots, e_{m(t)}\}$ of subsets of $V(t)$, satisfying: 1) $e_i \neq \emptyset, \forall e_j \in E$, and 2) $\bigcup_{e_j \in E(t)} e_j = V(t)$, and each element e_j of $E(t)$ is termed as hyperedge.

Hypergraph is a generalized graph where each hyperedge contains any subset of vertex set. Due to the insertion/deletion of vertexes/edges, the graph structure is dynamically changed, which intrinsically caused by the dynamic network topology. Given $V(t)$ and $E(t)$, the hypergraph can be expressed by a $n(t) \times m(t)$ incidence matrix. According to [40], the incidence matrix $X = [x_{j,k}]_{n(t) \times m(t)}$ specified from a hypergraph $G_H(t) = \{V(t), E(t)\}$ is a Boolean matrix, where $x_{ij} = 1$ or 0 can determine whether or not v_j belongs to e_k . The hypergraph has been widely used to model the cumulative interference effect caused by multiple weak interferers, so that users can avoid sharing spectrum with other users in the same hyperedge.

For the construction of hypergraph, the authors in [31] considered that the hyperedge is constructed only if the wanted signal ratio to the interference is below a threshold. Different from the scenario in [31] where the each spectrum sharing cluster only has one cluster member for D2D communication, we should focus on the cluster member with the worst ratio for multicast system. For cluster C_j , if the minimum wanted signal to the cumulative social untrusted interference ratio falls below a threshold η , the C_j as well as all the interferers form a hyperedge, i.e.,

$$\min \frac{P_j g_{i,j}(t) \beta_{j,l,t}}{\sum_{j' \in D_l \setminus C_j} P_{j'} h_{j',i}(t)} < \eta, \forall m_i \in CM(t), \quad (12)$$

where D_l is a set of devices sharing channel l , $\beta_{j,l,t}$ is the channel allocation binary variable, and $\beta_{j,l,t} = 1$ or 0 determines whether or not channel l is allocated to cluster C_j in time slot t .

To facilitate understanding, we depict the hypergraph structures in two consecutive time slots in Fig. 2, based on the dynamic network topology in Fig. 1. In the initial time slot, three UAVs disseminate emergency information to surviving devices in a multicast manner and each of them acts as a cluster head. Each active cluster is abstracted as an active node. Based on E.q. (12), their mutual interferences exceed a limit and we construct a hyperedge $E1$ to depict the interference relationships, which indicates that they cannot share the same spectrum. In the next time slot, those surviving devices receiving the information will act as a cluster head, and thus some new active nodes are inserted. Besides, the hyperedge will also change due to the dynamic interference relationships caused by dynamic cluster formation, e.g., hyperedge $E1$ connects three nodes in time slot t and two nodes in time slot $t+1$. Similar to [31], we assume that one hypergraph can cover at most three nodes to achieve a trade-off between interference management level and computational complexity.

Based on above descriptions and assumptions, we first introduce a binary variable y_{ij} to represent cluster j 's interference state in hyperedge e_j , which is defined as follows:

$$y_{j,k} = \begin{cases} 1, & \text{if } x_{j,k} = 1; \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Definition 5 (Cumulative interference): Let $CI_{j,l}(t)$ denote cluster j 's received cumulate social interference when selecting channel l which is given by

$$CI_{j,l}(t) = \sum_{e_k \in E(t)} \phi_j(e_k) y_{j,k}, \quad (14)$$

where the indicator function $\phi_j(e_k)$ is defined as follows:

$$\phi_j(e_k) = \begin{cases} 1, & \text{if } \beta_{j,l,t} = 1, \beta_{j,l,t} = \beta_{j',l,t}; \forall v'_j \in e_k; \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

Let $C_{j,l}(t)$ denote the normalized information diffusion capacity of cluster j , which is given by

$$IC_{j,l}(t) = \begin{cases} 1, & \text{if } CI_{j,l}(t) = 0; \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where $IC_{j,l}(t) = 0$ implies that cluster j cannot disseminate information successfully. Given the above assumption, the optimization problem is formulated as

$$\mathbf{P3:} \quad \max_{\beta(t)} \sum_{t=t'}^{t'+1} \sum_{j=1}^{N_h(t)} \beta_{j,l,t} IC_{j,l}(t), \quad (17a)$$

$$s.t. \quad C1: \sum_{t=t'}^{t'+1} \sum_{l=1}^L \beta_{j,l,t} \leq 1; \quad (17b)$$

where (17b) ensures that one cluster can only be allocated at most one channel in one time slot. The problem (17) is a NP-hard combinatorial optimization problem [31]. Graph coloring is a powerful and low complexity method for this kind of optimization problem. Herein, we formulate the channel resource as L different colors and allocate each node (cluster) a color to guarantee that the nodes in the same hyperedge cannot be allocated the same color. The authors in [31] introduced a greedy hypergraph coloring algorithm. However, this algorithm needs to recolor all nodes when the hypergraph structure is changed, which causes a higher complexity consumption and thus cannot be applied to the emergency scenarios. In the next section, we will introduce a dynamic hypergraph coloring approach, which can achieve high efficiency and low complexity consumption when the hypergraph structure is updated dynamically.

IV. DYNAMIC HYPERGRAPH COLORING APPROACH

In this section, we design a dynamic hypergraph coloring approach, which can achieve good coloring consistency and low complexity consumption. As mentioned in Section I, the coloring consistency can save the channel switch cost caused by the unnecessary channel reallocation. The main idea of the approach is to simplify the original hypergraph to an oriented graph and invoke a modified color-propagation

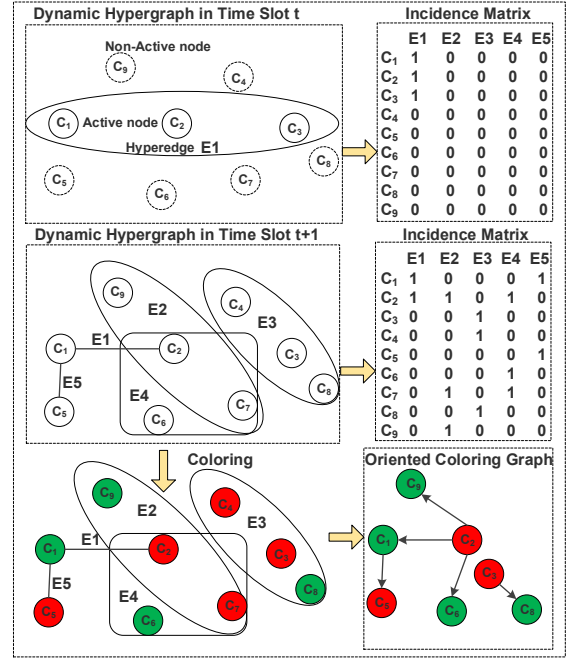


Fig. 2: Dynamic hypergraph structure in two consecutive time slots and its corresponding OCG.

algorithm based on [26] upon the simplified graph structure. The color-propagation can iteratively recolor a vertex and inform its in-neighbors. If this vertex changes its color, the informed vertices will be correspondingly recolored. Besides, we design a priority rule to ensure that one vertex is recolored once by only visiting the 2-hop neighbors of each vertex. According to this characteristic, we only need to maintain the color information of each vertex's in-neighbors, which can be used to determine whether any vertex's color will change in constant time before recomputing the color. In this way, we only need to explore the updated vertices and their neighbors when the hypergraph is dynamically updated. We first provide some preliminaries for hypergraph coloring.

