Planning While Flying: A Measurement-Aided Dynamic Planning of Drone Small Cells

Ning Lu, *Member, IEEE,* Yi Zhou, *Member, IEEE,* Chenhao Shi, Nan Cheng, *Member, IEEE,* Lin Cai, *Senior Member, IEEE,* and Bin Li, *Member, IEEE*

***Abstract*—The deployment of drone small cells has emerged as a promising solution to agile provisioning of Internet backbone access for Internet of Things (IoT) devices, and many other types of users/devices. In this paper, we consider the problem of deploying a set of drone cells operating on multiple channels in a target area to provide access to the backbone/core network, which is formulated as a combinatorial network utility maximization problem. Since an offline and centralized solution to such a problem is not feasible, a low-complexity and distributed online algorithm is highly desired. Therefore, we propose a measurement-aided dynamic planning (MAD-P) algorithm, where the dispatched drones perform position and channel configurations autonomously on the fly based on the real-time measurement of network throughput to solve the problem in a distributed fashion during flight with minimal centralized control. We prove that the proposed MAD-P algorithm is asymptotically optimal, and investigate how long it takes for the convergence to stationarity under the MAD-P algorithm by giving a mixing time analysis. We also derive an upper bound of the performance gap in presence of measurement errors. Simulation results are provided to validate our analytic results and demonstrate the effectiveness of our algorithm.**

***Index Terms*—Unmanned Aerial Vehicles, drone small cell, Internet of Things, optimal dynamic planning, distributed algorithm, Markov approximation.**

I. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have gained rapid development in both military and civilian domains. Recently, substantial interest has been attracted in the development of drone-based airborne communication networks for applications such as harvesting data from sensors deployed in hardly accessible areas [1]–[6], enhancing coverage of cellular networks [7], aiding communications between the reader and RFIDs (Radio Frequency Identifiers) in battery-free networks [8], and among others. Particularly, recent research [9] has demonstrated the feasibility of mounting a *small cell base station* (sBS) on a flying drone to extend the last-mile connectivity to ground users/IoT devices that require accessing the backbone/core network in a region of interest.

N. Lu is with the Department of Computing Science, Thompson Rivers University, Canada (e-mail: nlu@tru.ca).

Y. Zhou (corresponding author) and C. Shi are with the School of Computer & Information Engineering, Henan University, Kaifeng, China. (e-mail: zhouyi@henu.edu.cn).

N. Cheng is with the School of Telecom. Engineering, Xidian University, Xi'an, 710071, Shaanxi, China. (e-mail: wmchengnan@gmail.com).

L. Cai is with the Department of Electrical & Computer Engineering, University of Victoria, Canada. (e-mail: cai@uvic.ca).

B. Li is with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, USA. (e-mail: binli@uri.edu).

Unlike deployment of fixed access infrastructure which typically requires time-consuming network plannings taking factors such as propagation, geographic limitations, and traffic distribution (e.g., [10], [11]) into considerations, and is difficult to change or to optimize over time, the main advantage of leveraging drone sBSs is that their deployment can be *agile* and *readily reconfigurable* due to the flexible mobility of drones. For example, a cluster of drone sBSs can be quickly launched regardless of geographical terrain; the position of drones can be adjusted in response to variations of wireless connectivity; and the cluster of operating drone sBSs can scale up and down in response to the change of network traffic demand. This has emerged as a promising solution to agile cellular/ Internet service provisioning in many application scenarios (e.g., coverage and capacity enhancement of 5G/Beyond 5G cellular networks during temporary events [12], [13], offloading the computation-intensive tasks onto the cloud from mobile IoT devices [14], [15], etc.).

A. The Drone Small Cells Planning Problem

In this paper, we consider the deployment of a set of drone sBSs in a target area of multiple cells and over multiple communication channels to serve the data traffic demand in the target area (i.e., a model of providing access to the backbone network), as shown in Fig. 1. The data traffic demand in each cell is assumed unknown (before deployment) and time-invariant but varies across cells.

1) The key question to answer: The question we would like to answer is that what is the best deployment (configuration) of drone sBSs such that the most data traffic demand can be served? This is not trivial because different configurations of drone sBSs in terms of cell placement and channel assignment may yield different system capacity and therefore the system throughput (i.e., the total demand being served). This is mainly due to the mutual interference generated under a certain configuration. For example, intuitively, we intend to send three drones (all we have) to the three most demanding cells so that we could fully utilize the capacity. However, it happens that the three cells are very close to each other geographically and there is only one communication channel available (in an extreme case) such that the total serving capacity might be very limited and less that the total traffic demand of the three cells due to the strong mutual interference. This would lead to a sub-optimal configuration, and there may exist other configurations that can provide higher system throughput.

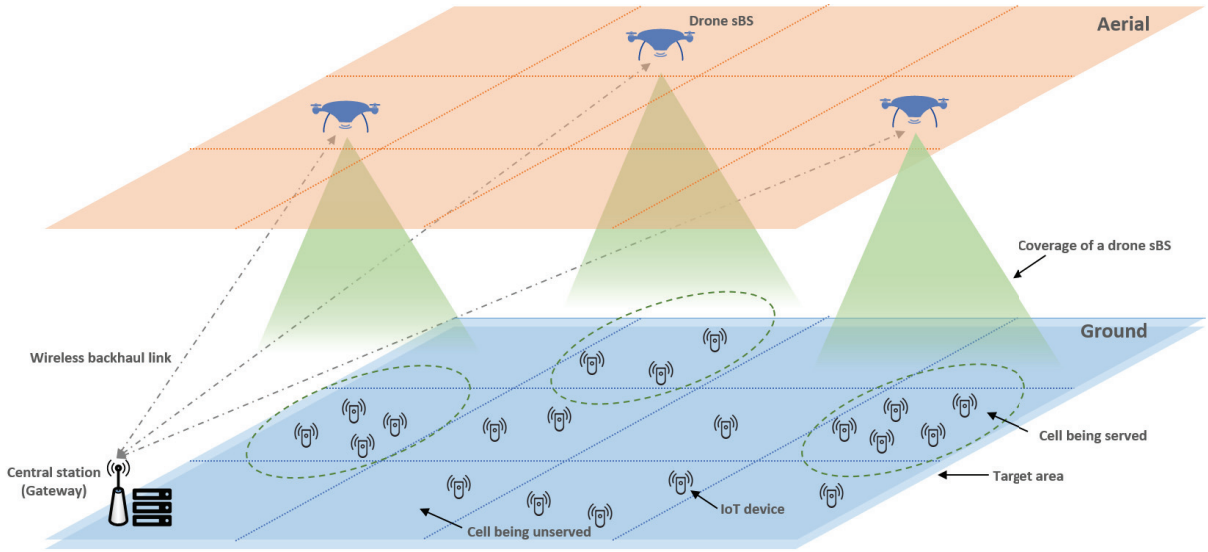


Fig. 1: A network of drone sSBs serving a target area.

2) The planning problem: The goal of the planning is to dispatch drone sBSs to cells and assign channels to drones such that the maximum system throughput is achieved. It can be seen that the planning process is essentially a joint cell placement (trying to assign drones to serve the most demanding cells) and channel assignment (trying to distribute drones evenly over channels and cells to minimize interference so as to maximize the total serving capacity) process given the data traffic demand of all cells (unknown before deployment). We further generalize the system throughput (the total demand being served) maximization problem to a utility maximization problem where the utility is a concave and increasing function of the throughput, representing a certain utility of interest (e.g., service satisfaction).

3) The need of low-complexity implementation: The drone sBSs planning problem is essentially a combinatorial optimization problem, i.e., finding an optimal configuration from the set of all possible configurations. However, since the data traffic demand is unknown before deployment (which is often the case in real-world scenarios), an offline and centralized solution (e.g., exhaustive search) is not feasible, and a low-complexity online solution is desired.

4) The need of distributed implementation: The desired planning algorithm should also be amenable to distributed implementations, such that computation is distributed over drones without having to be centralized at the backhaul gateway. Moreover, distributed implementations tend to be more scalable than centralized solutions. Also, due to the flexible mobility of drone sBSs, reconfiguration (moving to a different cell or switching to a different channel) can be agile and autonomous.

Therefore, we aim at solving the drone sBSs planning problem in a low-complexity and distributed way. In this paper, we propose an online measurement-aided dynamic planning (MAD-P) algorithm, where the dispatched drones perform self-configurations on the fly regarding the cell being located and the channel being used, based on the real-time measurement of network throughput. The proposed planning process is autonomous, and no prior knowledge of data traffic demand in the target area is required. The main contributions of this paper are highlighted as follows.

- The proposed algorithm leads to a joint planning of cell placement and channel selection, solves the underlining utility maximization problem by using the Markov approximation techniques, and is amenable to low-complexity and distributed implementation. In addition, we introduce a design parameter to control the preference of the exploration of cells over the exploration of channels.
- We prove the asymptotic optimality of the proposed algorithm. To understand how long it takes for the convergence to stationarity under the MAD-P algorithm, we derive an upper bound of the mixing time that captures the speed of convergence of the resulting Markov chain. We also show that the performance gap in presence of measurement errors is upper bounded.

B. Literature Review

The deployment of drone sBSs also faces great challenges. Key implementation issues have been investigated by recent studies, such as: (i) due to the lack of fiber-based connectivity, a viable wireless backhauling/fronthauling solution should be in place. The state-of-the-art solutions resort to different radio technologies including millimeter wave [16]–[18], satellite links [19], and free-space optics [20], each of which has its own pros and cons; and (ii) due to the short endurance of drones on a single charge, it is crucial to prolong the service duration or even provide persistent service during the mission via advanced charging technologies. The state-of-the-art solutions include equipping drones with solar panels [21], charging the battery during flight by using a high energy laser beam [22], and charging the battery by more powerful fixed-wing UAVs via wireless power transfer [23]. Moreover, some theoretical perspectives on efficient deployment of drone

sBSs have also been provided in the literature: (i) in [24], the optimal altitude of deployment is derived in maximizing the ground coverage; (ii) the service time maximization considering the travelling time of mobile small cells is given in [25]; (iii) dynamic repositioning of drone sBSs is considered in [26] to improve the spectral efficiency; (iv) the optimal placement of drone BSs considering energy efficiency is given in [27]; (v) algorithms for improving the communication throughput of a UAV network are proposed (e.g., [28], [29]); and (vi) channel assignment, or generally radio resource allocation, is considered in [30] and [31]. However, despite these advances, a joint cell placement and channel assignment planning process towards system throughput maximization has not been considered. Moreover, a low-complexity and distributed online algorithm relying on dynamic and autonomous configurations of drone sBSs should be in place towards the optimal planning, which has not been reported in existing research works either.

The remainder of the paper is organized as follows. Section II introduces the system model. The problem formulation is given in Section III. Section IV presents the proposed dynamic drone cell planning algorithm, followed by the performance analysis in Section V. In Section VI, we provide the simulation results and Section VII concludes the paper.

