Distributed Approaches for Inter-cell Interference Coordination in UAV-based LTE-Advanced HetNets

Simran Singh[1], Abhaykumar Kumbhar[2], İsmail Güvenç[1], and Mihail L Sichitiu[1]
[1]Dept. Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 27606
[2]Dept. Electrical and Computer Engineering, Florida International University, Miami, FL, 33174

***Abstract*—Unmanned aerial vehicles (UAVs) can supplement the existing ground-based HetNets by replacing/supporting damaged infrastructure, providing real-time video support at the site of an emergency, offloading traffic in congested areas, extending coverage, and filling coverage gaps. However, it is challenging to tune the parameters of these UAV-HetNets to maximize network performance and spectral efficiency. In this paper, we introduce distributed algorithms that leverage UAV mobility, enhanced inter-cell interference coordination (ICIC), and cell range expansion (CRE) techniques defined in 3GPP Release-10 and 3GPP Release-11. Through Monte-Carlo simulations, we compare the system-wide 5th percentile spectral efficiency (5pSE) while optimizing the performance using a brute force algorithm, a heuristic-based sequential algorithm, and a deep Q-learning algorithm. The autonomous UAVs jointly optimize their location, ICIC parameters, and CRE to maximize 5pSE gains and minimize the outage probability. Our results show that the ICIC technique relying on a simple heuristic outperforms the ICIC technique based on deep Q-learning. Taking advantage of the multiple optimization parameters for interference coordination, the heuristic-based ICIC technique can achieve 5pSE values that are reasonably close to those achieved with exhaustive brute force search techniques, at a significantly lower computational complexity.**

***Index Terms*—Deep Q learning, drone, FeICIC, HetNets, interference management, LTE-Advanced, range expansion, UAV.**

I. Introduction

A heterogeneous cellular network (HetNet) comprises ground-based fixed macro base stations (MBS) and small cells such as ground-based base station, unmanned aerial vehicles (UAVs), and cells on wheels [1]. Due to the mobile and agile nature of UAVs, they have found application for public safety communication during an emergency by restoring damaged infrastructure, addressing network congestion, and handling high traffic during a large public event such as music concerts and sports events.UAVs were used as base stations practically for the first time in Puerto Rico in November 2017: after the ground base stations were destroyed by hurricane Maria in 2017, AT&T used UAVs to temporarily restore wireless voice, text, data, and multimedia services [2].

The concept of a UAV HetNet is depicted in Fig. 1, which also shows how adding unmanned aerial base stations (UABSs) can complement existing MBSs by serving user equipments (UEs) that are far away from the MBS. While UABSs give their connected UEs a larger share of the spectrum, UABSs also cause interference to the UEs served by the MBSs. Research problems in UAV HetNets include deciding the best position and trajectory of the UABSs to optimize various performance metrics, reducing interference between the UABSs and MBSs, managing handovers, adapting to the UE mobility, and adapting to the changes in the demand for cellular service. UAV HetNets also have their own unique constraints. As UABSs operate on battery and have limited computational capabilities, the solutions should be computationally feasible and energy efficient. The solutions should also have low time complexity to allow the UABS to respond quickly to changes in demand. Further, solutions that involve communication between the UABSs and MBSs should consider the impact of the overhead.

This work has been supported in part by NSF under the grant numbers CNS-1453678 and CNS-1738093.

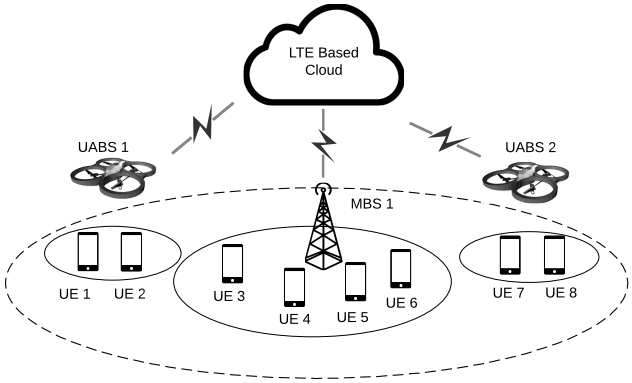

Fig. 1. A UAV assisted HetNet composed of UABSs and an MBS, where certain UEs are offloaded to UABSs for improving coverage and fairness.