A. Preliminaries

For a hypergraph $G_H(t) = (V(t), E(t))$, the number of vertices and hyperedges are $n(t)$ and $m(t)$. Every vertex v_i is assigned a unique ID $id(v_i, G_H(t))$. The set of neighbors of v_i is $nb(v_i, G_H(t))$, where any vertex is connected with v_i by at least one hyperedge. The monodegree of v_i represented by $mdeg(v_i, G_H(t))$ is the number of hyperedges involving v_i . For ease of expose, we omit $G_H(t)$ in the notations if the context is self-evident.

Definition 6 (Dynamic Hypergraph Coloring): For a dynamic hypergraph $G_H(t) = (V(t), E(t))$ where the structure changes over time, the hypergraph coloring of $G_H(t)$ can be defined as $f: V(t) \rightarrow \mathcal{C}$ from the vertex set $V(t)$ to the color set \mathcal{C} , such that there always exists at least two vertices in a hyperedge, their color is different in each time slot.

Given a hypergraph $G_H(t)$ and its coloring function f , $|f(G_H(t))|$ denotes color numbers used in f . Given a vertex v_i , $v_i.color = f(v_i)$ represents the assignment color of v_i .

Definition 7 (L -colorable): A Dynamic hypergraph $G_H(t)$ is L -colorable if $|f(G_H(t))| = L, \forall t \in \mathcal{T}$, and the chromatic number is defined as the smallest integer L .

The optimization object is to dynamic coloring the hypergraph with the given chromatic number L to maintain the cumulative interference level. The general approach for most existing static hypergraph/graph coloring algorithms is **Global** [41]. The general idea of **Global** is summarized as follows: All vertexes are first sorted in decreasing order according to monodegrees (increasing order of their IDs when their monodegrees are same). Next, each vertex is traversed in the sorted order and assigned the minimum available color that not assigned to its colored neighbors. In this paper, we first use it to coloring the initial hypergraph $G_H(1)$ and focus on the recoloring approach when the hypergraph is updated. We transfer the property of **Global** in conventional graph to dynamic hypergraph, which refers to finding a coloring f such that the color of vertex satisfies the following property:

Definition 8 (Global Property): For a hypergraph $G_H(t)$ and its coloring function f , v_i .color satisfies v_i .color = $\min\{c|c \in \mathcal{C} \setminus C(v_i)\}$, in which $C(v_i)$ records the color of one of v_i neighbors in each of the hyperedges e_j involving v_i . The recorded neighbor v_k having the maximum monodegree in e_j satisfies: 1) $mdeg(v_i) \leq mdeg(v_k)$, or 2) $mdeg(v_i) = mdeg(v_k)$, $id(v_k) < id(v_i)$.

Definition 8 indicates that one vertex only needs to focus on the color of at most one neighbor vertex for each hyperedge based on Definition 6, and the global property gives the priority to the vertexes with higher monodegree and lower ID ranking. The object of dynamic coloring is to maintain the global property when the hypergraph is updated. A naive approach is to iteratively invoke **Global** for every update, which is impractical for emergency scenarios. Intuitively, we can only recolor those vertexes that violates the global property. However, the color change of one vertex will trigger the violation of a series of vertexes. Inspired by [26], we transform the original hypergraph to a oriented coloring graph and focus on the chain reaction triggered by recoloring. We construct the oriented coloring graph according to the following order:

Definition 9: For a dynamic hypergraph $G_H(t)$ and two vertexes v_i and v_k , we define $v_i \triangleleft v_k$ if 1) $mdeg(v_i) \leq mdeg(v_k)$, or 2) $mdeg(v_i) = mdeg(v_k)$, $id(v_k) < id(v_i)$.

For two vertexes v_i and v_k , we say v_k dominates v_i if $v_i \triangleleft v_k$. Based on the definition of oriented coloring graph (OCG) converted from the conventional graph [26], we further give the definition of OCG for hypergraph:

Definition 10 (OCG): For a dynamic hypergraph $G_H(t) = (V(t), E(t))$, the OCG $G_o(t) = (V^*(t), E^*(t))$ of $G_H(t)$ is a directed acyclic graph, in which for two vertexes v_i and v_k , if $v_i \triangleleft v_k$, we depict a directed edge from v_k to v_i in $G_o(t)$, termed as $\langle v_k, v_i \rangle$.

Note that the aim of constructing OCG is to recognize the cause of the color change much clearer. Based on Definition 8, one vertex only focuses on one vertex at most in each hyperedge passing it. Globally, for each hyperedge, we can only focus on the vertex with highest monodegree and the vertex with lowest monodegree dynamically. When the vertex with lowest or highest monodegree has more than

one candidate, we randomly pick one for lowest monodegree and the other for highest monodegree, with different colors. Because the global property is satisfied once the colors of these two vertexes are different, we can further reduce the vertex number and edge number for constructing OCG, which is illustrated in Fig. 2. For hyperedge $E1$ and $E5$, since $mdeg(C_2) > mdeg(C_1) > mdeg(C_5)$, we depict a directed edge from C_2 to C_1 and then C_1 to C_5 . For hyperedge $E2$, $E3$, and $E4$, we only focus on the vertex with highest monodegree and the vertex with lowest monodegree, i.e., C_2 and C_9 in $E2$, C_3 and C_8 in $E3$, C_2 and C_8 in $E4$.

Similar to the definition in [26], for a directed edge $\langle v_k, v_i \rangle$, v_k is defined as the in-neighbor of v_i while v_i is the out-neighbor of v_k . We further use $nb^-(v_k)$ and $nb^+(v_k)$ to represent the in-neighbor set and out-neighbor set, respectively. In this regard, the in-degree and out-degree of v_k are denoted as $deg^-(v_k) = |nb^-(v_k)|$ and $deg^+(v_k) = |nb^+(v_k)|$, respectively. Since the essence of vertexes insertion/deletion is the related edges insertion/deletion, we first introduce the edge insertion/deletion and then turn into the vertex insertion/deletion. Without loss of generality, we use $G_o(t) + \langle v_k, v_i \rangle$ or $G_o(t) - \langle v_k, v_i \rangle$ to represent the updated OCG when the hypergraph is updated. Based on Definition 6 and 8, we can define the corresponding property of OCG.