II. System Model

We consider deploying a set of drone sBSs, denoted by $\{1, 2, \ldots, N\} \triangleq [N]$, to a rectangular target area consisting of $L \times W$ square cells, each of which is r^2 in size. Each square cell, indexed by (l, w), where $l \in \{1, 2, \ldots, L\} \triangleq [L]$ and $w \in \{1, 2, \ldots, W\} \triangleq [W]$, can be fully covered by a drone sBS at a given height H. Let $\mathcal{Q}$ denote the set of $L \times W$ distinct cells, i.e., $\mathcal{Q} = \{(l, w) : l \in [L], w \in [W]\}$, and $q_n = (l_n, w_n)$ denote the cell where drone sBS n is deployed, where $l_n \in [L]$ and $w_n \in [W]$. We consider that a cell is at most served by one drone sBS, and hence multiple drone sBSs are not allowed to be coexisting in one cell. All drone sBSs are assumed directly connected to a central station via point-to-multipoint microwave/mmWave backhauling with sufficient bandwidth. The central station serves as a gateway to the backbone network and a hub for exchanging messages among drone sBSs, as shown in Fig. 1. For simplicity, we consider that each drone can be replaced in time to maintain system operation. A drone will monitor its battery status periodically. If the battery level is lower than a preset threshold, the drone will send a message to the control station to schedule a fully charged drone for replacement.

A multi-channel system is considered to reduce the co-channel interference among ground-air links in different cells. we consider that each drone sBS can select a channel to operate on from a set of M non-overlapping channels of equal bandwidth, denoted by $\{1, 2, \ldots, M\} \triangleq [M]$. The channel selected by drone sBS n can be represented as c_n, where $c_n \in [M]$. Moreover, we consider that the number of channels is limited and no more than the number of dispatched drones, i.e., $M < N$. Since each drone sBS is functioning in a certain cell under a certain channel at any given time, we represent by the drone cell planning process $\{\psi(t)\}_{t>0}$. At any time t, $\psi(t)$ takes a value f from a finite set $\mathcal{F}$, where $f \triangleq ((q_n, c_n))_{n\in[N]}$ represents a certain planning/configuration of all drone sBSs in the system and $\mathcal{F}$ denotes the set of all possible configurations.

We assume that a number of long-lived data flows are distributed in the target area unevenly, leading to heterogeneous data traffic demand in cells. Each long-lived data flow is a traffic stream that is always in the network and continually generates bits at a certain rate. The data traffic demand in the target area is denoted by $\mathbf{D} = (d_{l,w})_{l\in[L],w\in[W]}$, where $d_{l,w}$ is the aggregate rate generated from all data traffic flows in cell (l, w). In this work, the data traffic demand matrix $\mathbf{D}$ is assumed time-invariant. However, as what we will discuss in detail in Section IV-D, the optimality of the proposed algorithm remains if $\mathbf{D}$ changes slowly with time.

Given the drone cell planning $\psi(t)$ at time t, a certain capacity vector $\mathbf{S}(t) = (s_n^{\psi(t)})_{n\in[N]}$ is obtained, where $s_n^{\psi(t)}$ is the maximum data rate at which the data traffic demand is served by drone sBS n, and is determined by the interference pattern under $\psi(t)$ assuming a uniform ground-air transmission power over all cells. Therefore, the actual data demand served by (i.e., the ground-air throughput of) drone sBS n under $\psi(t)$ is given by $\gamma_n^{\psi(t)} = \min\{s_n^{\psi(t)}, d_{l_n,w_n}\}$, i.e., the throughput of a drone cell is the minimum of the traffic demand and service capacity, and we denote the ground-air throughput vector by $\mathbf{\Gamma}(t) = (\gamma_n^{\psi(t)})_{n\in[N]}$. Note that the unserved traffic demand will be either ignored or delivered by other network infrastructures (e.g., cellular macro cells and satellite) if available. Also note that $\gamma_n^{\psi(t)}$ can be measured by drone sBS n in real time. That is the reason why the prior knowledge of $\mathbf{D}$ is not required for our algorithm design. We will discuss the affect of measurement errors on the proposed algorithm in Section V.

III. Problem Formulation

The drone cell planning process $\{\psi(t)\}_{t>0}$ is essentially a joint *cell placement* and *channel assignment* process for all drone sBSs over time. The object of the planning process is to choose a configuration f from $\mathcal{F}$ given $\mathbf{D}$ such that the maximum system throughput is achieved, i.e., maximizing $\|\mathbf{\Gamma}(t)\|_1$, where $\|\cdot\|_1$ stands for the l_1 norm. However, we generalize the problem to a utility maximization problem in terms of $\mathbf{\Gamma}(t)$. Specifically, we consider the problem of finding the optimal drone cell planning $\psi(t)$ for all t to maximize the system-wide normalized utility, which is formulated as follows:

$$\mathbf{DDP}: \quad \max_{f\in\mathcal{F}} \quad \frac{1}{N}\sum_{n\in[N]} \Theta_n\left(\gamma_n^f\right),$$

where $\Theta_n(\gamma_n^f)$ is the utility obtained with respect to throughput γ_n^f by drone sBS n under configuration f. The utility functions Θ_n for all n are assumed to be twice differentiable, strictly increasing and concave, and bounded by finite constants $\Theta_{\min}$ and $\Theta_{\max}$, i.e.,

$$\Theta_{\min} \le \Theta_n(\gamma_n^f) \le \Theta_{\max}, \forall\ n \in [N] \text{ and } f \in \mathcal{F}.$$

As it can be seen, a planning policy (static) that selecting f^* by solving the combinatorial problem $\mathbf{DDP}$ is optimal.

However, such a policy tends to be computationally prohibitive since the size of $\mathcal{F}$ grows exponentially fast as the number of dispatched drones increases. The computational complexity of such a planning policy also depends on the number of available channels and the size of the target region. Therefore, our goal is to design a planning policy such that (i) the computational complexity is significantly reduced, and (ii) drone sBSs make planning decisions distributedly without the centralized coordination.

IV. Dynamic Drone Cell Planning

In this section, we first propose a Markov approximation to the original combinatorial optimization problem such that a static time-sharing policy is obtained. Then, we design a dynamic and distributed algorithm to implement the time-sharing policy by constructing a continuous time-reversible Markov Chain over all possible configurations.

A. Markov Approximation

Considering that the original problem $\mathbf{DDP}$ is hard to solve, inspired by [32], we obtain the following optimization problem by applying the Markov approximation techniques to the original problem:

$$\mathbf{DDP-MA}: \quad \max_{\boldsymbol{p}\geq 0} \quad \frac{p_f}{N}\sum_{n\in[N]}\Theta_n\big(\gamma_n^f\big)-\frac{1}{\beta}\sum_{f\in\mathcal{F}}p_f\log p_f$$
$$\text{subject to} \quad \sum_{f\in\mathcal{F}}p_f=1,$$

where $\boldsymbol{p}=(p_f)_{f\in\mathcal{F}}$ and p_f represents the fraction of time that configuration f is being used in the process $\{\psi(t)\}_{t>0}$; β is a positive constant and will be discussed later. The optimal solution $\boldsymbol{p}^*$ to this problem is given by

$$p_f^*=\frac{\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^f)\right)}{\sum_{f'\in\mathcal{F}}\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^{f'})\right)},\forall f\in\mathcal{F}. \qquad (1)$$

This corresponds to a time-sharing policy where each configuration f is selected according to its time fraction p_f^*. Intuitively, a configuration f that leads to the maximum system-wide normalized utility is being used most often as it has the largest time fraction among all configurations, and the performance gap between such a time-sharing policy and the optimal static policy would be closed as β tends to infinity. It is worth noting that solving both $\mathbf{DDP}$ and $\mathbf{DDP-MA}$ problems require the knowledge of the throughput vector $\mathbf{\Gamma}(f)=(\gamma_n^f)_{n\in[N]}$ for each f, which depends on the accurate modeling of the capacity vector $\mathbf{S}(f)=(s_n^f)_{n\in[N]}$ under a certain configuration f and a prior knowledge of $\mathbf{D}$. This could be very difficult and computationally infeasible especially when $\mathbf{D}$ becomes time-variant. Considering that the throughput vector under a certain configuration can be obtained by the real-time measurement, we design an online measurement-based algorithm to implement the time-sharing policy given by (1) in a distributed manner.

B. Distributed Algorithm Design in General

The idea of distributed algorithm design is that we first construct a continuous time reversible Markov chain with the state space being $\mathcal{F}$ and the desired stationary distribution given by (1); secondly, we design a distributed algorithm to realize the state transitions that drive the Markov Chain. Next, we will present the proposed Measurement-Aided Dynamic Planning (MAD-P) algorithm.

In the proposed algorithm design, a drone sBS may change its configuration by moving to a new cell or hopping to a new channel. Note that we do not consider the traveling time of drones which is negligible compared to the time scale where reconfigurations occur (e.g., at a time scale of minutes)[1]. Further, we assume the channel switching time (e.g., a few milliseconds) is negligible as well. Note that there is energy cost (due to movements) and slightly performance reduction (due to service discontinuity during cell reconfiguration) of repositioning drones. That could be carefully modeled and quantified. However, considering the cost of repositioning drones brings an additional layer of complexity in the algorithm design. It would be more moderate to consider it in our future works.

Since changing either the cell placement or the channel assignment of a single drone sBS out of N is sufficient to lead to a new configuration, with the MAD-P algorithm employed, a configuration f of drone sBSs may change to another configuration f' at time t by having only one of drone sBSs either moving to a new cell or hopping to a new channel. Given an $f\in\mathcal{F}$, the cell placement and channel assignment of drone sBS n can be represented as q_n^f and c_n^f respectively, where $q_n^f=(l_n^f,w_n^f)$, $l_n^f\in[L]$, $w_n^f\in[W]$, and $c_n^f\in[M]$. It is possible to change configuration f to some other configuration f' if there exists an n_1, such that regarding the following conditions,

c1: $q_{n_1}^f=q_{n_1}^{f'}$ and $c_{n_1}^f\neq c_{n_1}^{f'}$;
c2: $q_{n_1}^f\neq q_{n_1}^{f'}$ and $c_{n_1}^f=c_{n_1}^{f'}$;
c3: $\forall n_2\in[N]\setminus n_1$, $q_{n_1}^{f'}\neq q_{n_2}^f$;
c4: $\forall n_2\in[N]\setminus n_1$, $q_{n_2}^f=q_{n_2}^{f'}$ and $c_{n_2}^f=c_{n_2}^{f'}$,

either **c1** and **c4** hold (corresponding to a single drone sBS hopping to a new channel) or **c2**, **c3**, and **c4** hold (corresponding to a single drone sBS moving to a new cell that is not being occupied/served).

We denote by $\mathcal{Q}_n^f$ the set of cells that drone sBS n could possibly move to under current configuration f, which is dependent only on f and n. In the general design, we consider an *arbitrary mobility pattern*, where $\mathcal{Q}_n^f$ for each f and each n is an arbitrary set of unoccupied cells under f as long as the following two constraints hold:

c5: For each f and each n, $\forall q'\in\mathcal{Q}_n^f$, if f' is due to that n moves from cell q to q', then $q\in\mathcal{Q}_n^{f'}$;
c6: $\mathcal{Q}(N)\subset\{\{q_n^f\}_{n\in[N]}\}_{f\in\mathcal{F}}$,

where $\mathcal{Q}(N)$ is the set containing all N-combination sets[2] of $\mathcal{Q}$. **c5** states that it is always possible for each drone to move directly back to the old cell after moving to the new cell; while

[1] We consider later to limit the mobility of drones for minimizing the impact of traveling time.

[2] A k-combination of a set $\mathcal{S}$ is a subset of k distinct elements of $\mathcal{S}$.

c6 states that the set of cell configurations of N drones due to some arbitrary mobility pattern should include all possible cell configurations, each of which is a set of N distinct cells chosen from $\mathcal{Q}$. Note that $\{\{q_n^f\}_{n\in[N]}\}_{f\in\mathcal{F}}$ may contain permutations of one or more N-combination sets of $\mathcal{Q}$. We consider that all permutations of the same N-combination set of $\mathcal{Q}$ with exactly the same channel configuration of drones lead to the same system utility, given that all drones are identical.