The integration of UAVs with cellular networks has been attracting the attention of researchers. While [3] and [4] provided interference reduction techniques to calculate the best UABS position and trajectory, they did not leverage LTE 3GPP interference management techniques. Novel approaches to optimally partition a geographical area into UABS and MBS cells were proposed in [5] and [6], using optimal transport theory and neural-based cost functions, respectively, without explicitly tackling interference mitigation challenges. Interference mitigation using 3GPP Release-10 enhanced inter-cell interference coordination techniques (eICIC) and Release-11 further enhanced inter-cell interference coordination (FeICIC) techniques in HetNets were studied in [7]–[10]. While [7] jointly optimized the ICIC parameters, UE cell association rules, and the spectrum resource sharing between the macro and pico cells, it did not use 3GPP Release-11 FeICIC nor cell range expansion (CRE) techniques. Moreover, [7] only studied LTE HetNets and not UAV HetNets. In [8], the authors developed a stochastic geometry based framework to study and compare the effectiveness of 3GPP FeICIC techniques and eICIC techniques, but [8] also did not study UAV HetNets. The use of 3GPP Release-10/11 techniques along with UABS mobility in UAV HetNets was evaluated in [10]. However, this study did not individually optimize the 3GPP ICIC parameters, but rather applied the same ICIC parameter values to each MBS and UABS, which is suboptimal, as we will demonstrate later in this paper. To the best of our knowledge, distributed and low complexity approaches to optimize 3GPP Release-10/11 interference management parameters in UAV HetNets have not been studied in literature.

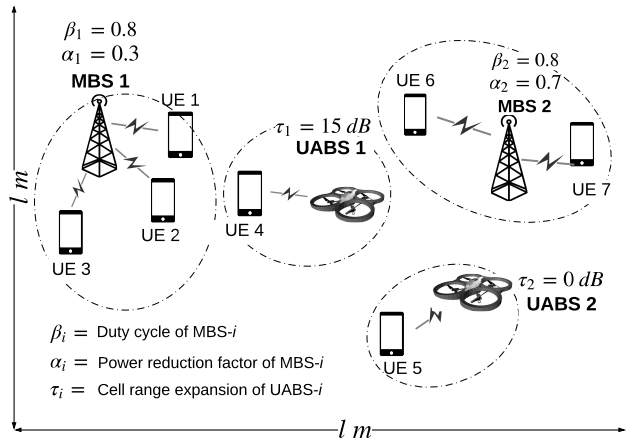

Fig. 2. Illustration of a UAV HetNet where the ICIC parameters of all MBSs and UABSs, and location of UABSs are optimized individually. This approach is in contrast to some existing work, such as [10], which, while optimizing locations for each UABS individually, assigns the same ICIC parameters to all UABSs and MBSs.

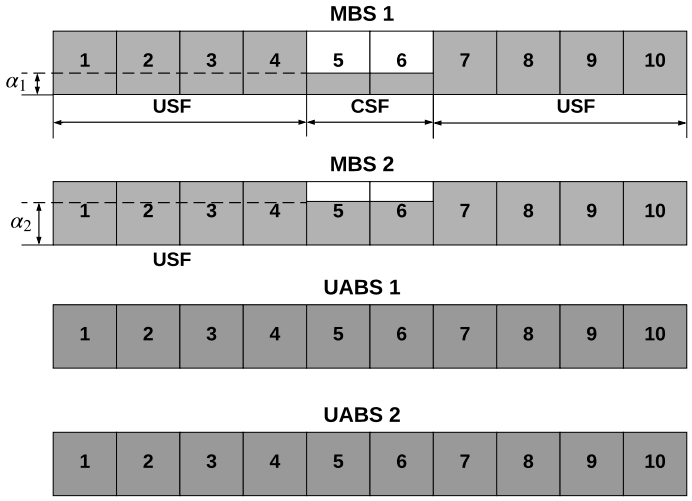

Fig. 3. 3GPP Release-11 FeICIC with reduced power almost blank subframes, and different power reduction factors for each MBS, for the scenario in Fig. 2