Definition 11 (OCG Coloring [26]): For an OCG $G_o(t) = (V^*(t), E^*(t))$, the OCG coloring is a coloring function f_o , where the colors of any two incident vertexes are different, i.e., $\forall \langle v_k, v_i \rangle \in E^*(t)$, v_i .color \neq v_k .color.

Definition 12 (Oriented Global Property [26]): For an OCG $G_o(t) = (V^*(t), E^*(t))$ and its coloring f_o , the color of v_i satisfies oriented global property if v_i .color = $\min\{c|c \in \mathcal{C} \setminus C_o(v_i)\}$, where $C_o(v_i) = \bigcup_{v_k \in nb^-(v_i)} v_k$.color.

This property means that each vertex should choose the color with the smallest number from those colors not assigned to its in-neighbors. For simplicity, we use $\Sigma(G_o(t))$ to denote the global oriented coloring of $G_o(t)$, and $\chi(G_o(t))$ to denote the oriented global property. For $G_o(t)$, once $\langle v_k, v_i \rangle$ is inserted or deleted, the $\Sigma(G_o(t) \pm \langle v_k, v_i \rangle)$ has to be recomputed by recoloring those vertexes violating $\chi(G_o(t))$. In line of Definition 12, the color of those vertexes that still dominate v_i or v_k in $\Sigma(G_o(t))$ and $\Sigma(G_o(t) \pm \langle v_k, v_i \rangle)$ remains unchanged. But the colors of other vertexes in $\Sigma(G_o(t))$ may violate $\chi(G_o(t) \pm \langle v_k, v_i \rangle)$ and those vertexes need to be recolored to maintain $\chi(G_o(t) \pm \langle v_k, v_i \rangle)$, which can be denoted by the following equation [26]:

$$f_o^{new} \leftarrow \min\{c|c \notin \bigcup_{x \in nb^-(w)} f_o^{old}(x)\}. \quad (18)$$

where f_o^{new} and f_o^{old} denote the coloring functions before and after the recoloring of w respectively. E.q. (18) means that the recoloring strategy should maintain the oriented global property.

Lemma 1: Given an OCG $G_o(t)$ and its global oriented coloring $\Sigma(G_o(t))$, considering the insertion/deletion of edge $\langle v_k, v_i \rangle$, the coloring f_o is $\Sigma(G_o(t) \pm \langle v_k, v_i \rangle)$ when E.q. (18) converges for all $w \in G_o(t)$.

Proof: Lemma 1 can be easily proved by contradiction. Assumed the coloring is not $\Sigma(G_o(t) \pm \langle v_k, v_i \rangle)$, there must exist

a vertex that violates $\chi(G_o(t) \pm \langle v_k, v_i \rangle)$, which contradicts the condition that E.q. (18) converges.

B. Color Propagation based Dynamic Coloring

Based on Lemma 1, we can iteratively recolor those vertexes that violates $\chi(G_o(t) \pm \langle v_k, v_i \rangle)$ to satisfy E.q. (18). Next, we introduce the color propagation (CP) algorithm according to [26], which consists of three stages: 1) Color Collection: For an OCG $G_o(t)$ and a vertex v_i , the CP collects the color information $C^{in}(v_i)$ of v_i 's in-neighbors, i.e., $C^{in}(v_i) = \cup_{v_k \in nb^-(v_i)} v_k.color$; 2) Assign Color: it assigns the smallest color in $\mathcal{C} \setminus C^{in}(v_i)$ to v_i , and then returns a boolean indicator $\delta = 1$ or 0 to determine whether or not $v_i.color$ changes. 3) Inform Color: it informs the out-neighbor set of v_i to recolor if $\delta = 1$. The general idea of CP is to perform the above mentioned three stages sequentially. Given the insertion/deletion of $\langle v_k, v_i \rangle$, we have to select the seed vertexes in addition to v_k and v_i , since the degree as well as the dominant relationship of their neighbors will change correspondingly. Those neighbor vertexes may violate $\chi(G_o(t) \pm \langle v_k, v_i \rangle)$, since their dominant relationship with respect to v_k and v_i are correspondingly updated. Therefore, the seed vertex selection rule is given as follows:

Lemma 2: For the edge insertion case $G_o(t) + \langle v_k, v_i \rangle$, the seed vertex set should be $\{v_i \cup v_k \cup I_{v_i} \cup I_{v_k}\}$, where $I_{v_i} = nb^-(v_i, G_o(t)) \cap nb^+(v_i, G_o(t) + \langle v_k, v_i \rangle)$. For the edge deletion case $G_o(t) - \langle v_k, v_i \rangle$, the seed vertex set should be $v_i \cup v_k \cup D_{v_i} \cup D_{v_k}$, where $D_{v_i} = nb^+(v_i, G_o(t)) \cap nb^-(v_i, G_o(t) - \langle v_k, v_i \rangle)$.

Proof: Please see Appendix A.

The color propagation based dynamic coloring (CPDC) algorithm is summarized in Alg. 1. The general idea is to recolor the vertexes in queue σ . Given an inserted/deleted edge $\langle u, v \rangle$, it invokes OCG-Update to check whether or not the dominant relationships are changed by the insertion/deletion of $\langle u, v \rangle$, and then reconstruct the OCG based on the newly updated dominant relationships. The OCG-update returns the seed vertex set \mathcal{S} based on Lemma 2 and pushes these seed vertexes into queue σ . The CP procedure (line 5-25) is iteratively invoked to recolor the affected vertexes and maintains the vertexes to be recolored in σ until σ is empty. Note that we slightly modify the color propagation strategy proposed by [26] in graph construction and assign color stages to adapt to the property of hypergraph.

Theorem 1: For an OCG $G_o(t)$ and its global oriented coloring $\Sigma(G_o(t))$, the coloring obtained by Alg. 1 is $\Sigma(G_o(t) \pm \langle u, v \rangle)$ for $G_o(t) \pm \langle u, v \rangle$.

Proof: Please see Appendix B.