Moreover, let $\mathcal{C}_n^f$ be the set of channels that are not used by drone sBS n under current configuration f, i.e., $\mathcal{C}_n^f = [M] \setminus c_n^f$. With the MAD-P algorithm employed, at a certain time instant, one of drone sBSs, say n, under configuration f would either move to a new cell in $\mathcal{Q}_n^f$ or hop to a new channel in $\mathcal{C}_n^f$ probabilistically, leading to a transition from f to a new configuration, say f'. The MAD-P algorithm is described in Algorithm 1.

In the proposed algorithm, we introduce a fixed parameter $\zeta \in (0,1)$ to control the transitions between configurations. Clearly, increasing ζ increases the chance of seeing a cell reconfiguration; while decreasing ζ increases the chance of seeing a channel hopping. Under different system settings, we may want to choose different values of ζ. For example, in the case where we have $M \approx N \ll L \times W$, a value of ζ approaching one is preferable to explore the target area as much as possible to identify the most demanding cells to serve. In the extreme case where $\zeta = 1$, our algorithm reduces to a pure cell placement algorithm; while in the extreme case where $\zeta = 0$, our algorithm reduces to a pure channel assignment algorithm. Therefore, the MAD-P algorithm is considered as a randomized algorithm unifying both cell placement and channel assignment processes.

Proposition 1: *With the MAD-P algorithm employed, the drone cell planning process $\{\psi(t)\}_{t>0}$ is a continuous-time time-reversible ergodic Markov chain with the stationary distribution given by* (1)*.*

Proof: It can be seen that $\{\psi(t)\}_{t>0}$ is a continuous-time homogeneous Markov chain since the amount of time the planning process stays in configuration/state f is exponentially distributed and dependent only on state f, and the transition probabilities are independent of time. Moreover, $\{\psi(t)\}_{t>0}$ is irreducible since it is always possible to transit from state f to any other state f' in some finite time (either directly or indirectly with $\mathcal{F}$ being finite and due to **c5**). Therefore, $\{\psi(t)\}_{t>0}$ has a unique stationary distribution from [33]. Next, we will show that $\{\psi(t)\}_{t>0}$ is time-reversible such that according to [34] the stationary distribution of $\{\psi(t)\}_{t>0}$ is given by (1).

To do so, it suffices to show that for any $f \in \mathcal{F}$ and any f' that f could possibly transit to, under Algorithm 1, the detailed balance equation holds, i.e.,

$$p_f^* \lambda(f \to f') = p_{f'}^* \lambda(f' \to f) \tag{3}$$

where $\lambda(f \to f')$ and $\lambda(f' \to f)$ are positive transition rates between state f and f'.

In Algorithm 1, since each drone sBS n counts down independently with a rate $\frac{\alpha\left(\zeta|\mathcal{Q}_n^f|+(1-\zeta)|\mathcal{C}_n^f|\right)}{\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^f)\right)}$, denoted by λ_n^f, in state f, the drone cell planning process will leave state f upon one of countdown processes expires at a rate of $\sum_{n\in[N]}\lambda_n^f$. Without loss of generality, we assume that the countdown process of drone sBS n^* expires first among all drone sBSs, i.e., the transition from f to f' is due to that drone sBS n^* either moves to a new cell or hops to a new channel. Since the length of each countdown process follows an individual exponential distribution, the probability

Algorithm 1 MAD-P Algorithm

Input: The set of drone sBSs $[N]$, the set of cells $((l,w))_{l\in[L],w\in[W]}$, the set of channels $[M]$, the utility functions Θ_n for all $n \in [N]$, and predefined parameters $\zeta \in (0,1)$, $\alpha > 0$, $\beta > 0$, and H.

Output: Drone cell planning process $\{\psi(t)\}_{t>t_0}$.

initialization:

At $t = t_0$, arbitrarily dispatch N drones to N distinct cells at the given height H and each drone randomly and uniformly selects a channel to operate on. This leads to an initialized configuration $f_0 \in \mathcal{F}$. Each drone sBS n obtains its ground-air throughput $\gamma_n^{f_0}$ via measurement and informs the central station of $(q_n^{f_0}, c_n^{f_0})$ and $\Theta_n(\gamma_n^{f_0})$. Then, the central station broadcasts f_0 and $(\Theta_n(\gamma_n^{f_0}))_{n\in[N]}$ to all drone sBSs. We have $\{\psi(t) = f_0\}_{t_0<t\leq t_1}$, where t_1 is the time when the 1^{st} reconfiguration occurs.

while $i > 0$ (i^{th} reconfiguration) **do**

  for $n \in [N]$ **do**

    As soon as drone sBS n receives f_{i-1} and $(\Theta_n(\gamma_n^{f_{i-1}}))_{n\in[N]}$ from the central station, it finds $\mathcal{Q}_n^{f_{i-1}}$ and $\mathcal{C}_n^{f_{i-1}}$, and counts down according to a generated random number from an exponential distribution with mean equal to

$$\frac{\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^{f_{i-1}})\right)}{\alpha\left(\zeta|\mathcal{Q}_n^{f_{i-1}}| + (1-\zeta)|\mathcal{C}_n^{f_{i-1}}|\right)}; \tag{2}$$

  end for

  At $t = t_i$, the countdown of drone sBS n^* expires first among all drone sBSs. Drone sBS n^* immediately informs other drones via the central station to terminate their countdown processes, and does the following:

  With probability $\frac{\zeta|\mathcal{Q}_{n^*}^{f_{i-1}}|}{\zeta|\mathcal{Q}_{n^*}^{f_{i-1}}|+(1-\zeta)|\mathcal{C}_{n^*}^{f_{i-1}}|}$, drone n^* randomly and uniformly moves to one of the cells in $\mathcal{Q}_{n^*}^{f_{i-1}}$; while with probability $\frac{(1-\zeta)|\mathcal{C}_{n^*}^{f_{i-1}}|}{\zeta|\mathcal{Q}_{n^*}^{f_{i-1}}|+(1-\zeta)|\mathcal{C}_{n^*}^{f_{i-1}}|}$, drone n^* randomly and uniformly switches to one of the channels in $\mathcal{C}_{n^*}^{f_{i-1}}$.

  for $n \in [N]$ **do**

    With the new configuration f_i, drone sBS n obtains its ground-air throughput $\gamma_n^{f_i}$ via measurement and informs the central station of $(q_n^{f_i}, c_n^{f_i})$ and $\Theta_n(\gamma_n^{f_i})$.

  end for

  The central station immediately broadcasts f_i and $(\Theta_n(\gamma_n^{f_i}))_{n\in[N]}$ to all drone sBSs.

  Let $i \leftarrow i+1$, and we have $\{\psi(t) = f_{i-1}\}_{t_{i-1}<t\leq t_i}$.

end while

that the countdown of drone sBS n^* expires first is given by $\frac{\lambda_{n^*}^f}{\sum_{n\in[N]}\lambda_n^f}$. Considering the case where f and f' differ only on the cell configuration of drone sBS n^* and it can be seen from Algorithm 1 that under f drone sBS n^* moves to $q_{n^*}^{f'}$ in $\mathcal{Q}_{n^*}^f$ with probability $\frac{\zeta|\mathcal{Q}_{n^*}^f|}{\zeta|\mathcal{Q}_{n^*}^f|+(1-\zeta)|\mathcal{C}_{n^*}^f|}\cdot\frac{1}{|\mathcal{Q}_{n^*}^f|}$, we have

$$\begin{aligned}
&p_f^*\lambda(f\to f')\\
&=\frac{\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^f)\right)}{\sum_{f''\in\mathcal{F}}\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^{f''})\right)}\left(\sum_{n\in[N]}\lambda_n^f\right)\\
&\quad\cdot\left(\frac{\lambda_{n^*}^f}{\sum_{n\in[N]}\lambda_n^f}\right)\frac{\zeta}{\zeta|\mathcal{Q}_{n^*}^f|+(1-\zeta)|\mathcal{C}_{n^*}^f|}\\
&=\frac{\alpha\zeta}{\sum_{f''\in\mathcal{F}}\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^{f''})\right)}. \qquad (4)
\end{aligned}$$

Since state f' transits to state f by changing $q_{n^*}^{f'}$ to $q_{n^*}^f$, similarly, we have

$$p_{f'}^*\lambda(f'\to f)=\frac{\alpha\zeta}{\sum_{f''\in\mathcal{F}}\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^{f''})\right)}. \qquad (5)$$

By (4) and (5), we have (3). Now consider the case where f transits to f' by changing $c_{n^*}^f$ to $c_{n^*}^{f'}$, and f' transits to f by changing $c_{n^*}^{f'}$ to $c_{n^*}^f$. Similarly, we can prove that

$$\begin{aligned}
p_f^*\lambda(f\to f')&=p_{f'}^*\lambda(f'\to f)\\
&=\frac{\alpha(1-\zeta)}{\sum_{f''\in\mathcal{F}}\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^{f''})\right)}. \qquad (6)
\end{aligned}$$

■

Proposition 1 shows that with the MAD-P algorithm, we achieve the desired stationary distribution over configurations such that the time-sharing policy according to (1) is implemented in a distributed manner where each drone sBS makes the decision locally on activating either a channel reconfiguration or a cell reconfiguration.

C. Two Special Designs

Next, we propose two special designs based on the MAD-P algorithm by specifying the mobility pattern of drones whenever they reconfig their cell locations.

1) Maximum Mobility: In this special design, at any time a drone needs to change its cell placement, the drone can reach any unserved cell in the target area. Specifically, we consider that for each configuration f and each drone sBS n, $\mathcal{Q}_n^f$ contains all unoccupied cells, i.e., $\mathcal{Q}_n^f=\{q: q\in\mathcal{Q}, q\neq q_{n'}^f, \forall n'\in[N]\}$, which leads to the MAD-P algorithm with Maximum Mobility, called the MAD-P/MM algorithm. Note that actually given a configuration f, all drone sBSs have the same $\mathcal{Q}_n^f$, denoted by $\mathcal{Q}^f$, and we have $|\mathcal{Q}^f|=LW-N$.

2) Limited Mobility: In this special design, at any time a drone needs to change its cell placement, the drone can only move to one of the unoccupied adjacent cells. Specifically, we consider that for each configuration f and each drone sBS n, $\mathcal{Q}_n^f=\{q=(l,w): |l-l_n^f|+|w-w_n^f|=1, q\in\mathcal{Q}, q\neq q_{n'}^f, \forall n'\in[N]\}$, such that each drone randomly and uniformly walks to one of the unoccupied adjacent cells (at most four) with a travelling distance of r. This gives us the MAD-P algorithm with limited mobility, called the MAD-P/LM algorithm. Limiting the mobility of drones is one way of minimizing the impact of traveling time since a drone sBS only needs to move to one of the neighboring cells (unserved) if the cell configuration needs to be changed.