Our contribution is that we propose a custom sequential algorithm and a deep Q learning based algorithm, to individually optimize 3GPP Release-10/11 interference coordination parameters and UABS position in order to maximize the fifth percentile spectral efficiency (5pSE). Taking into account the constraints of UAV-Hetnets, we attempt to minimize the computational and time complexity. We also compare these two computationally efficient algorithms with an optimal but computationally complex brute force algorithm.

The rest of this paper is organized as follows. Section II presents the system model and 3GPP Release 10/11 interference management techniques, and sheds light on the large size of our parameter space, hence motivating the use of computationally simpler, heuristic based approaches. Section III explains brute force and sequential algorithms for optimizing UABS positions and 3GPP Release-10/11 parameters, while Section IV presents the proposed machine learning techniques. Section V presents our results and finally Section VI concludes our paper.

II. System Model

We consider a two tier HetNet with MBSs and UABSs within a simulation area of $l \times l$ square meters as shown in Fig. 2. We assume that MBSs and UEs are randomly distributed in this area using a Poisson point process with intensities λ_{mbs} and λ_{ue}, respectively. The number of MBSs and UEs in the simulation are given by $N_{\text{mbs}} = \lambda_{\text{mbs}} \times l^2$ and $N_{\text{ue}} = \lambda_{\text{ue}} \times l^2$, respectively. The 3D locations of all the MBSs, UABSs, and UEs are captured in the matrices $\mathbf{X}_{\text{mbs}} \in \mathbb{R}^{N_{\text{mbs}} \times 3}$, $\mathbf{X}_{\text{uabs}} \in \mathbb{R}^{N_{\text{uabs}} \times 3}$ and $\mathbf{X}_{\text{ue}} \in \mathbb{R}^{N_{\text{ue}} \times 3}$, respectively. We assume that all MBSs transmit at a power of P_{mbs} and the UABSs transmit at a power of P_{uabs}. We assume antenna gains of the MBS and UABS to be K and K', respectively. Therefore, the effective transmit power for the MBS and UABS are $P'_{\text{mbs}} = KP_{\text{mbs}}$ and $P'_{\text{uabs}} = K'P_{\text{uabs}}$, respectively.

We assume that the UABSs and MBSs exchange information over the X2 interfaces. Furthermore, we consider that, the bandwidth available to an MBS or a UABS, in the downlink, is divided equally amongst all the UEs served by that MBS or UABS. The downlink data is always assumed to be present for the UE, i.e., the downlink UE traffic buffers are always full. For an arbitrary UE n, where $n \in \{1, 2, ..., N_{\text{ue}}\}$, we define the macro-cell of interest (MOI) as the nearest MBS, and the UAV-cell of interest (UOI) as the nearest UABS. For example, in the specific scenario shown in Fig 2, MBS 1 is the MOI, and UABS 1 is the UOI of UE 4. MBS 1 is also the MOI for UEs 1, 2 and 3, while UABS 1 is the UOI for UEs 1, 2, 3, 4, 6 and 7. We denote the reference symbol received power (RSRP) of the nth UE from the MOI and the UOI as $S_{\text{mbs}}(d_{mn})$ and $S_{\text{uabs}}(d_{un})$, respectively, where d_{mn} is the distance from the nearest MOI, and d_{un} is the distance from the nearest UOI for the nth UE. In this system model, we use Okumura Hata suburban propagation model without any Rayleigh or Rician fading.

A. Spectral Efficiency with 3GPP Rel.-10/11 ICIC Techniques

In a HetNet, the MBSs transmit at higher powers and have higher ranges compared to the lower power UABSs. Nevertheless, the UABSs can extend their coverage and associate a larger number of UEs by using the cell range expansion (CRE) technique defined in 3GPP Release-8. The CRE of a UABS is defined as the factor by which UEs are biased to associate with that UABS. For example, in Fig. 2, UABS 1 uses a CRE of 15 dB to force UE 4 to associate with itself. The use of CRE, however, results in increased interference to those UEs in the extended cell regions.