It is worth mentioning that the complexity of CPDC is not bounded, since it depends on the number of vertexes in queue σ . In simulation process, we find that one vertex may be pushed in σ more than once, due to the fact that the recoloring of its in-neighbors will push this vertex into σ repeatedly in IC procedure. This domino effect leads the unbounded vertex number of σ . In this subsection, we introduce the Dynamic In-Neighbor Color Index (DINC-Index) in [26] to bound the vertex number pushed into σ and further reduce the complexity

Algorithm 1 Color Propagation based Dynamic Coloring (CPDC) Algorithm

```

1: When the network topology changes, cluster heads exchange information with each other and dynamically update the OCG structure. For any two nodes  $u$  and  $v$  whose interference relationship changes, invoking CPDC.
2: Procedure CPDC ( $G_o(t)$ ,  $\langle u, v \rangle$ )
3: Queue  $\sigma \leftarrow \emptyset$ ;
4:  $\mathcal{S} \leftarrow$  OCG-Update ( $G_o(t)$ ,  $\langle u, v \rangle$ )
5: for each  $w \in \mathcal{S}$  do
6:   Push  $w$  into  $\sigma$ ;
7:   while  $\sigma \neq \emptyset$  do
8:      $u \leftarrow \sigma.pop()$ ;  $C^{in}(u) \leftarrow \emptyset$ ;
9:     for each  $v \in nb^-(u)$  do
10:       $C^{in}(u) \leftarrow C^{in}(u) \cup v.color$ ;
11:   end for
12:    $C \leftarrow \{0, \dots, mdeg(u)\}$ ;  $c_{new} \leftarrow \min\{c | c \in C \setminus C^{in}(u)\}$ ;
13:   if  $c_{new} \neq u.color$  then
14:      $u.color \leftarrow c_{new}$ ;  $\delta = 1$ ;
15:   else
16:      $\delta = 0$ ;
17:   end if
18:   if  $\delta = 1$  then
19:     for each  $v \in nb^+(u)$  do
20:       if  $v \notin \sigma$  then
21:         Push  $v$  into  $\sigma$ ;
22:       end if
23:     end for
24:   end if
25: end while
26: end for
27: Procedure OCG-Update ( $G_o(t)$ ,  $\langle u, v \rangle$ )
28:  $\mathcal{S} \leftarrow \emptyset$ ;  $\mathcal{S} \leftarrow \mathcal{S} \cup v \cup u$ ;
29: if  $\langle u, v \rangle$  is inserted then
30:    $G_o(t) \leftarrow G_o(t) + \langle u, v \rangle$ ;
31:   for each  $u' \in nb^-(u)$ , and  $v' \in nb^-(v)$  do
32:     if  $u' \triangleleft u$  or  $v' \triangleleft v$  then
33:       replace  $\langle u', u \rangle$  ( $\langle v', v \rangle$ ) by  $\langle u, u' \rangle$  ( $\langle v, v' \rangle$ ),  $\mathcal{S} \leftarrow \mathcal{S} \cup u' (v')$ .
34:     end if
35:   end for
36: else
37:    $G_o(t) \leftarrow G_o(t) - \langle u, v \rangle$ ;
38:   for each  $u' \in nb^+(u)$ , and  $v' \in nb^+(v)$  do
39:     if  $u \triangleleft u'$  or  $v \triangleleft v'$  then
40:       replace  $\langle u, u' \rangle$  ( $\langle v, v' \rangle$ ) by  $\langle u', u \rangle$  ( $\langle v', v \rangle$ ),  $\mathcal{S} \leftarrow \mathcal{S} \cup u' (v')$ .
41:     end if
42:   end for
43: end if
44: Return  $\mathcal{S}$ ;

```

in three steps based on the structure of OCG graph. Next we will introduce how to migrate the DINC-Index in [26] to Alg.1.

In the first step, we will handle the domino effect caused by the case where a vertex is recolored before its in-neighbors. The general idea is that vertex can be recolored until all its in-neighbors have been recolored, which indicates that we have to recolor the vertex pushed into σ in an appropriate order. Since OCG belongs to the directed acyclic graph [26], we can organize the recoloring order along the topological order of vertexes. Herein, we give each vertex a unique priority in σ . Considering that the direction of edges in $G_o(t)$ is constructed based on the dominant relationships, we can design the priority rule as: for u and v , if $u \triangleleft v$, we can say v has a higher priority than u . We can replace the queue σ in Alg.1 with priority queue σ^* based on the priority rule.

Theorem 2 ([26]): For an OCG $G_o(t)$, considering the case $G_o(t) \pm \langle u, v \rangle$, let Λ denote those vertexes containing u and v , whose colors are changed in $\Sigma(G_o(t) \pm \langle u, v \rangle)$ compared with $\Sigma(G_o(t))$, and the number of vertexes pushed into σ with priority rule can be bounded by $n_\Lambda = |\cup_{u \in \Lambda} nb^+(u) \cup nb^-(u) \cup \Lambda|$.

Proof: Please refer to [26] for more details.

In the second step, we further reduce the n_Λ by color pre-computation. According to the priority rule, the vertexes pushed into σ can be classified into two types: 1) the vertexes in Λ , whose colors are changed in $\Sigma(G_o(t) \pm \langle u, v \rangle)$ compared with $\Sigma(G_o(t))$; 2) the out-neighbors of the first type of vertexes; and those vertexes not of the first type. Considering that each vertex w in σ^* is processed in CP procedure, the in-neighbors of the second type of vertex w need to be explored by the CP procedure since it is not obvious whether or not w belongs to the second type. Since a vertex belonging to the second type is a neighbor of a vertex belonging to the first type, we only need to explore the 2-hop neighbors of some vertexes belonging to the first type and avoid exploring the neighbors of belonging to the second type vertexes. We introduce a dynamic in-neighbor color index (DINC-Index) to record the coloring information based on [26]. The DINC-Index \mathcal{DI} consists of two components: 1) Color Counts $C_{cnt}^u(c)$ records u 's in-neighbors whose color is c for each $u \in V(t)$, and $c \leq deg^-(u)$; 2) Recolor Candidates R_{can}^u records the colors not assigned to any in-neighbors of u , and these colors are smaller than $u.color$.

The use of DINC-Index are two fold. Firstly, for any vertex u , its color satisfying $u.color \leq deg^-(u)$. If all the colors of in-neighbors of u that is smaller than the $deg^-(u)$ is known, $u.color$ can be determined. Hence, it is reasonable to record the color counts for $c \leq deg^-(u)$ in $C_{cnt}^u(c)$. Correspondingly, we can get $R_{can}^u = \{c | c < u.color, C_{cnt}^u(c) = 0\}$. In line of the expression of R_{can}^u , a global oriented coloring $\Sigma(G_o(t))$ for $G_o(t)$ is satisfied only if $R_{can}^u = \emptyset$. Hence, for $\Sigma(G_o(t)) \pm \langle u, v \rangle$, the color of u is changed in CP procedure if $R_{can}^u \neq \emptyset$ or $C_{cnt}^u(c) \neq 0$. For the former condition, $u.color = \min\{c | c \in R_{can}^u\}$. For the latter condition, $u.color = \min\{c | c \in C, C_{cnt}^u(c) = 0\}$. In this way, we can know whether or not u will change its color in $\mathcal{O}(1)$, and thus avoid exploring the neighbors of those vertexes belonging to the second type. Since the DINC-Index maintenance algorithm is described in detail in [26], we briefly describe two important