D. Discussion on Implementations

1) Exploration vs. Exploitation: In the MAD-P algorithm, the time spent in a certain configuration f is exponentially distributed with a rate equal to

$$\sum_{n\in[N]}\lambda_n^f=\frac{\zeta\sum_{n\in[N]}|\mathcal{Q}_n^f|+(1-\zeta)\sum_{n\in[N]}|\mathcal{C}_n^f|}{\frac{1}{\alpha}\exp\left(\frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^f)\right)}. \qquad (7)$$

As it can be seen, the configuration that leads to a larger system-wide normalized utility tends to stay longer before it transits to other configurations. We can even prolong the sojourn time of a configuration (on average) by decreasing α such that we could potentially exploit a good configuration as much as possible when it occurs. We can also increase α to speed up the transitions among configurations. This particularly works for the scenario where the data traffic demand varies a lot across different cells since the algorithm should be able to identify the most demanding cells very quickly (i.e., the more exploration, the better). It is worth noting that tuning α does not change the stationary distribution of the planning process, and only affects the transition rates. As discussed before, the stationary distribution can be adjusted by changing β (from (1)). For example, we can simply increase the value of β to increase the time fraction that the system spent in the optimal configuration f^*. However, this could slow down the exploration over configurations according to (7). In a nutshell, the parameters α and β are used to tune the short-run tradeoff between exploration and exploitation of configurations in the planning process.

2) Message Passing: Note that the proposed algorithm depends on message passing among drone sBSs with the central station involved. It is associated with state transitions and is necessary to facilitate the decision making process at each drone sBS. The overhead of message passing will be reduced in terms of *messages per unit time* if we slow down the state transitions in the planning process by tuning the parameters. In real-world implementations, the overhead introduced by message passing can be further reduced via the proper design of communication protocols.

3) Time-Varying Traffic Demand: The data traffic demand matrix $\mathbf{D}$ is assumed time-invariant. However, the optimality of the proposed algorithm can remain if $\mathbf{D}$ changes slowly with time. In addtion, we need to choose an increasing function $g(\cdot)$ and reformulate the **DDP** problem as $\max_{f\in\mathcal{F}} g(\frac{1}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^f))$. For example, $g(\cdot)=\log\log(\cdot)$. Allowing $\mathbf{D}$ changes slowly actually corresponds to the relaxation of time-scale separation assumption in the research of queue-length based CSMA algorithm, which has been well studied in the literature (e.g., [35]).

V. Performance Analysis

In this section, we first prove that the proposed MAD-P algorithm is asymptotically optimal to the original **DDP** problem. Then, we will consider the problem of bounding the time taken by the planning process $\{\psi(t)\}_{t>0}$ to reach the desired stationary distribution under the MAD-P/MM algorithm. We leave the mixing time analysis for the MAD-P/LM algorithm and the general MAD-P algorithm for future works. Finally, we will discuss the impact of measurement errors on the algorithm performance.

A. Asymptotic Optimality

Theorem 1: *The MAD-P algorithm is asymptotically utility-optimal to the* **DDP** *problem, i.e., for any* $\epsilon \in (0,1)$, $\delta \in (0,1)$, *there exists a constant* $C > 0$, *such that whenever* $U(f^*) > C$, *we have*

$$\mathbf{Pr}\{U(f) > (1-\epsilon)U(f^*)\} > 1-\delta, \tag{8}$$

where $U(f) = \frac{\beta}{N}\sum_{n\in[N]}\Theta_n(\gamma_n^f)$ *is the weighted system-wide normalized utility under configuration* f *at time* t *and* $U(f^*) = \max_{f\in\mathcal{F}} U(f)$.

Proof: Let $\mathcal{I}$ denote the set of configurations that generate a weighted system-wide normalized utility less than $(1-\epsilon)U(f^*)$, i.e.,

$$\mathcal{I} = \{f \in \mathcal{F} : U(f) < (1-\epsilon)U(f^*)\}$$

Recall that p_f^* is the probability that configuration f is being used in steady state under the proposed MAD-P algorithm. Let $Z = \sum_{f\in\mathcal{F}}\exp(U(f))$. Then, we have

$$\begin{aligned}
&\mathbf{Pr}\{U(f) < (1-\epsilon)U(f^*)\}\\
&\quad= \sum_{f\in\mathcal{I}} p_f^* = \sum_{f\in\mathcal{I}}\frac{1}{Z}\exp(U(f))\\
&\quad\le \frac{|\mathcal{I}|}{Z}\exp\left((1-\epsilon)U(f^*)\right)\\
&\quad< |\mathcal{I}|\exp(-\epsilon U(f^*))\\
&\quad< \frac{(LW)!}{(LW-N)!}M^N\exp(-\epsilon U(f^*))
\end{aligned}$$

where the last two inequities are due to the fact that $\exp(U(f^*)) < Z$ and $\mathcal{I} < |\mathcal{F}| \le \frac{(LW)!}{(LW-N)!}M^N$, respectively. As it can be seen, if we let $C = \frac{1}{\epsilon}(N\ln M + \ln\frac{(LW)!}{(LW-N)!} + \ln\frac{1}{\delta})$, and whenever $U(f^*) > C$, we have (8). Hence, the theorem holds. ∎

Note that in theory Theorem 1 also provides a guideline for the design of utility functions $(\Theta_n)_{n\in[N]}$ for drone sBSs. Once parameters ϵ and δ are fixed, the constant C can be figured out such that $(\Theta_n)_{n\in[N]}$ can be designed wisely to ensure $U(f^*) > C$ under the optimal configuration f^*. In practice, it is difficult to obtain $U(f^*)$ before we actually run the algorithm since the ground-air throughput vector $\mathbf{\Gamma}(f) = (\gamma_n^f)_{n\in[N]}$ needs to be measured on the go. Furthermore, we do not have the knowledge that which configuration is optimal until all the configurations in $\mathcal{F}$ have been explored. One practical strategy is that we can run the algorithm for a short period of time T, find out $f_{\max} = \arg\max_{\psi(t),t\in(0,T]} U(\psi(t))$, adjust $(\Theta_n)_{n\in[N]}$ to have $U(f_{\max}) > C$, and then rerun the algorithm. As a result of doing this, we must have $U(f^*) > C$. On the other hand, since $U(f) \ge \beta\Theta_{\min}$, $\forall f \in \mathcal{F}$, we could simply have $\Theta_{\min} > C/\beta$ when we design $(\Theta_n)_{n\in[N]}$.

B. Speed of Convergence to Stationarity

The time sharing among configurations according to (1) is achieved once the resulting Markov chain enters its steady state (or equivalently, reaches its stationary distribution $\boldsymbol{p}^*$). Therefore, it is important to understand how long it takes for the convergence to stationarity under the MAD-P/MM algorithm, and how different parameters affect the convergence behavior. Thus, we provide an upper bound of the mixing time that captures the speed of convergence of the resulting Markov chain $\{\psi(t)\}_{t>0}$ under the MAD-P/MM algorithm.

The mixing time (convergence time) of $\{\psi(t)\}_{t>0}$ is defined as follows:

$$\tau_{mix}(\epsilon) \triangleq \inf\left\{t \ge 0 : \max_{f\in\mathcal{F}} d_{TV}\left(\mathbf{P}_{f,\cdot}(t), \boldsymbol{p}^*\right) \le \epsilon\right\},$$

where $d_{TV}(\cdot,\cdot)$ denotes the total variation distance of two probability distributions, and $\mathbf{P}_{f,\cdot}(t)$ is the probability distribution of $\psi(t)$ over $\mathcal{F}$ at time t given that the initial state is f.

1) Uniformization: To facilitate the mixing time analysis, motivated by [32], we follow the technique of *uniformization* in [36] to obtain a uniformization version of the continuous-time Markov chain $\{\psi(t)\}_{t>0}$, which is characterized by an embedded discrete-time Markov chain Ψ and a Poisson process with rate ρ. By Theorem 3.4 in [36], the original Markov chain and its uniformization version have the same transition rates between states.

We denote by $\mathbf{Q} = (Q_{ff'})_{|\mathcal{F}|\times|\mathcal{F}|}$ [3] the transition rate matrix of $\{\psi(t)\}_{t>0}$. Let $\widehat{\mathbf{P}} = (\widehat{P}_{ff'})_{|\mathcal{F}|\times|\mathcal{F}|}$ denote the one-step transition matrix of Ψ. By the uniformization technique, we have

$$\widehat{\mathbf{P}} = \mathbf{I} + \rho^{-1}\mathbf{Q}, \tag{9}$$

where $\mathbf{I}$ is the identity matrix. Recall that the time required to make a transition from configuration f has an exponential distribution with rate given by (7). With the MAD-P/MM algorithm employed, $\zeta\sum_{n\in[N]}|\mathcal{Q}_n^f| + (1-\zeta)\sum_{n\in[N]}|\mathcal{C}_n^f|$ in (7) is equal to $N\cdot R$ for any configuration f, where $R \triangleq \zeta(LW-N) + (1-\zeta)(M-1)$. Then, we choose

$$\rho = \alpha NR\sum_{f\in\mathcal{F}}\exp(-U(f)).$$

From Algorithm 1, it can be obtained that $Q_{ff'}^{cell} = \alpha\zeta\exp(-U(f))$, where $Q_{ff'}^{cell}$ denotes the transition rate from f to some different f' due to one of the drone sBSs moving to a new cell; $Q_{ff'}^{channel} = \alpha(1-\zeta)\exp(-U(f))$, where $Q_{ff'}^{channel}$ denotes the transition rate from f to some different f' due to one of the drone sBSs hopping to a new channel. Accordingly, we have $\widehat{P}_{ff'}^{cell} = \frac{\alpha\zeta\exp(-U(f))}{\rho}$ and $\widehat{P}_{ff'}^{channel} = \frac{\alpha(1-\zeta)\exp(-U(f))}{\rho}$ for any f' that f could possibly transit to.

[3] With a slight abuse of notation, we also use f to denote the index of configuration f in the transition rate matrix.

Next, we construct the Markov chain Ψ as follows according to (9). When the current state of Ψ is f:

1. Choose a drone sBS n^* from $[N]$ uniformly at random (i.e., with probability $\frac{1}{N}$);
2. With probability $\frac{\alpha N R \exp(-U(f))}{\rho}$, it is allowed to change state; otherwise, n^* remains its configuration.
3. When it is allowed to change state, do the following: with probability $\frac{\zeta(LW-N)}{R}$, drone n^* randomly and uniformly moves to one of the cells in $\mathcal{Q}_{n^*}^f$; otherwise, drone n^* randomly and uniformly switches to one of the channels in $\mathcal{C}_{n^*}^f$.

As it can be seen, by the above process, we indeed have a Markov chain Ψ according to (9). Next, we bound the mixing time of $\{\psi(t)\}_{t>0}$ through the mixing time analysis of Ψ given that $\{\psi(t)\}_{t>0}$ is represented directly as a discrete-time Markov chain Ψ with transitions governed by an independent Poisson process with rate ρ.

2) Coupling: We now apply the path coupling techniques [32], [37] to construct a coupling of Ψ for the mixing time analysis. A coupling of the Markov chain Ψ on the state space $\mathcal{F}$ is a pair process $(\Psi_t, \widetilde{\Psi}_t)$ on $\mathcal{F} \times \mathcal{F}$, such that (i) each of $(\Psi_t, \cdot)$ and $(\cdot, \widetilde{\Psi}_t)$, viewed separately, is a faithful copy of the Markov chain Ψ, and (ii) if $\Psi_t = \widetilde{\Psi}_t$, we have $\Psi_{t+1} = \widetilde{\Psi}_{t+1}$. By the path coupling techniques in [37], we only need to construct a one-step coupling starting with any two adjacent states on a path, i.e., to construct $(\Psi_0, \widetilde{\Psi}_0) \to (\Psi_1, \widetilde{\Psi}_1)$, where Ψ_0 can make a transition to $\widetilde{\Psi}_0$ in one step.