This interference from MBSs to UEs near the edge of range-extended UABS cells can be mitigated using time-domain based ICIC techniques defined in 3GPP Release-10/11. These techniques require the MBS to transmit with reduced power during specific subframes on the physical downlink shared channel (PDSCH). Radio subframes with reduced power are termed coordinated subframes (CSF), and those with full power are termed uncoordinated subframes (USF). We denote this power reduction factor by α where $0 \leq \alpha \leq 1$. We note that $\alpha = 1$ implies no ICIC, while eICIC techniques use $\alpha = 0$, and FeICIC techniques allow α to vary between 0 and 1. We use β to denote the USF duty cycle, and hence, the CSF duty cycle is given by $(1 - \beta)$. Fig. 3 shows, for the scenario depicted in 2, how MBS 1 and MBS 2 use power reduction factors, α_1 and α_2 respectively, to reduce interference to UE 4. We note that $\alpha_1 < \alpha_2$, as MBS 2, being farther away from UE 4, can transmit at a higher power without degrading the performance of UE 4.

Individual MBSs or UABSs can schedule their UEs in USF or CSF based on scheduling thresholds ρ and ρ', respectively.

Then, a UE may be served either by an MOI or UOI, and by the CSF or USF resources of the MOI/UOI, resulting in four different association categories. Let Γ denote the signal to interference ratio (SIR) at the MOI-USF, Γ' denote the SIR at the UOI-USF, and τ denote the CRE that positively biases the UABS SIR to expand its coverage. Then, the four different resource categories where a UE may be scheduled can be summarized as follows. If $\Gamma > \tau\Gamma'$, we associate the UE with the MOI, else we schedule it with the UOI. The intuition behind this condition is straightforward: associate the UE with the nearest base station that gives the best SIR, taking into account CRE. On the other hand, if $\Gamma > \rho$ or $\Gamma' \leq \rho'$, we schedule the UE in CSF, and otherwise in USF. This condition is based on the following intuition: scheduling a UE in the CSF of an MOI degrades that UE's SIR, whereas scheduling a UE in the CSF of a UOI improves that UE's SIR. So, the "stronger" UEs that have sufficiently high SIRs and are close to the MBS should be scheduled in the CSFs of that MBS, as these "stronger" UEs can take the performance hit. Similarly, the "weaker" UEs that have low SIRs and are close to the cell edge of a UABS, should be scheduled in the CSF of that UABS, as they need to be protected on priority.

Using this framework of eICIC and FeICIC and following an approach similar to that of [8], [10] for a HetNet, the SIR (Γ, Γ_{csf}, Γ', Γ'_{csf}) and the SE ($C^{\text{mbs}}_{\text{usf}}$, $C^{\text{mbs}}_{\text{csf}}$, $C^{\text{uabs}}_{\text{usf}}$, $C^{\text{uabs}}_{\text{csf}}$) experienced by an arbitrary UE n can be defined for four different scenarios as follows:

1) ***UE associated with MOI and scheduled in USF:***

$$\Gamma = \frac{S_{\text{mbs}}(d_{mn})}{S_{\text{uabs}}(d_{un}) + Z} \qquad C^{\text{mbs}}_{\text{usf}} = \frac{\beta \log_2(1+\Gamma)}{N^{\text{mbs}}_{\text{usf}}} \ . \tag{1}$$

2) ***UE associated with MOI and scheduled in CSF:***

$$\Gamma_{\text{csf}} = \frac{\alpha S_{\text{mbs}}(d_{mn})}{S_{\text{uabs}}(d_{un}) + Z} \quad C^{\text{mbs}}_{\text{csf}} = \frac{(1-\beta)\log_2(1+\Gamma_{\text{csf}})}{N^{\text{mbs}}_{\text{csf}}}. \tag{2}$$