parts. In the DINC-Index maintenance algorithm, the Color-Update is to maintain the R_{can}^u and $C_{cnt}^u(c)$ by inserting or deleting a color c for u . For the case of color insertion, only the case $c \leq deg^-(u)$ and is considered and $C_{cnt}^u(c)$ is increased by 1. With the guarantee of $C_{cnt}^u(c) \neq 0$, c is deleted from R_{can}^u according to the expression of R_{can}^u . The color deletion procedure is similar to insertion, and its complexity is $\mathcal{O}(1)$. To main the DINC-Index, $u.color$ is inserted for u' and $u'.color$ is deleted from DIM for each $\langle u, v \rangle$ to be reversed. Besides, $u.color$ is inserted for v along with the insertion of $\langle u, v \rangle$. For a special case that only the condition that $c \leq deg^-(v)$ for v is considered, $deg^-(v)$ is increased by 1 along with the insertion of $\langle u, v \rangle$, and its in-neighbors whose color is $deg^-(v)$ for v should be added. The general idea for the deletion case is similar. Only the special condition that $deg^-(v)$ is decreased by 1 along with the edge deletion, and $c \leq deg^-(v)$ in $C_{cnt}^v(c)$ should be considered, which can be adjusted by simply setting $C_{cnt}^v(deg^-(v) + 1) = 0$.

In the third step, we further investigate how to find those out-neighbors for u , which are not affected by the recoloring of u . We can further reduce the queue size by avoiding pushing those vertexes into σ . Herein, $u.old$ and $u.now$ represent the colors of u before and after the CP procedure. For simplicity, we summarize three cases for u and its out-neighbor v . 1) If $u.now = v.now$, v needs to be recolored; 2) if $u.now \neq v.now$ and $u.old < v.old$, v needs to be recolored. Because their current color condition may be $u.now > v.now$, and v can use $u.old$; 3) if $u.now \neq v.now$ and $u.old > v.old$, $v.now$ remains unchanged. Because its color is not affected by $u.now$. Based on the above discussed three steps, the improved CPDC can be summarized in Alg. 2.

The main framework of Alg.2 is similar to Alg.1, the difference is that we integrate the above discussed three steps into CP. Note that Alg. 2 can also handle the vertex insertion/deletion case. In case 1, given an inserted vertex u , u is firstly assigned with color 0, and the edges involving u are inserted into the DINC-Index. The difference is that we return \mathcal{S} until the edge insertion process is finished. Finally, we push the vertexes into σ^* and perform ICPDC iteratively.

Theorem 3: The complexity of Alg.3 is $\mathcal{O}(n_\Lambda \log(n_\Lambda))$.

Proof: Please see Appendix C.

Finally, we summarize the dynamic cluster formation and spectrum sharing (DCFSS) algorithm in Alg. 3.

Remark 1: Considering that the rotation-swap matching algorithm can achieve the stable cluster structure, and Alg.3 can achieve the global coloring property, the convergence of DCFSS can be easily proved since the value of optimization object in problem (10), (11) and (15) will improved or remain unchanged after each iteration. Due to the limited spectrum resource and device number, DCFSS will final converge to the stable state. The sum complexity is $\mathcal{O}(K_1(N(t) \log(N(t)) + N(t)^3 + K_2 C_{N(t)}^2 N(t)^2))$ in the worst case, where K_1 is the iteration number of DCFSS and K_2 is the iteration number of the rotation-swap matching algorithm in [2]. In fact, the iteration number is relatively lower than the device number, which can be verified by the following simulation results.

Algorithm 2 Improved CPDC (ICPDC) Algorithm

```

1: When the network topology changes, cluster heads exchange information with each other and dynamically update the OCG structure and the available color set. Priority Queue  $\sigma^* \leftarrow \emptyset$ .
2: Case 1: For a newly joined/left user (an inserted/deleted vertex)  $u$ ,  $u$  is firstly assigned with color 0. The edges  $e$  connecting  $u$  and another  $v$  are inserted into the  $\mathcal{DI}$ .  $\mathcal{S} \leftarrow \text{Index-Maintenance}(\mathcal{DI}, G_o(t), \langle u, v \rangle)$  in [26],  $\sigma^*.push(u)$ .
3: Case 2: When the interference relationship between  $u$  and  $v$  changes,  $\mathcal{S} \leftarrow \text{Index-Maintenance}(\mathcal{DI}, G_o(t), \langle u, v \rangle)$  in [26].
4: Procedure ICPDC( $G_o(t), \langle u, v \rangle, \mathcal{DI}$ )
5: for each  $w \in \mathcal{S}$  do
6:   Push  $w$  into  $\sigma^*$ ;
7:   while  $\sigma \neq \emptyset$  do
8:      $u \leftarrow \sigma^*.pop()$ ;
9:     if  $R_{can}^u \neq \emptyset$  then
10:       $c_{new} \leftarrow \min\{c \in R_{can}^u\}$ ;
11:    end if
12:    if  $C_{cnt}^u(u.color) \neq 0$  then
13:       $c_{new} \leftarrow \min\{c \in \mathcal{C}, C_{cnt}^u(c) = 0\}$ ;
14:    end if
15:    if  $c_{new} \neq \emptyset$  then
16:       $c_{old} \leftarrow u.color$ ;
17:      for each  $v \in nb^+(u)$  do
18:        Color-Update( $\mathcal{DI}, v, u.color$ );
19:        Color-Update( $\mathcal{DI}, u, c_{new}$ );
20:      end for
21:       $u.color \leftarrow c_{new}$ ;  $R_{can}^u \leftarrow \emptyset$ ;
22:      for each  $v \in nb^+(u)$  do
23:        if  $v \notin \sigma^*$  and  $c_{old} < v.color$  or  $u.color = v.color$  then
24:           $\sigma^*.push(v)$ ;
25:        end if
26:      end for
27:    end if
28:  end while
29: end for

```

Algorithm 3 Dynamic Cluster Formation and Spectrum Sharing (DCFSS) Algorithm

```

1: Phase 0: Proceed to the next time slot  $t$ ;
2: Each UAV determine whether to recharge based on the remaining battery capacity. Reserved UAVs and newly arrived devices update preference profile.
3: Loop in time slot  $t$ 
4: Phase 1 Cluster Formation: Both the UAVs and surviving devices self-organize the cluster structure based on the rotation-swap matching algorithm in [2].
5: Phase 2 Spectrum Sharing: Given the cluster structures, each cluster head update the channel allocation strategy based on the Alg.3.
6: Repeat Phase 1~ Phase 2 until the cluster structure and channel allocation strategy become stable in  $t$ .