Next, we define the Hamming distance $H(f, f')$ between any two states f and f' in $\mathcal{F}$, which is simply the number of drone sBSs $n \in [N]$ such that $(q_n^f, c_n^f) \neq (q_n^{f'}, c_n^{f'})$[4]. Initially, we have $H(\Psi_0, \widetilde{\Psi}_0) = 1$, where Ψ_0 and $\widetilde{\Psi}_0$ are two adjacent states on a path, i.e., they only differ by the cell configurations or the channel configurations of the same drone sBS. With path coupling, the following lemma shows that the Hamming distance between Ψ_1 and $\widetilde{\Psi}_1$ decreases in expectation compared to $H(\Psi_0, \widetilde{\Psi}_0)$, i.e., $\mathbb{E}[H(\Psi_1, \widetilde{\Psi}_1)|\Psi_0, \widetilde{\Psi}_0] < H(\Psi_0, \widetilde{\Psi}_0)$.

Lemma 1: *With path coupling under the MAD-P/MM algorithm, for any pair of Ψ_0 and $\widetilde{\Psi}_0$ (adjacent states) on $\mathcal{F} \times \mathcal{F}$, we have*

$$\mathbb{E}\left[H(\Psi_1, \widetilde{\Psi}_1)|\Psi_0, \widetilde{\Psi}_0\right] \leq (1-B)H(\Psi_0, \widetilde{\Psi}_0),$$

where $0 < B < 1$, specifically,

$$B = \frac{(N-1)\left[1+\kappa-\exp\left(2\beta(\Theta_{\max}-\Theta_{\min})\right)\right]}{N|\mathcal{F}|\exp\left(\beta(\Theta_{\max}-\Theta_{\min})\right)},$$

given that $\kappa = \min\left\{\frac{\zeta(LW-2N+2)}{(N-1)R}, \frac{(1-\zeta)M}{(N-1)R}\right\}$, $0 < \beta < \frac{\ln(1+\kappa)}{2(\Theta_{\max}-\Theta_{\min})}$, and $N < \frac{LW}{2}+1$.

Proof: See Appendix for the details. ∎

Theorem 2: *(Rapid Mixing) Under the MAD-P/MM algorithm, the mixing time of the planning process $\{\psi(t)\}_{t>0}$ is upper bounded as follows:*

$$\tau_{mix}(\epsilon) \leq \frac{\frac{1}{\alpha R}\exp\left(\beta(2\Theta_{\max}-\Theta_{\min})\right)\ln\frac{2N}{\epsilon}}{(N-1)\left[1+\kappa-\exp\left(2\beta(\Theta_{\max}-\Theta_{\min})\right)\right]},$$

given that $\kappa = \min\left\{\frac{\zeta(LW-2N+2)}{(N-1)R}, \frac{(1-\zeta)M}{(N-1)R}\right\}$, $0 < \beta < \frac{\ln(1+\kappa)}{2(\Theta_{\max}-\Theta_{\min})}$, and $N < \frac{LW}{2}+1$.

Proof: For any $(\Psi_t, \widetilde{\Psi}_t) \in \mathcal{F} \times \mathcal{F}$, applying Lemma 1 iteratively, we have

$$\begin{aligned}\mathbf{Pr}\{\Psi_t \neq \widetilde{\Psi}_t\} &= \mathbf{Pr}\{H(\Psi_t, \widetilde{\Psi}_t) \geq 1\} \\ &\leq \mathbb{E}\left[H(\Psi_t, \widetilde{\Psi}_t)\right] \leq (1-B)^t \cdot \mathit{diam}(\mathcal{F}),\end{aligned}$$

where $\mathit{diam}(\mathcal{F})$ is the diameter of $\mathcal{F}$, i.e., the maximum of the minimum number of transitions required to go from Ψ to Ψ' over all pairs of positive-recurrent states $\Psi, \Psi' \in \mathcal{F}$. Under the MAD-P/MM algorithm, $\mathit{diam}(\mathcal{F}) \leq 2N$.

Therefore, for the Markov chain Ψ according to (9) and followed by the coupling lemma in [37] (Lemma 1), we have

$$d_{TV}\left(\widehat{\mathbf{P}}^t(\Psi_0, \cdot), \widehat{\mathbf{P}}^t(\widetilde{\Psi}_0, \cdot)\right) \leq \mathbf{Pr}\{\Psi_t \neq \widetilde{\Psi}_t\} \leq 2N(1-B)^t,$$

where $\widehat{\mathbf{P}}^t(\Psi_0, \cdot)$ ($\widehat{\mathbf{P}}^t(\widetilde{\Psi}_0, \cdot)$) denotes the t-step transition probability distribution starting from Ψ_0 ($\widetilde{\Psi}_0$).

Consider the Markov chain $\{\psi(t)\}_{t>0}$. Recall that $\mathbf{P}_{f,\cdot}(t)$ the probability distribution over $\mathcal{F}$ at time t given that the initial state is f, and $\boldsymbol{p}^* = (p_f^*)_{f\in\mathcal{F}}$ given by (1). Hence,

$$\begin{aligned}&d_{TV}\left(\mathbf{P}_{f,\cdot}(t), \boldsymbol{p}^*\right) \\ &= d_{TV}\left(\sum_{k=0}^{\infty}\widehat{\mathbf{P}}^k(f,\cdot)\frac{\exp(-\rho t)(\rho t)^k}{k!}, \boldsymbol{p}^*\right) \\ &\overset{(a)}{\leq} \sum_{k=0}^{\infty}\frac{\exp(-\rho t)(\rho t)^k}{k!} d_{TV}\left(\widehat{\mathbf{P}}^k(f,\cdot), \boldsymbol{p}^*\right) \\ &\leq 2N\sum_{k=0}^{\infty}\frac{\exp(-\rho t)(\rho(1-B)t)^k}{k!} \\ &= 2N\exp(-\rho B t)\end{aligned}$$

where (a) is due to Jensen's inequality. Therefore, we have

$$\tau_{mix}(\epsilon) \leq \frac{\ln\frac{2N}{\epsilon}}{\rho B} \leq \frac{\frac{1}{\alpha R}\exp\left(\beta(2\Theta_{\max}-\Theta_{\min})\right)\ln\frac{2N}{\epsilon}}{(N-1)\left[1+\kappa-\exp\left(2\beta(\Theta_{\max}-\Theta_{\min})\right)\right]},$$

where $\kappa = \min\left\{\frac{\zeta(LW-2N+2)}{(N-1)R}, \frac{(1-\zeta)M}{(N-1)R}\right\}$. ∎

From Theorem 2, it can be seen that the mixing time decreases with the increase of α. In other words, the planning process $\{\psi(t)\}_{t>0}$ rapidly converges to its stationary distribution if α goes large. This is because, as discussed before, increasing α can speed up the transitions among configurations. Note that we require $N < \frac{LW}{2}+1$ to have the rapid mixing of $\{\psi(t)\}_{t>0}$. Intuitively, compared to the total number of cells LW, if N is relatively large, it typically takes longer to transit between non-adjacent configurations since drones do not have much freedom to move.

C. Effects of Measurement Error

In the MAD-P algorithm, the ground-air throughput (γ_n^f, for any drone sBS n under any configuration f) should be measured in real time, based on which the transition rates that drive the planning process can be determined (see (2)). However, measurements of ground-air throughput could be inaccurate so that the algorithm would compute the planning based on a

[4]We have $(q_n^f, c_n^f) \neq (q_n^{f'}, c_n^{f'})$ if $q_n^f \neq q_n^{f'}$ or $c_n^f \neq c_n^{f'}$ or both.

different set of transition rates, leading to a performance gap towards the optimal solution $\boldsymbol{p}^*$. It is important to understand how measurement errors affect the algorithm performance and hence further characterize the performance gap. For simplicity, based on the throughput measurements from all drone sBSs under f, the broadcasted weighted system-wide normalized utility $\widehat{U}(f)$ by the central station undergoes random errors in a bounded region $[-\delta_f, \delta_f]$ compared to the true utility $U(f)$, where δ_f is positive and dependent only on f. Hence, we have $\widehat{U}(f) \in [U(f)-\delta_f, U(f)+\delta_f]$. Further, we assume that $\widehat{U}(f)$ takes only $2n_f+1$ discrete values, i.e.,

$$\Big[U(f)-\delta_f, U(f)-\frac{n_f-1}{n_f}\delta_f, \ldots, U(f)-\frac{1}{n_f}\delta_f, U(f), \\ U(f)+\frac{1}{n_f}\delta_f, \ldots, U(f)+\frac{n_f-1}{n_f}\delta_f, U(f)+\delta_f\Big],$$

following certain probability distribution $\boldsymbol{p}_f^{error}$, where n_f is a positive integer only dependent on f. With the above modeling, the performance gap in the presence of measurement errors is bounded by the following theorem.

Theorem 3: *Under the MAD-P algorithm, the performance gap on the expected system utility in the presence of measurement errors is given by*

$$\left|\bar{\boldsymbol{p}}\boldsymbol{U}^T - \boldsymbol{p}^*\boldsymbol{U}^T\right| = \left|\sum_{f\in\mathcal{F}}(\bar{p}_f - p_f^*)U(f)\right| \\ \leq 2\Theta_{\max}\big(1-\exp(-2\beta\delta_{\max})\big),$$

where $\bar{\boldsymbol{p}} = (\bar{p}_f)_{f\in\mathcal{F}}$ denotes the stationary distribution of $\{\psi(t)\}_{t>0}$ in the present of measurement errors, $\boldsymbol{U} = (U(f))_{f\in\mathcal{F}}$ denotes the vector of weighted system-wide normalized utilities, and $\delta_{\max} = \max_{f\in\mathcal{F}}\delta_f$.

The proof of Theorem 3 is omitted here since it is almost the same to the proof given by [38]. From Theorem 3, it can be seen that the performance gap decreases exponentially fast as the measurement errors diminish.

VI. Simulation Results

In this section, we present three sets of simulations to demonstrate the performance of the proposed MAD-P algorithm, and how different design parameters (ζ, α, and β) and measurement errors affect its performance.

In the first set of simulations, we consider a target area consisting of 4×3 square cells, each of which is 100×100 m^2 in size. There are $N=4$ drone sBSs dispatched to the target area with a height of 150 m, and there are three channels available for use, each of 15 MHz bandwidth. We consider that a fixed number of long-lived data flows are distributed in the target area, and the data traffic demand does not change over time but can be very different across cells. To compute the capacity vector of drone sBSs, a uniform ground-air transmission power of 20dbm is used. In addition, we consider a logarithmic function as the utility function, and we set $\zeta = 0.5$.

The impact of design parameters α and β on the algorithm performance is shown in Fig. 2. The performance ratio of system utility, defined as the ratio of the running average system-wide normalized utility under the MAD-P algorithm and the optimal system-wide normalized utility obtained by the exhaustive search, is plotted with respect to the number of reconfiguration run in the simulation. As it can be seen in Fig. 2a and Fig. 2b, with the increase of β, the performance ratio improves (i.e., the gap to the optimality is reduced) since the time fraction that the system spent in the "better" configurations increases according to (1). In Fig. 2c, it is shown that under the MAD-P/LM algorithm the number of reconfigurations required before convergence can be large when α is small. In other words, as shown by the analytic results (Theorem 2), the convergence of the system-wide normalized utility speeds up as α goes large.