3) ***UE associated with UOI and scheduled in USF:***

$$\Gamma' = \frac{S_{\text{uabs}}(d_{un})}{S_{\text{mbs}}(d_{mn}) + Z'} \qquad C^{\text{uabs}}_{\text{usf}} = \frac{\beta \log_2(1+\Gamma')}{N^{\text{uabs}}_{\text{usf}}}. \tag{3}$$

4) ***UE camped on UOI and scheduled in CSF:***

$$\Gamma'_{\text{csf}} = \frac{S_{\text{uabs}}(d_{un})}{\boldsymbol{\alpha} S_{\text{mbs}}(d_{mn}) + Z'} \quad C^{\text{uabs}}_{\text{csf}} = \frac{(1-\beta)\log_2(1+\Gamma'_{\text{csf}})}{N^{\text{uabs}}_{\text{csf}}}. \tag{4}$$

In (1)-(4), Z and Z' are, respectively, the interference at a UE from all the MBSs and the UABSs except the UOI, and except the MOI. They have the same meaning, but different values.

B. Performance Metric

Our chosen performance metric is the 5pSE, which represents the performance of the UEs that are performing poorly, e.g. UEs at the cell edge. 5pSE is also one of the metrics used by ITU to specify the performance requirements of IMT-2020 (5G).

The objective of our study is to design algorithms to calculate the best FeICIC and eICIC parameters individually for all MBSs and UABSs, and the positions of the UABSs, so as to maximize our performance metric, the 5pSE. The algorithm should find the best state, $\mathbf{S}'$, out of all possible states $\mathbf{S}$ such that:

$$\mathbf{S}' = \arg\max_{\mathbf{S}} \ \mathbf{C}_{5\text{th}}(\mathbf{S}), \tag{5}$$

TABLE I. Size of the multi dimensional parameter space

Parameter	Range	Search space size	Applicable to
α	$0, \Delta_\alpha, 2\Delta_\alpha, ...1$	$1/\Delta_\alpha + 1$	Each MBS
β	$0, \Delta_\beta, 2\Delta_\beta, ...1$	$1/\Delta_\beta + 1$	Each MBS
ρ	$\rho_{\text{low}}, \rho_{\text{low}}+\Delta_\rho, \rho_{\text{low}}+2\Delta_\rho ... \rho_{\text{high}}$	$\frac{(\rho_{\text{high}}-\rho_{\text{low}})}{(\Delta_\rho)}$	Each MBS
ρ'	$\rho'_{\text{low}}, \rho'_{\text{low}}+\Delta'_\rho, \rho'_{\text{low}}+2\Delta_{\rho'} ... \rho'_{\text{high}}$	$\frac{(\rho'_{\text{high}}-\rho'_{\text{low}})}{(\Delta'_\rho)}$	Each UABS
τ	$0, \Delta_\tau, 2\Delta_\tau, ...\tau_{\text{high}}$	$\frac{\tau_{\text{high}}}{\Delta_\tau}$	Each UABS
X coordinate of UABS	$-l/2, -l/2 + \Delta_x, -l/2+2\Delta_x, ...l/2$	$\frac{l}{\Delta_x}$	Each UABS
Y coordinate of UABS	$-l/2, -l/2 + \Delta_y, -l/2+2\Delta_y, ...l/2$	$\frac{l}{\Delta_y}$	Each UABS