```

V. SIMULATION RESULTS

In this section, we conduct simulations in different scenarios to demonstrate the validity of our theoretical analysis. We evaluate our proposed algorithm DCFSS in terms of convergence and trade-off between the average diffusion speed of emergency information and channel switching cost. Note that we use the number of cluster heads switching to another channel in two continuous slots, i.e., color consistency, to roughly represent the channel switching cost. Herein, the average diffusion speed represents the difference between the number of UAVs receiving the task during two consecutive time slots [42]. Similar to the setting of mobility in [42], we assume that terrestrial users move with different speeds (i.e., 0~5 meter/second) based on a random walk model. The main simulation parameters are listed in Table II. We set the parameters based on the existing works [9], [34], [43], which are applicable for UAV communication of most heterogeneous cellular networks. Generally, the parameters are set as follows unless being specified otherwise. Besides, the parameters of E^{hov} can be found in [9] and the energy threshold E_{th} is 200 KJ. The quota q for each multicast cluster is 5, and the hyperedge construction threshold for η is 20dB. In social domain, we consider the Erdos–Renyi social graph with adjustable social link probability as social graph model [44].

TABLE II: Main Simulation Parameters

Parameters	Value
Network area	$(1 \times 1 \times 1\text{km}^3)$
Initial number of UAVs and survivors	5, 50
Number of time slot and duration	100, 1s
Arrival rate	10 survivors/s
Air-time ratio	0.25 survivors/s
Number of available channels	5
Normalized bandwidth block	1MHz
Transmit powers of UAV and terrestrial device	5W, 0.1W
Path loss parameters for A2G and G2G links	2, 3
Power gain at $d_0 = 1$ m	-60dB
Noise power	-120dBm
Size of each data packet	1Mbits

Furthermore, we adopt different benchmark algorithms to objectively and comprehensively evaluate our proposed algorithm. Firstly, we compare DCFSS with two dynamic graph coloring algorithms: DC-Local [23], DC-Global [26], and one static hypergraph coloring algorithm: SHC [31]. Note that we do not compare DCFSS with SHC for the evaluation of diffusion speed, since they achieve the same coloring strategy. The difference is that SHC recomputes the coloring for all vertexes in each time slot and DCFSS only explores a small number of vertexes. To investigate the influence of social tie, we also consider the performance of our proposed approach without considering the social tie in cluster formation, termed as "Without Social Tie (WST)". Moreover, we provide the performance upper bound achieved by "Exhaustive Search(ES)".

To evaluate the convergence performance of DCFSS, we investigate the cumulative distribution function (CDF) versus the iteration numbers. In Fig. 3, we plot the CDF of iteration number asymptotically converging to the stable state. Meanwhile, we note that the needed iteration number of SCS grows

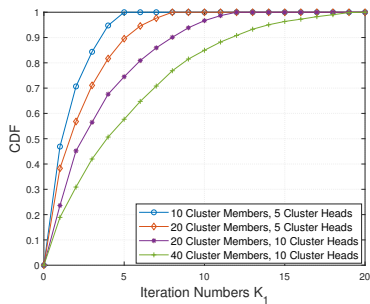


Fig. 3: Distribution of the iteration numbers in DCFSS.

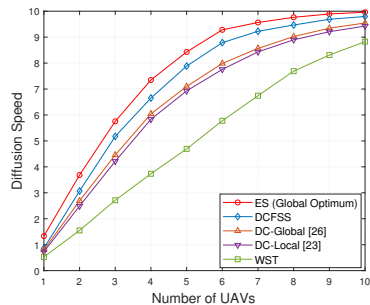


Fig. 4: The diffusion speed with varying number of UAVs.

with the cluster head and member numbers. It is because the number of nodes will affect the interplay between spectrum allocation and cluster formation. The probability of deviating to other better choice will increase in both spectrum allocation and cluster formation phases. In addition, Fig. 3 further reveals that the needed iteration number is relatively small with respect to the D2D user number and thus demonstrating the validity of the convergence analysis in Remark 1.

In Fig. 4, the emergency information diffusion speed is shown with varying number of UAVs. In this simulation, the social link probability is set to 0.6. It is obvious that the diffusion speed increases with the number of UAVs, because more UAVs can provide communication services for more terrestrial devices in each time slot. Note that DCFSS can achieve the faster diffusion speed than DC-Global and DC-Local. The reason is that the hypergraph can model the mutual interference relationships more accurately. Moreover, the benchmark WCT cannot utilize the social information in cluster formation, which decreases its performance. Fig. 5 compares the channel switching times attained by our proposed DCFSS with other benchmarks. With the increasing arrival rate of newcomers, the number of newly inserted vertex is correspondingly increased in graph coloring. Therefore, the channel switching times (recoloring times) will be increased. Note that some vertexes can be deleted during the procedure where hypergraph is transformed into OCG, and all vertexes have to be reserved during the procedure where traditional graph is transformed into OCG. And the SHC has to recompute the coloring strategy for all vertexes when the hypergraph is updated. This is the reason why DCFSS outperforms other benchmarks.

In Fig. 6 and Fig. 7, we investigate the influence of social

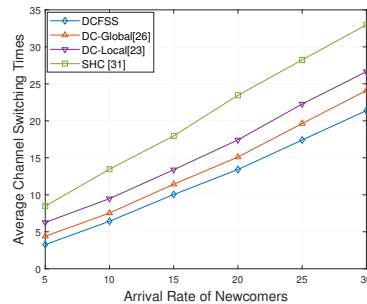


Fig. 5: The channel switching times versus the arrival rate.

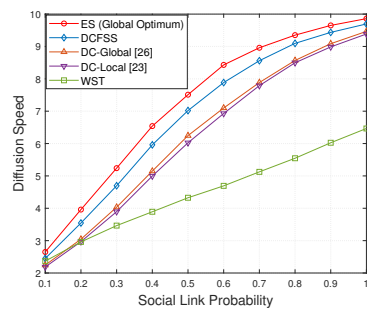


Fig. 6: The diffusion speed versus the social link probability.

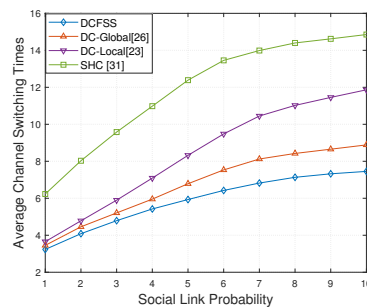


Fig. 7: The switching times versus the social link probability.

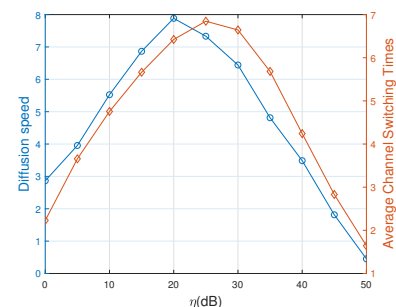


Fig. 8: The diffusion speed and switching times versus η .

tie on the performance of both the diffusion speed and channel switching times. Fig. 6 demonstrates that the information diffusion speed increases with the social link probability. It is because that the terrestrial device as cluster head can inform more newcomers if the probability of a social tie between them increases. Similarly, if more newcomers can be served, more clusters will be formed. As a result, more vertexes will be inserted into the OCG and thus the channel switching times (recoloring times) will be increased.