To see how ζ affects the algorithm performance, in the second set of simulations, we consider a scenario where $L=6$, $W=5$, $M=3$, $N=4$, $\alpha=1$, and $\beta=15$, with all other parameters same to those in the first set of simulations. In this particular scenario, since the number of dispatched drone sBSs is close to the number of available channels, but much less than the number of cells in the target area, it will be more rewarding to explore the cells as much as possible so as to identify the most demanding cells very quickly. Fig. 3a and Fig. 3b show that the convergence of the system-wide normalized utility speeds up by selecting a value of ζ that is close to one (i.e., more cell explorations).

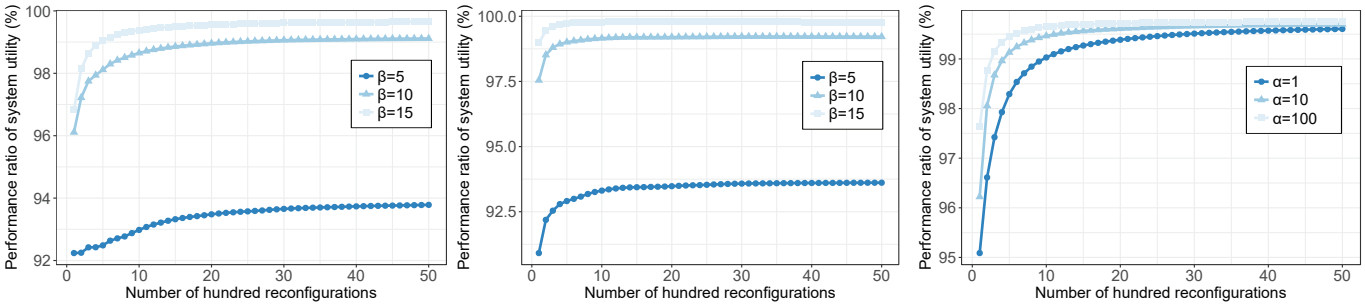
We perform the third set of simulations to evaluate the impact of measurement errors on the algorithm performance, where the setting is the same to that in the first set of simulations with $\alpha=1$ and $\beta=15$. In addition, we introduce a uniformly generated random error when the throughput vector is measured. Fig. 4a and Fig. 4b show the performance ratio of system utility in presence of measurement errors under two algorithms, respectively. As it can be seen, the measurement errors lead to an increase of utility gap. However, from the analytic results (Theorem 3), it is clear that the performance gap is upper bounded.

VII. Conclusion

In this paper, we considered the problem of planning a set of drone small cells operating on multiple channels in a target area to provide access to the backbone/core network. We have formulated the drone cell planning to a combinatorial network utility maximization problem, and then proposed a measurement-aided dynamic planning (MAD-P) algorithm to solve the problem in a distributed fashion during flight with minimal centralized control. We proved that the MAD-P algorithm is asymptotic optimal, and derived an upper bound of the mixing time that captures the speed of convergence of the dynamic planning process. We also derived an upper bound of the performance gap in presence of measurement errors. Simulation results have validated our analytical results and demonstrated the effectiveness of our algorithm.

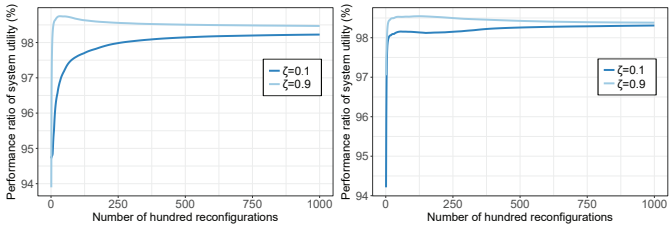
Appendix
Proof of Lemma 1

Denote $\Psi_0 = f^0 = (q_n^0, c_n^0)_{n\in[N]}$ and $\widetilde{\Psi}_0 = \widetilde{f}^0 = (\widetilde{q}_n^0, \widetilde{c}_n^0)_{n\in[N]}$, and we first assume Ψ_0 and $\widetilde{\Psi}_0$ (adjacent states)



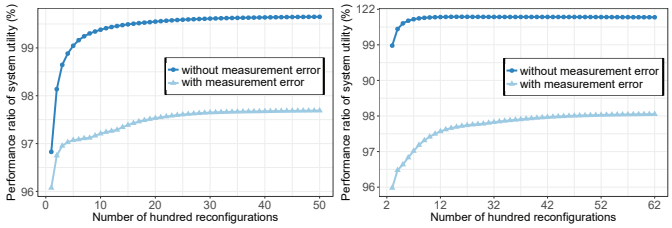
(a) MAD-P/MM, $\alpha = 1$ (b) MAD-P/LM, $\alpha = 1$ (c) MAD-P/LM, $\beta = 15$

Fig. 2: Impact of α and β on the algorithm performance



(a) MAD-P/MM (b) MAD-P/LM

Fig. 3: Impact of ζ on the algorithm performance



(a) MAD-P/MM (b) MAD-P/LM

Fig. 4: Impact of measurement errors on the algorithm performance

only differ by the cell configurations of the same drone sBS, say drone sBS 1 without loss of generality, we first assume Ψ_0 and $\widetilde{\Psi}_0$ (adjacent states) only differ by the cell configurations of drone sBS 1, i.e., $(q_n^0, c_n^0) = (\widetilde{q}_n^0, \widetilde{c}_n^0)$, $\forall n \in [2, N]$, $q_1^0 \neq \widetilde{q}_1^0$ and $c_1^0 = \widetilde{c}_1^0$.

Next, we define the coupling at state $(\Psi_0, \widetilde{\Psi}_0)$ by choosing the next state $(\Psi_1, \widetilde{\Psi}_1)$ according to the following procedure.

Step 1. Choose drone sBS n^* from $[N]$ uniformly at random, and both Ψ and $\widetilde{\Psi}$ update the same n^*.

Step 2. Do the following if $n^* \neq 1$, and go to Step 3 otherwise.

According to the way of constructing the Markov chain Ψ, drone sBS n^* may remain its configuration in the next step or move to a new cell or hop to a new channel in the next step. We define the decision space of n^* conditional on f^0 is $\Omega(n^*|f^0) = \{stay, \mathcal{Q}_{n^*}^{f^0}, \mathcal{C}_{n^*}^{f^0}\}$, and the probability distribution $\boldsymbol{p}_{n^*} = (p_{n^*}(\omega))_{\omega \in \Omega(n^*|f^0)}$, where

$$p_{n^*}(\omega) = \begin{cases} 1 - \frac{\alpha N R \exp(-U(f^0))}{\rho} & \omega = stay, \\ \frac{\alpha \zeta N \exp(-U(f^0))}{\rho} & \omega \in \mathcal{Q}_{n^*}^{f^0}, \\ \frac{\alpha(1-\zeta) N \exp(-U(f^0))}{\rho} & \omega \in \mathcal{C}_{n^*}^{f^0}. \end{cases}$$

Similarly, we define the decision space of n^* conditional on $\widetilde{f}^0$ is $\Omega(n^*|\widetilde{f}^0) = \{stay, \mathcal{Q}_{n^*}^{\widetilde{f}^0}, \mathcal{C}_{n^*}^{\widetilde{f}^0}\}$, and the probability distribution $\widetilde{\boldsymbol{p}}_{n^*} = (\widetilde{p}_{n^*}(\omega))_{\omega \in \Omega(n^*|\widetilde{f}^0)}$, where

$$\widetilde{p}_{n^*}(\omega) = \begin{cases} 1 - \frac{\alpha N R \exp(-U(\widetilde{f}^0))}{\rho} & \omega = stay, \\ \frac{\alpha \zeta N \exp(-U(\widetilde{f}^0))}{\rho} & \omega \in \mathcal{Q}_{n^*}^{\widetilde{f}^0}, \\ \frac{\alpha(1-\zeta) N \exp(-U(\widetilde{f}^0))}{\rho} & \omega \in \mathcal{C}_{n^*}^{\widetilde{f}^0}. \end{cases}$$

Based on $\boldsymbol{p}_{n^*}$ over $\Omega(n^*|f^0)$ and $\widetilde{\boldsymbol{p}}_{n^*}$ over $\Omega(n^*|\widetilde{f}^0)$, we define $\boldsymbol{p}'_{n^*}$ and $\widetilde{\boldsymbol{p}}'_{n^*}$ over the same sample space $\Omega(n^*) = \Omega(n^*|f^0) \cup \Omega(n^*|\widetilde{f}^0) = \{stay, \mathcal{Q}_{n^*}^{f^0} \cup \mathcal{Q}_{n^*}^{\widetilde{f}^0}, \mathcal{C}_{n^*}^{f^0}\}$. Note that $\mathcal{C}_{n^*}^{f^0} = [M] \setminus c_{n^*}^0 = \mathcal{C}_{n^*}^{\widetilde{f}^0} = [M] \setminus \widetilde{c}_{n^*}^0$ since $c_{n^*}^0 = \widetilde{c}_{n^*}^0$. Specifically, we have

$$p'_{n^*}(\omega) = \begin{cases} p_{n^*}(\omega) & \forall \omega \in \Omega(n^*|f^0); \\ 0 & \forall \omega \in \Omega(n^*) \setminus \Omega(n^*|f^0). \end{cases} \tag{10}$$

$$\widetilde{p}'_{n^*}(\omega) = \begin{cases} \widetilde{p}_{n^*}(\omega) & \forall \omega \in \Omega(n^*|\widetilde{f}^0); \\ 0 & \forall \omega \in \Omega(n^*) \setminus \Omega(n^*|\widetilde{f}^0). \end{cases} \tag{11}$$

We further define the following three probability distributions over the sample space $\Omega(n^*)$:

$$\boldsymbol{p}_{n^*}^{\min} = \frac{(\min\{p'_{n^*}(\omega), \widetilde{p}'_{n^*}(\omega)\})_{\omega \in \Omega(n^*)}}{1 - d_{TV}(\boldsymbol{p}'_{n^*}, \widetilde{\boldsymbol{p}}'_{n^*})}, \tag{12}$$

$$\boldsymbol{p}_{n^*}^{+} = \frac{(\max\{0, p'_{n^*}(\omega) - \widetilde{p}'_{n^*}(\omega)\})_{\omega \in \Omega(n^*)}}{d_{TV}(\boldsymbol{p}'_{n^*}, \widetilde{\boldsymbol{p}}'_{n^*})}, \tag{13}$$

$$\widetilde{\boldsymbol{p}}_{n^*}^{+} = \frac{\big(\max\{0, \widetilde{p}'_{n^*}(\omega) - p'_{n^*}(\omega)\}\big)_{\omega \in \Omega(n^*)}}{d_{TV}(\boldsymbol{p}'_{n^*}, \widetilde{\boldsymbol{p}}'_{n^*})}, \tag{14}$$

Recall that $d_{TV}(\cdot, \cdot)$ denotes the total variation distance of two probability distributions.

Now, we are ready to update drone sBSs n^* in both Markov chains Ψ and $\widetilde{\Psi}$ for $(\Psi_1, \widetilde{\Psi}_1)$:

i With probability $1-d_{TV}(\boldsymbol{p}'_{n^*},\widetilde{\boldsymbol{p}}'_{n^*})$, select a configuration ω from $\Omega(n^*)$ according to $\boldsymbol{p}^{\min}_{n^*}$ given in (12), and both Ψ and $\widetilde{\Psi}$ update n^* to the new configuration ω;

ii Otherwise, Ψ and $\widetilde{\Psi}$ update n^* independently. Specifically, Ψ updates n^* according to $\boldsymbol{p}^{+}_{n^*}$ given in (13), and $\widetilde{\Psi}$ updates n^* according to $\widetilde{\boldsymbol{p}}^{+}_{n^*}$ given in (14).