where $C_{5\text{th}}(.)$ denotes the function that calculates 5pSE over the whole network area for a given state $\mathbf{S} = [\mathbf{X}_{\text{uabs}}, \mathbf{S}^{\text{ICIC}}_{\text{mbs}}, \mathbf{S}^{\text{ICIC}}_{\text{uabs}}]$. First, we calculate the SE for each UE as per Eqns (1)-(4). Then, we calculate the 5pSE as the value of SE such that 5% of the UEs have a value equal to or less than the 5pSE. As defined previously, $\mathbf{X}_{\text{uabs}}$ is the matrix representing the location of the N_{uabs} UABSs in three dimensions, $\mathbf{S}^{\text{ICIC}}_{\text{mbs}} = [\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}] \in \mathbb{R}^{N_{\text{mbs}} \times 2}$ is a matrix that captures individual ICIC parameters for each MBS, and $\mathbf{S}^{\text{ICIC}}_{\text{uabs}} = [\boldsymbol{\tau}, \boldsymbol{\rho'}] \in \mathbb{R}^{N_{\text{uabs}} \times 2}$ is a matrix that captures individual ICIC parameters for each UABS. The vectors $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_{N_{\text{mbs}}}]^T$ and $\boldsymbol{\rho} = [\rho_1, ..., \rho_{N_{\text{mbs}}}]^T$ capture the power reduction factors and scheduling thresholds, respectively, of each MBS. On the other hand, for each UABS, $\boldsymbol{\tau} = [\tau_1, ..., \tau_{N_{\text{uabs}}}]^T$ and $\boldsymbol{\rho'} = [\rho'_1, ..., \rho'_{N_{\text{uabs}}}]^T$ denote the CRE and scheduling threshold, respectively.

C. Parameter Optimization

The 5pSE is a function of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\tau}$, $\boldsymbol{\rho}$, $\boldsymbol{\rho'}$ and $\mathbf{X}_{\text{uabs}}$, which are our optimization space. Either these parameters can be optimized for individual MBSs and UABSs, or the same value of each parameter can be used for all MBS and UABS, which is suboptimal but computationally less complex.

To show that optimizing the above parameters individually for each MBS and UABS gives a better performance than optimizing the ICIC parameters jointly, we consider the hypothetical situation depicted in Fig. 2. Here, UE 4 is the critical UE to be protected from interference. Intuitively, as MBS 1 is closer to UE 4 compared to MBS 2, it is desirable for MBS 1 to transmit at a lower power during CSFs. Mathematically, $\alpha_1 = 0.3 < \alpha_2 = 0.7$. Also, as UE 5, that is served by UABS 2, is farther away from all the MBSs, UABS 2 does not have to use a large CRE to encourage UE 5 to associate with itself. In contrast, UABS 1 would have to use a larger CRE to encourage UE 4 to associate with itself. Mathematically, $\tau_1 = 15$ dB $> \tau_2 = 0$ dB. Therefore, optimizing the parameters individually for each MBS and UABS gives better performance.

The large size of the search space can be appreciated by referring to Table I, which gives the range and size of the parameters to be optimized. In this table, Δ_α, Δ_β, , Δ_ρ, $\Delta_{\rho'}$, Δ_x, and Δ_y denote the step sizes for α, β, ρ, ρ', x coordinate of a UABS's location, and y coordinate of a UABS's location, respectively, while ρ_{low} and ρ'_{low} denote the lower bounds for ρ and ρ', respectively. Similarly, ρ_{high} and ρ'_{high} denote the upper bounds for ρ and ρ', respectively. The actual size of the parameter space is depicted in Fig. 4, which shows the number of possible states in $\mathbf{S}$, over which the individual or joint optimization algorithms have to search, in order to

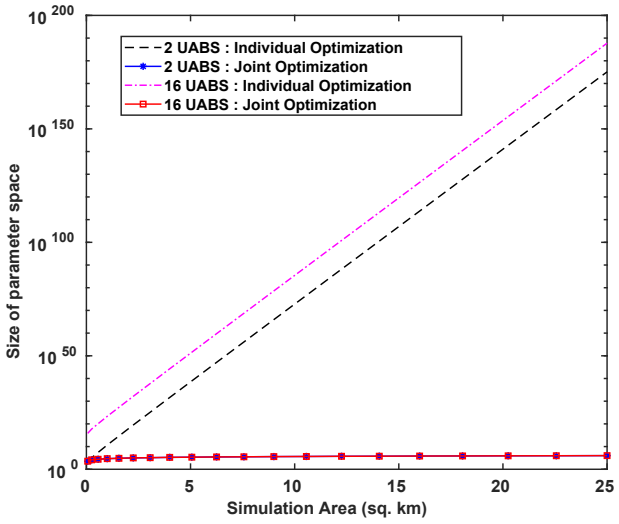

Fig. 4. Increase in the parameter search space size as a function of the total simulation area.