In Fig. 8, we investigate the impact of threshold η in hyperedge construction on the performance of diffusion speed and channel switching times achieved by DCFSS. It is obvious that both curves tend to increase first and then decrease. When the threshold is below 20dB for diffusion speed curve (20dB for channel switching times curve), more hyperedges will enable cluster heads to switch to the appropriate channel, and thus the channel switching times as well as the diffusion speed performances increase. In this stage, less hyperedges will reduce the diffusion speed performance and channel switching times, since more cluster heads will share the same channel without the constraint. When the threshold is above 20dB for diffusion speed curve (20dB for channel switching times curve), i.e., the number of hyperedges is sufficiently large, fewer cluster heads will be allocated to channels, and thus the diffusion speed performance and channel switching times decrease.

Through comparison, our proposed approach is verified to achieve a better trade-off among diffusion speed performance, channel switching cost and computational complexity compared with other benchmarks in dynamic emergency scenarios.

VI. CONCLUSION

In this paper, we have investigated the UAV-assisted emergency communications in SIoT and solved both the dynamic cluster formation and spectrum sharing problems in stochastic environments. For the cluster formation problem, we formulated it as a many-to-one matching game with externality and adopt the rotation-swap algorithm to maximize the expected number of users receiving alert messages in each time slot. For the dynamic spectrum sharing problem, we proposed a dynamic hypergraph coloring approach to model the cumulative interference, and maintain the mutual interference at a low level by exploring a small number of vertexes, when the graph is dynamically updated. Moreover, we proved some crucial properties including global stability, convergence, and complexity. Finally, simulation results demonstrated that our proposed approach can achieve a better trade-off among the information diffusion speed, channel switch cost, and complexity.

APPENDIX A PROOF OF LEMMA 2

We first prove the edge insertion case by contradiction. Assume that the seed vertex set is not inadequate, which indicates that there must exist a vertex $w \notin \{v_i \cup v_k \cup I_{v_i} \cup I_{v_k}\}$ and its color is changed. However, w is not informed in the second stage of CP. In line of E.q. (17), the vertexes in

$\{v_i \cup v_k \cup I_{v_i} \cup I_{v_k}\}$ trigger the color propagation since their in-neighbor are updated in $G_o(t) + \langle u, v \rangle$. The second stage of CP will not inform a vertex only if the colors of its in-neighbors remain unchanged. Since w is not informed, we can say that the colors of all its in-neighbors are remain unchanged before and after the edge insertion, which contradicts the oriented global property if the color of w changes. The edge deletion case can be proved in the similar way.

APPENDIX B PROOF OF THEOREM 1

For the insertion/deletion of edge $\langle u, v \rangle$, CPDC can adequately return the seed sets based on Lemma 2. Besides, the CP procedure iteratively recolors the vertex that violates $\chi(G_o(t) \pm \langle u, v \rangle)$ until the queue σ is empty. In line of Lemma 1, the global oriented coloring is $\Sigma(G_o(t) \pm \langle u, v \rangle)$ after the convergence of recoloring procedure.

APPENDIX C PROOF OF THEOREM 3

For the edge insertion case, ϖ is assumed to be the number of vertexes pushed into σ^* . The complexity of push/pop process in a priority queue is $\mathcal{O}(1)/\mathcal{O}(\log(\varpi))$ using Fibonacci heap, and the sum complexity is $\mathcal{O}(\varpi \log(\varpi))$ to process ϖ vertexes. As discussed above, we can determine whether or not a vertex will be recolored in $\mathcal{O}(1)$ by invoking Alg.2. Thus, the complexity of the first two stages can be bounded by $\mathcal{O}(\varpi)$. Since we have proved that ϖ is bounded by $n_\Lambda = |\cup_{u \in \Lambda} nb^+(u) \cup nb^-(u) \cup \Lambda|$ in Theorem 2, the complexity of Alg.2 is bounded by $\mathcal{O}(n_\Lambda \log(n_\Lambda))$. For vertex insertion/deletion, the complexity is bounded by $\mathcal{O}(n_v \log(n_v))$, where n_v is the vertex pushed into σ^* and bounded by the number of all vertexes $N(t)$. Note that the complexity of the static hypergraph coloring algorithm in [31] is $\mathcal{O}((L + N(t))^3)$, which is much higher than our proposed ICPDC. For the complexity of constructing the hypergraph, since we have assumed that one hypergraph can cover at most three nodes to achieve a trade-off between interference management level and computational complexity, the complexity is bounded by $\mathcal{O}(N(t)^3)$, which is acceptable in dynamic environments. Note that invoking the static hypergraph coloring algorithm in dynamic topology also needs the same complexity cost.

REFERENCES

- [1] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The Social Internet of Things (SIoT) - when social networks meet the Internet of Things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, pp. 3594–3608, Nov. 2012.
- [2] B. Wang, Y. Sun, S. Li, and Q. Cao, "Hierarchical matching with peer effect for low-latency and high-reliable caching in social IoT," *IEEE Internet of Things J.*, vol. 6, pp. 1193–1209, Feb. 2019.
- [3] A. Masaracchia, L. D. Nguyen, T. Q. Duong, and M. Nguyen, "An energy-efficient clustering and routing framework for disaster relief network," *IEEE Access*, vol. 7, pp. 56520–56532, Jan. 2019.
- [4] G. Araniti, A. Orsino, L. Militano, L. Wang, and A. Iera, "Context-aware information diffusion for alerting messages in 5G mobile social networks," *IEEE Internet Things J.*, vol. 4, pp. 427–436, Apr. 2017.
- [5] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On a novel adaptive UAV-mounted cloudlet-aided recommendation system for LBSNs," *IEEE Trans. Emerg. Top. Com.*, pp. 1–1(In Press), Jan. 2018.
- [6] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for UAV relay networks," *IEEE Commun. Letters*, vol. 22, pp. 161–164, Jan. 2018.