Step 3. Given that $n^*=1$, $q_1^0\neq\widetilde{q}_1^0$, and $c_1^0=\widetilde{c}_1^0$, we define two probability distributions $\boldsymbol{p}'_1$ and $\widetilde{\boldsymbol{p}}'_1$ over the sample space $\Omega(1)=\{(q_1^0,c_1^0),(\widetilde{q}_1^0,c_1^0),\mathcal{Q}_1^{f^0}\cap\mathcal{Q}_1^{\widetilde{f}^0},\mathcal{C}_1^{f^0}\}$. Specifically,

$$p'_1(\omega)=\begin{cases}1-\frac{\alpha NR\exp(-U(f^0))}{\rho} & \omega=(q_1^0,c_1^0),\\ \frac{\alpha\zeta N\exp(-U(f^0))}{\rho} & \omega=(\widetilde{q}_1^0,c_1^0),\\ \frac{\alpha\zeta N\exp(-U(f^0))}{\rho} & \omega\in\mathcal{Q}_1^{f^0}\cap\mathcal{Q}_1^{\widetilde{f}^0},\\ \frac{\alpha(1-\zeta)N\exp(-U(f^0))}{\rho} & \omega\in\mathcal{C}_1^{f^0},\\ 0 & \textit{otherwise.}\end{cases}$$

$$\widetilde{p}'_1(\omega)=\begin{cases}1-\frac{\alpha NR\exp(-U(\widetilde{f}^0))}{\rho} & \omega=(\widetilde{q}_1^0,c_1^0),\\ \frac{\alpha\zeta N\exp(-U(\widetilde{f}^0))}{\rho} & \omega=(q_1^0,c_1^0),\\ \frac{\alpha\zeta N\exp(-U(\widetilde{f}^0))}{\rho} & \omega\in\mathcal{Q}_1^{f^0}\cap\mathcal{Q}_1^{\widetilde{f}^0},\\ \frac{\alpha(1-\zeta)N\exp(-U(\widetilde{f}^0))}{\rho} & \omega\in\mathcal{C}_1^{f^0},\\ 0 & \textit{otherwise.}\end{cases}$$

Similarly, we further define the following three probability distributions over the sample space $\Omega(1)$:

$$\boldsymbol{p}_1^{\min}=\frac{(\min\{p'_1(\omega),\widetilde{p}'_1(\omega)\})_{\omega\in\Omega(1)}}{1-d_{TV}(\boldsymbol{p}'_1,\widetilde{\boldsymbol{p}}'_1)},\tag{15}$$

$$\boldsymbol{p}_1^{+}=\frac{(\max\{0,p'_1(\omega)-\widetilde{p}'_1(\omega)\})_{\omega\in\Omega(1)}}{d_{TV}(\boldsymbol{p}'_1,\widetilde{\boldsymbol{p}}'_1)},\tag{16}$$

$$\widetilde{\boldsymbol{p}}_1^{+}=\frac{\big(\max\{0,\widetilde{p}'_1(\omega)-p'_1(\omega)\}\big)_{\omega\in\Omega(1)}}{d_{TV}(\boldsymbol{p}'_1,\widetilde{\boldsymbol{p}}'_1)},\tag{17}$$

Now, we are ready to update drone sBSs n^* in both Markov chains Ψ and $\widetilde{\Psi}$ for $(\Psi_1,\widetilde{\Psi}_1)$:

i With probability $1-d_{TV}(\boldsymbol{p}'_1,\widetilde{\boldsymbol{p}}'_1)$, select a configuration ω from $\Omega(1)$ according to $\boldsymbol{p}_1^{\min}$ given in (15), and both Ψ and $\widetilde{\Psi}$ update drone sBS 1 to the new configuration ω;

ii Otherwise, Ψ and $\widetilde{\Psi}$ update drone sBS 1 independently. Specifically, Ψ updates drone sBS 1 according to $\boldsymbol{p}_1^{+}$ given in (16), and $\widetilde{\Psi}$ updates drone sBS 1 according to $\widetilde{\boldsymbol{p}}_1^{+}$ given in (17).

It is not hard to show that the above procedure leads to a valid coupling. Initially, we have $H(\Psi_0,\widetilde{\Psi}_0)=1$. With path coupling, we next show that the Hamming distance between Ψ_1 and $\widetilde{\Psi}_1$ decreases in expectation compared to $H(\Psi_0,\widetilde{\Psi}_0)$, i.e., $\mathbb{E}[H(\Psi_1,\widetilde{\Psi}_1)|\Psi_0,\widetilde{\Psi}_0]<H(\Psi_0,\widetilde{\Psi}_0)$.

By the above coupling procedure, it can be seen that when $n^*=1$, we have $\Psi_1=\widetilde{\Psi}_1$ if drone sBS 1 remains its configuration from Ψ_0 to Ψ_1 and moves from $\widetilde{q}_1^0$ ($\widetilde{\Psi}_0$) to q_1^0 ($\widetilde{\Psi}_1$)); or drone sBS 1 remains its configuration from $\widetilde{\Psi}_0$ to $\widetilde{\Psi}_1$ and moves from q_1^0 (Ψ_0) to $\widetilde{q}_1^0$ (Ψ_1); or both chains update drone sBS 1 to a new cell in $\mathcal{Q}_1^{f^0}\cap\mathcal{Q}_1^{\widetilde{f}^0}$. Note that $|\mathcal{Q}_1^{f^0}\cap\mathcal{Q}_1^{\widetilde{f}^0}|=LW-N-1$. Hence, we have

$$\begin{aligned}&\mathbb{E}\Big[H(\Psi_1,\widetilde{\Psi}_1)-1|\Psi_0,\widetilde{\Psi}_0,n^*=1\Big]\\&=-\sum_{\omega\in\Omega(1)\setminus\mathcal{C}_1^{f^0}}\Big(1-d_{TV}(\boldsymbol{p}'_1,\widetilde{\boldsymbol{p}}'_1)\Big)p_1^{\min}(\omega)\\&=-\sum_{\omega\in\Omega(1)\setminus\mathcal{C}_1^{f^0}}\min\{p'_1(\omega),\widetilde{p}'_1(\omega)\}\\&\leq-\frac{\zeta(LW-N+1)\exp\left(-\beta(\Theta_{\max}-\Theta_{\min})\right)}{R|\mathcal{F}|}.\end{aligned}\tag{18}$$

When $n^*\in[2,N]$, with probability $1-d_{TV}(\boldsymbol{p}'_{n^*},\widetilde{\boldsymbol{p}}'_{n^*})$, $H(\Psi_1,\widetilde{\Psi}_1)=1$; otherwise, $H(\Psi_1,\widetilde{\Psi}_1)=2$. Then, we have,

$$\begin{aligned}&\mathbb{E}\Big[H(\Psi_1,\widetilde{\Psi}_1)-1|\Psi_0,\widetilde{\Psi}_0,n^*\in[2,N]\Big]\\&=d_{TV}(\boldsymbol{p}'_{n^*},\widetilde{\boldsymbol{p}}'_{n^*})=1-\sum_{\omega\in\Omega(n^*)}\min\{p'_{n^*}(\omega),\widetilde{p}'_{n^*}(\omega)\}.\end{aligned}\tag{19}$$

From the MAD-P/MM algorithm, it can be obtained that for any $n^*\in[2,N]$, $|\mathcal{Q}_{n^*}^{f^0}|=|\mathcal{Q}_{n^*}^{\widetilde{f}^0}|=LW-N$ and $|\mathcal{Q}_{n^*}^{f^0}\cap\mathcal{Q}_{n^*}^{\widetilde{f}^0}|=LW-N-1$. Therefore,

$$\begin{aligned}&\sum_{\omega\in\Omega(n^*)}\min\{p'_{n^*}(\omega),\widetilde{p}'_{n^*}(\omega)\}\\&\geq1-\frac{\exp\left(\beta(\Theta_{\max}-\Theta_{\min})\right)}{|\mathcal{F}|}\\&\quad+\frac{\zeta(LW-N-1)\exp\left(-\beta(\Theta_{\max}-\Theta_{\min})\right)}{R|\mathcal{F}|}\\&\quad+\frac{(1-\zeta)(M-1)\exp\left(-\beta(\Theta_{\max}-\Theta_{\min})\right)}{R|\mathcal{F}|}\\&=1-\frac{\exp\left(-\beta(\Theta_{\max}-\Theta_{\min})\right)}{|\mathcal{F}|}\\&\quad\cdot\left(\exp\left(2\beta(\Theta_{\max}-\Theta_{\min})\right)-\frac{R-\zeta}{R}\right)\end{aligned}\tag{20}$$

Here, we assume that

$$\exp\left(\beta(\Theta_{\max}-\Theta_{\min})\right)<|\mathcal{F}|,\tag{21}$$

since $1-\frac{\exp(\beta(\Theta_{\max}-\Theta_{\min}))}{|\mathcal{F}|}$ has to be a valid probability. However, we will immediately have (21) after we bound β shortly. By plugging in (18), (19), and (20), we bound the conditional expected Hamming distance between Ψ_1 and $\widetilde{\Psi}_1$:

$$\begin{aligned}&\mathbb{E}\Big[H(\Psi_1,\widetilde{\Psi}_1)-1|\Psi_0,\widetilde{\Psi}_0\Big]\\&=\sum_{n\in[N]}\frac{1}{N}\mathbb{E}\Big[H(\Psi_1,\widetilde{\Psi}_1)-1|\Psi_0,\widetilde{\Psi}_0,n^*=n\Big]\\&\leq\frac{1}{N}\cdot\frac{1}{|\mathcal{F}|}(N-1)\exp\left(-\beta(\Theta_{\max}-\Theta_{\min})\right)\\&\quad\cdot\left[\exp\left(2\beta(\Theta_{\max}-\Theta_{\min})\right)-\left(1+\frac{\zeta(LW-2N+2)}{(N-1)R}\right)\right].\end{aligned}\tag{22}$$

We further denote the right-hand side of (22) by $-B_1$, i.e.,

$$B_1\triangleq\frac{(N-1)\left[1+\frac{\zeta(LW-2N+2)}{(N-1)R}-\exp\left(2\beta(\Theta_{\max}-\Theta_{\min})\right)\right]}{N|\mathcal{F}|\exp\left(\beta(\Theta_{\max}-\Theta_{\min})\right)},$$

which is positive if the following two conditions hold:

$$N < \frac{LW}{2} + 1 \tag{23}$$

$$0 < \beta < \frac{1}{2(\Theta_{\max} - \Theta_{\min})} \ln\left(1 + \frac{\zeta(LW - 2N + 2)}{(N-1)R}\right). \tag{24}$$

Due to (24), we certainly have (21). Therefore, for any pair of Ψ_0 and $\widetilde{\Psi}_0$ (adjacent states) only differing by the cell configurations of the same drone sBS, it follows that

$$\mathbb{E}\left[H(\Psi_1, \widetilde{\Psi}_1)|\Psi_0, \widetilde{\Psi}_0\right] \leq 1 - B_1 = (1 - B_1)H(\Psi_0, \widetilde{\Psi}_0),$$

where $0 < B_1 < 1$.