find the best state, $\mathbf{S}'$. Fig. 4 shows that, as the geographical area and correspondingly the number of MBSs and UABSs increase, the size of the search space increases much more rapidly for individual optimization than for joint optimization. This behaviour can be understood by calculating the number of possible permutations of α for 2 MBSs considered by joint and individual optimization approaches, assuming that α can take 4 different values from 0 to 1. While the joint optimization approach will only compare the 5pSE at these 4 values of α, the individual optimization approach will need to compare the 5pSE for $4^2 = 16$ different permutations of α values of the 2 MBSs. We also observe that, for the individual optimization approach, the size of the state space exceeds a googol (10^{100}) states as the simulation area increases beyond 15 km^2.

III. UAV Location and Interference Management

We considered and compared the performance of 3 algorithms for FeICICs - a brute force algorithm, a sequential algorithm, and deep Q learning algorithm. In this section, we present the brute force and sequential algorithms, while Section IV will present the deep Q learning algorithm.

A. Brute Force Algorithm

The brute force algorithm investigates the entire search state space of all possible values of UABS locations and individual eICIC and FeICIC parameters, and returns the state with the best 5pSE. This is illustrated in Algorithm 1. Since it individually searches for all possible parameter values, it is computationally infeasible for large areas and large number of MBSs and UABSs. This is evident from its time complexity, which is exponential in the number of MBSs and UABSs: $\mathcal{O}((\frac{1}{\Delta_\alpha+1})^{N_{MBS}} \times (1/\Delta_\beta+1)^{N_{MBS}} \times (\frac{(\rho_{\text{high}}-\rho_{\text{low}})}{(\Delta_\rho)})^{N_{MBS}} \times (\frac{(\rho'_{\text{high}}-\rho'_{\text{low}})}{(\Delta'_\rho)})^{N_{UABS}} \times (\frac{l}{\Delta_x})^{N_{UABS}} \times (\frac{l}{\Delta_y})^{N_{UABS}})$

B. Sequential Algorithm

In order to reduce the time complexity, we utilized a heuristic algorithm that initially assumes that there is only one UABS in the system and finds the best location, eICIC and FeICIC parameters for this UABS. It then considers that there are two UABSs in the system, with the parameters of the first UABS set to those found earlier, and finds the best parameters for this second UABS. Similarly, it then finds the best parameters for the third UABS considering the first two UABSs to be at their earlier determined states. The algorithm continues this procedure for the given number of UABSs. A similar approach is used to optimize parameters of MBSs. This algorithm is summarized in Algorithm 2. As we will see in the simulation results, this algorithm, somewhat surprisingly, performs close to the brute force search, and outperforms the DQN algorithm, to be discussed in the next section. A main reason for this is that the large number of parameters in Table 1 allows to *compensate* for non ideal selection of parameters in earlier stages. E.g. by tuning the scheduling thresholds and CRE, non-ideal location of a UABS can still result in good 5pSE. We also note that the sequential algorithm has linear time complexity, unlike the brute force algorithm, which has exponential time complexity. The sequential algorithm's time complexity, ignoring the constant terms, is: $\mathcal{O}(N_{MBS}) + \mathcal{O}(N_{UABS})$

Algorithm 1 Brute force algorithm

```
Best state, S' ← NULL
best5pSE ← −1
for all State S do
    current5pSE ← C_5th(S)
    if current5pSE > best5pSE then
        best5pSE ← current5pSE
        S' ← S
    end if
end for
```

Algorithm 2 Sequential algorithm

```
for all UABS do
    Assume all previous UABS to be at their best location and best ICIC parameters.
    Find best location and FeICIC parameters for current UABS.
end for
for all MBS do
    Assume all previous MBS to be at their best ICIC parameters.
    Find best ICIC parameters for current MBS.
end for
```

IV. Machine Learning Techniques for UAV ICIC and Placement

A. Q Learning

The Q learning algorithm is one of the commonly used reinforcement learning algorithms [11]. In reinforcement learning, an agent interacts with an environment by taking actions in different states and observing the costs or rewards of the actions. The agent starts out with random actions and eventually, by observing the rewards and by exploring different states, it learns the best action to take in each state in order to optimize the cumulative reward.