- [7] T. Q. Duong, L. D. Nguyen, and L. K. Nguyen, "Real-time deployment and resource allocation for distributed UAV systems in disaster relief," in *Proc. IEEE IWCMC*, (Tangier, Morocco), pp. 1–6, Jun. 2019 (accepted and appeared in IEEE Xplore Digital Library).
- [8] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Applications and open issues," in *Proc. IEEE ICNC*, (Kauai, Hawaii), pp. 1–5, Feb. 2016.
- [9] M. Y. Selim and A. E. Kamal, "Post-disaster 4G/5G network rehabilitation using drones: Solving battery and backhaul issues," in *Proc. IEEE GC Wkshps*, (Abu Dhabi, United Arab Emirates), pp. 1–6, Dec. 2018.
- [10] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K. Wong, "UAV-assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, pp. 45–51, Feb. 2019.
- [11] M. Liu, J. Yang, and G. Gui, "DSF-NOMA: UAV-assisted emergency communication technology in a heterogeneous internet of things," *IEEE Internet Things J.*, vol. 6, pp. 5508–5519, Jun. 2019.
- [12] X. Liu and N. Ansari, "Resource allocation in UAV-assisted M2M communications for disaster rescue," *IEEE Wireless Commun. Letters*, vol. 8, pp. 580–583, Apr. 2019.
- [13] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Trans. Wireless Commun.*, vol. 17, pp. 2233–2246, Apr. 2018.
- [14] Y. Wu, D. Wu, L. Yang, X. Shi, L. Ao, and Q. Fu, "Matching-coalition based cluster formation for D2D multicast content sharing," *IEEE Access*, vol. 7, pp. 73913–73928, Aug. 2019.
- [15] D. Zhai, R. Zhang, Y. Wang, H. Sun, L. Cai, and Z. Ding, "Joint user pairing, mode selection, and power control for D2D-capable cellular networks enhanced by nonorthogonal multiple access," *IEEE Internet Things J.*, vol. 6, pp. 8919–8932, Oct. 2019.
- [16] X. Deng, J. Luo, L. He, Q. Liu, X. Li, and L. Cai, "Cooperative channel allocation and scheduling in multi-interface wireless mesh networks," *Peer Peer Netw. Appl.*, vol. 12, pp. 1–12, Jan. 2019.
- [17] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, pp. 5141–5154, Dec. 2018.
- [18] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT," *IEEE Commun. Mag.*, vol. 56, pp. 80–86, Sep. 2018.
- [19] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anticoordination game based partially overlapping channels assignment in combined UAV and D2D-based networks," *IEEE Trans. Veh. Technol.*, vol. 67, pp. 1672–1683, Feb. 2018.
- [20] J. Chen, Y. Xu, Q. Wu, Y. Zhang, X. Chen, and N. Qi, "Interference-aware online distributed channel selection for multicenter FANET: A potential game approach," *IEEE Trans. Veh. Technol.*, vol. 68, pp. 3792–3804, Apr. 2019.
- [21] L. Feng, P. Zhao, F. Zhou, M. Yin, P. Yu, W. Li, and X. Qiu, "Resource allocation for 5G D2D multicast content sharing in social-aware cellular networks," *IEEE Commun. Mag.*, vol. 56, pp. 112–118, March 2018.
- [22] B. Balasundaram and S. Butenko, *Graph Domination, Coloring and Cliques in Telecommunications*, pp. 865–890. Boston, MA: Springer US, 2006.
- [23] D. Preuveneers and Y. Berbers, "ACODYGRA: an agent algorithm for coloring dynamic graphs," vol. 6, pp. 381–390, Jan. 2004.
- [24] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. ACM USENIXATC*, USENIXATC'10, (Boston, MA), pp. 21–21, USENIX Association, Jan. 2010.
- [25] J. Manweiler and R. Roy Choudhury, "Avoiding the rush hours: WiFi energy management via traffic isolation," *IEEE Trans. Mobile Comput.*, vol. 11, pp. 739–752, May. 2012.
- [26] L. Yuan, L. Qin, X. Lin, L. Chang, and W. Zhang, "Effective and efficient dynamic graph coloring," *Proc. VLDB Endow.*, vol. 11, pp. 338–351, Nov. 2017.
- [27] S. Ghosal and S. C. Ghosh, "A decentralize algorithm for perturbation minimization in 5G D2D communication," in *Proc. IEEE WONS*, (Wengen, Switzerland), pp. 72–78, Jan. 2019.
- [28] B. Bai, L. Wang, Z. Han, W. Chen, and T. Svensson, "Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework," *IEEE Wireless Commun.*, vol. 23, pp. 74–81, Aug. 2016.
- [29] H. Zhang, L. Song, Y. Li, and G. Y. Li, "Hypergraph theory: Applications in 5G heterogeneous ultra-dense networks," *IEEE Commun. Mag.*, vol. 55, pp. 70–76, Dec. 2017.
- [30] J. Chai, L. Feng, F. Zhou, P. Zhao, P. Yu, and W. Li, "Energy-efficient resource allocation based on hypergraph 3d matching for d2d-assisted mmTc networks," in *Proc. IEEE GLOBECOM 2018*, (Abu Dhabi, United Arab Emirates), pp. 1–7, Dec 2018.
- [31] H. Zhang, L. Song, and Z. Han, "Radio resource allocation for device-to-device underlay communication using hypergraph theory," *IEEE Trans. Wireless Commun.*, vol. 15, pp. 4852–4861, Jul. 2016.
- [32] X. Liu, Z. Li, N. Zhao, W. Meng, G. Gui, Y. Chen, and F. Adachi, "Transceiver design and multihop D2D for UAV IoT coverage in disasters," *IEEE Internet Things J.*, vol. 6, pp. 1803–1815, April 2019.
- [33] G. Karypis and V. Kumar, "Multilevel graph partitioning schemes," in *Proc. 24th Intern. Conf. Par.*, pp. 113–122, CRC Press, 1995.
- [34] P. Grippa, D. A. Behrens, F. Wall, and C. Bettstetter, "Drone delivery systems: job assignment and dimensioning," *Autonomous Robots*, vol. 43, pp. 261–274, Feb. 2019.
- [35] X. Lin, V. Jainanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H. Maattanen, M. Bergstrom, S. Euler, and Y. E. Wang, "The sky is not the limit: Lte for unmanned aerial vehicles," *IEEE Commun. Mag.*, vol. 56, pp. 204–210, Apr. 2018.
- [36] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, pp. 2109–2121, Mar. 2018.
- [37] Y. Li and L. Cai, "Uav-assisted dynamic coverage in a heterogeneous cellular system," *IEEE Netw.*, vol. 31, pp. 56–61, Jul. 2017.
- [38] Z. Xue, J. Wang, G. Ding, Q. Wu, Y. Lin, and T. A. Tsiftsis, "Device-to-device communications underlying UAV-supported social networking," *IEEE Access*, vol. 6, pp. 34488–34502, Jun. 2018.
- [39] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
- [40] A. Bretto, *Hypergraph Theory: An Introduction*. Springer Publishing Company, Incorporated, 2013.
- [41] D. J. A. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Comput. J.*, vol. 10, pp. 85–86, Jan. 1967.
- [42] M. N. Soorki, W. Saad, M. H. Manshaei, and H. Saidi, "Social community-aware content placement in wireless device-to-device communication networks," *IEEE Trans. Mobile Comput.*, vol. 18, pp. 1938–1950, Aug. 2019.
- [43] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, pp. 3949–3963, Jun. 2016.
- [44] M. E. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99 Suppl 1, pp. 2566–72, 2002.