Follow the above analysis, similarly, for any pair of Ψ_0 and $\widetilde{\Psi}_0$ (adjacent states) only differing by the channel configurations of the same drone sBS, we have

$$\mathbb{E}\left[H(\Psi_1, \widetilde{\Psi}_1)|\Psi_0, \widetilde{\Psi}_0\right] \leq 1 - B_2 = (1 - B_2)H(\Psi_0, \widetilde{\Psi}_0),$$

where

$$B_2 \triangleq \frac{(N-1)\left[1 + \frac{(1-\zeta)M}{(N-1)R} - \exp\left(2\beta(\Theta_{\max} - \Theta_{\min})\right)\right]}{N|\mathcal{F}|\exp\left(\beta(\Theta_{\max} - \Theta_{\min})\right)},$$

which is positive when

$$0 < \beta < \frac{1}{2(\Theta_{\max} - \Theta_{\min})} \ln\left(1 + \frac{(1-\zeta)M}{(N-1)R}\right). \tag{25}$$

Now, we choose any β which satisfies

$$0 < \beta < \frac{\ln(1+\kappa)}{2(\Theta_{\max} - \Theta_{\min})}, \tag{26}$$

where $\kappa \triangleq \min\left\{\frac{\zeta(LW-2N+2)}{(N-1)R}, \frac{(1-\zeta)M}{(N-1)R}\right\}$. We further define $B \triangleq \min\{B_1, B_2\}$. From the path coupling theorem in [37], for any $(\Psi_0, \widetilde{\Psi}_0) \in \mathcal{F} \times \mathcal{F}$, we have

$$\mathbb{E}\left[H(\Psi_1, \widetilde{\Psi}_1)|\Psi_0, \widetilde{\Psi}_0\right] \leq (1 - B)H(\Psi_0, \widetilde{\Psi}_0), \tag{27}$$

where $0 < B < 1$. Thus, Lemma 1 follows.

ACKNOWLEDGMENT

This work was supported by NSERC of Canada, Henan International Science & Technology Cooperation Program (182102410050), Henan Young Scholar Promotion Program (2016GGJS-018), the Program for Science & Technology Development of Henan Province (162102210022), and the Key Project of Science & Technology Research of the Education Department of Henan Province (17A413001).

REFERENCES

[1] Y. Zhou, N. Cheng, N. Lu, and X. S. Shen, "Multi-UAV-aided networks: aerial-ground cooperative vehicular networking architecture," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 36–44, 2015.

[2] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[3] Z. Xu, L. Chen, C. Chen, and X. Guan, "Joint clustering and routing design for reliable and efficient data collection in large-scale wireless sensor networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 520–532, 2016.

[4] Y. Li and L. Cai, "UAV-assisted dynamic coverage in a heterogeneous cellular system," *IEEE Network*, vol. 31, no. 4, pp. 56–61, 2017.

[5] Y. Li, G. Feng, M. Ghasemiahmadi, and L. Cai, "Power allocation and 3-D placement for floating relay supporting indoor communications," *IEEE Transactions on Mobile Computing*, 2018.

[6] Y. Yang, Z. Zheng, K. Bian, L. Song, and Z. Han, "Real-time profiling of fine-grained air quality index distribution using UAV sensing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 186–198, 2018.

[7] W. Guo, C. Devine, and S. Wang, "Performance analysis of micro unmanned airborne communication relays for cellular networks," in *Proc. IEEE International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2014, pp. 658–663.

[8] Y. Ma, N. Selby, and F. Adib, "Drone relays for battery-free networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 335–347.

[9] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, 2016.

[10] J. Laiho, A. Wacker, and T. Novosad, *Radio network planning and optimisation for UMTS*. John Wiley & Sons, 2006.

[11] T. Q. Quek, *Small cell networks: Deployment, PHY techniques, and resource management*. Cambridge University Press, 2013.

[12] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *arXiv preprint arXiv:1803.00680*, 2018.

[13] X. Li, J. B. Rao, and H. Zhang, "Engineering machine-to-machine traffic in 5G," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 609–618, 2016.

[14] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou *et al.*, "Air-ground integrated mobile edge networks: Architecture, challenges and opportunities," *IEEE Communications Magazine*, to appear.

[15] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

[16] O. Semiari, W. Saad, M. Bennis, and Z. Dawy, "Inter-operator resource management for millimeter wave multi-hop backhaul networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5258–5272, 2017.

[17] W. Wu, Q. Shen, K. Aldubaikhy, N. Cheng, N. Zhang, and X. S. Shen, "Enhance the edge with beamforming: Performance analysis of beamforming-enabled WLAN," in *Proc. IEEE WiOpt*, Shanghai, China, December 2018.

[18] V. Garcia, Y. Zhou, and J. Shi, "Coordinated multipoint transmission in dense cellular networks with user-centric adaptive clustering," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4297–4308, 2014.

[19] M. Casoni, C. A. Grazia, M. Klapez, N. Patriciello, A. Amditis, and E. Sdongos, "Integration of satellite and LTE for disaster recovery," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 47–53, 2015.

[20] M. Alzenad, M. Z. Shakir, H. Yanikomeroglu, and M.-S. Alouini, "Fso-based vertical backhaul/fronthaul framework for 5G+ wireless networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 218–224, 2018.

[21] A. Alsharoa, H. Ghazzai, A. Kadri, and A. E. Kamal, "Energy management in cellular hetnets assisted by solar powered drone small cells," in *Proc. IEEE WCNC*, San Francisco, CA, USA, March 2017.

[22] T. J. Nugent and J. T. Kare, "Laser power for UAVs," *Laser Motive White Paper-Power Beaming for UAVs, NWEN*, 2010.

[23] S. Aldhaher, P. D. Mitcheson, J. M. Arteaga, G. Kkelis, and D. C. Yates, "Light-weight wireless power transfer for mid-air charging of drones," in *Proc. The 11th European Conference on Antennas and Propagation (EUCAP)*, Paris, France, March 2017.

[24] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE Globecom*, San Diego, CA, USA, December 2015.

[25] S.-F. Chou, Y.-J. Yu, and A.-C. Pang, "Mobile small cell deployment for service time maximization over next-generation cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5398–5408, 2017.

[26] A. Fotouhi, M. Ding, and M. Hassan, "Dynamic base station repositioning to improve spectral efficiency of drone small cells," in *Proc. IEEE WoWMoM*, Macao, China, June 2017.

[27] J. Lu, S. Wan, X. Chen, Z. Chena, P. Fan, and K. B. Letaief, "Beyond empirical models: Pattern formation driven placement of UAV base stations," *IEEE Transactions on Wireless Communications*, 2018.

[28] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks," *IEEE Network*, vol. 30, no. 1, pp. 100–105, 2016.

[29] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[30] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anti-coordination game based partially overlapping channels assignment in combined UAV and D2D-based networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1672–1683, 2018.

[31] D. Takaishi, Y. Kawamoto, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Virtual cell based resource allocation for efficient frequency utilization in unmanned aircraft systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3495–3504, 2018.

[32] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6301–6327, 2013.

[33] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[34] F. P. Kelly, *Reversibility and stochastic networks*. Cambridge University Press, 2011.

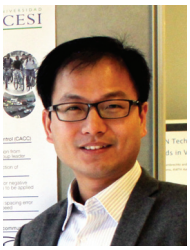
[35] J. Ghaderi and R. Srikant, "On the design of efficient CSMA algorithms for wireless networks," in *Proc. of IEEE Conference on Decision and Control (CDC)*, Atlanta, Georgia, USA, December 2010.

[36] W. Whitt, "Continuous-Time Markov Chains," http://www.columbia.edu/~ww2040/6711F13/CTMCnotes120413.pdf, 2012, [Online].

[37] R. Bubley and M. Dyer, "Path coupling: A technique for proving rapid mixing in Markov chains," in *Proc. IEEE Annual Symposium on Foundations of Computer Science*, 1997, pp. 223–231.

[38] S. Zhang, Z. Shao, and M. Chen, "Optimal distributed p2p streaming under node degree bounds," in *Proc. IEEE ICNP*, 2010, pp. 253–262.

Ning Lu (S'12-M'15) received the B.E. (2007) and M.E. (2010) degrees from Tongji University, Shanghai, China, and Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, all in electrical engineering. He is currently an assistant professor in the Department of Computing Science at Thompson Rivers University, Canada. Previously, he was a postdoctoral fellow with the Coordinated Science Laboratory in the University of Illinois at Urbana-Champaign. He also spent the summer of 2009 as an intern in the National Institute of Informatics, Tokyo, Japan. His current research interests include real-time scheduling, distributed algorithms, and reinforcement learning for wireless networks.



Yi Zhou (M'15) received the B.S. degree in electronic engineering from the First Aeronautic Institute of Air Force, China, in 2002, and the Ph.D. degree in control system and theory from Tongji University, China, in 2011. From 2002 to 2005, he was a Research and Development Engineer with Beijing Centergate Technologies Co, Ltd., China. He was a Visiting Researcher with the Telecommunications Research Center (ftw.), Vienna, Austria, in 2009, and the National Institute of Informatics, Tokyo, Japan, in 2010. From 2014 to 2015, he was a Post-Doctoral Fellow with the University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the School of Computer and Information Engineering, Henan University, Kaifeng, China. He is the Director of the International Joint Research Laboratory for Cooperative Vehicular Networks, Henan, China. He also leads the Vehicular Networking Institute of Central Plains Silicon Valley. His research interests include vehicular cyber-physical systems and multi-agent design for vehicular networks.

Chenhao Shi received the M. Eng. degree in control theory and control engineering from Henan University, China, in 2018. His research interests include heterogeneous vehicular networking, air-ground cooperation and the application of machine learning algorithms in vehicular networks. Currently, he is devoted to apply deep learning algorithms into Intelligent Transportation System (ITS).

Nan Cheng (S'12-M'16) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo in 2016, and B.E. degree and the M.S. degree from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively. He is currently a professor with School of Telecommunication Engineering, Xidian University, Shaanxi, China. He worked as a Post-doctoral fellow with the Department of Electrical and Computer Engineering, University of Toronto and Department of Electrical and Computer Engineering, University of Waterloo under the supervision of Prof. Ben Liang and Prof. Sherman (Xuemin) Shen, from 2017 to 2018. His current research focuses on space-air-ground integrated system, big data in vehicular networks, and self-driving system. His research interests also include performance analysis, MAC, opportunistic communication, and application of AI for vehicular networks.

Lin Cai (S'00-M'06-SM'10) received her M.A.Sc. and PhD degrees in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical & Computer Engineering at the University of Victoria, and she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and Internet of Things.

She was a recipient of the NSERC Discovery Accelerator Supplement (DAS) Grants in 2010 and 2015, respectively, and the Best Paper Awards of IEEE ICC 2008 and IEEE WCNC 2011. She has founded and chaired IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has served as a member of the Steering Committee of the IEEE Transactions on Big Data, an Associate Editor of the IEEE Internet of Things Journal, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, EURASIP Journal on Wireless Communications and Networking, International Journal of Sensor Networks, and Journal of Communications and Networks (JCN), and as the Distinguished Lecturer of the IEEE VTS Society. She has served as a TPC symposium co-chair for IEEE Globecom'10 and Globecom'13. She is a registered professional engineer of British Columbia, Canada.

Bin Li (S'12-M'16) received his B.S. degree in Electronic and Information Engineering in 2005, M.S. degree in Communication and Information Engineering in 2008, both from Xiamen University, China, and Ph.D. degree in Electrical and Computer Engineering from The Ohio State University in 2014. Between 2014 and 2016, he worked as a Postdoctoral Researcher in the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign. He is currently an Assistant Professor in the Department of Electrical, Computer, and Biomedical Engineering at the University of Rhode Island.

His research interests include communication networks, virtual/augmented reality, fog computing, data centers, resource allocation and management, distributed algorithm design, queueing theory, and optimization theory.