Fig. 5 shows the interaction of the MBS and the UABS agent with the UAV HetNet environment considered in this paper. Q learning allows the agents to act and learn as follows: being in a state s after selecting action a, and receiving the immediate cost c, a UABS or MBS agent n updates its knowledge $Q_n(s,a)$ for the particular state-action pair through the following operation:

$$Q_n(s,a) \leftarrow (1-\eta)Q_n(s,a) + \eta\Big[c_n + \lambda \min_a Q_n(s',a)\Big] \ , \tag{6}$$

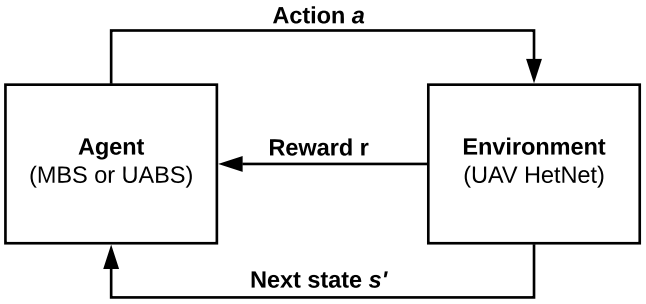

Fig. 5. Interaction of the MBS or the UABS agent with the UAV HetNet environment for Q-learning.

where η , the learning rate, is the agent's willingness to learn from the UAV HetNet environment, λ is the discount factor, and s is the next state. Lower values of λ give more importance to immediate rewards. We note that, conventionally, α, rather than η, is used to denote the agent's learning rate. We use η, as, in our work, we have already used α to denote the ICIC power reduction factor.

For our scenario, each MBS and each UABS is an agent. The state space is a sufficient representation of the environment at a point in time, containing all the information required by the agent to choose its next action. In our case, all MBS and UABS agents share the same state space, which is the set $[\mathbf{X}_{\text{ue}}, \mathbf{X}_{\text{mbs}}, \mathbf{X}_{\text{uabs}}, \mathbf{S}_{\text{mbs}}^{\text{ICIC}}, \mathbf{S}_{\text{uabs}}^{\text{ICIC}}]$, that captures the value of each ICIC parameter for each UABS and MBS, and also the location of each MBS, UABS and UE. However, the UABS and MBS agents have different action spaces. At each state, a UABS agent may choose any value for its ICIC parameters, τ, and ρ'. Additionally, the UABS agent may choose to move in any of the eight directions to change its location, or, alternately stay at its current location. An MBS agent, similarly, may choose any value for its ICIC parameters, α, β and ρ. Specifically, an MBS's action space is the set $\{\alpha, \beta, \rho\} \mid \alpha \in \{0, \Delta_\alpha, 2\Delta_\alpha, ...1\}, \beta \in \{0, \Delta_\beta, 2\Delta_\beta, ...1\}$, $\rho \in \{\rho_{low}, \rho_{low} + \Delta_\rho, ...\rho_{high}\}$, and a UABS's action space is the set $\{\tau, \rho', \delta_x, \delta_y\} \mid \tau \in \{0, \Delta_\tau, ...\tau_{high} + \Delta_\tau, \tau_{high}\}$, $\rho' \in \{\rho'_{low}, \rho'_{low} + \Delta_{\rho'}, ...\rho'_{high}\}, \delta_x \in \{0, \Delta_x, -\Delta_x\}$, $\delta_y \in \{0, \Delta_y, -\Delta_y\}$. δ_x and δ_y represent the change in the X and Y coordinate, respectively, of the UABS, as a result of the action. All the agents act one by one, in turn, changing the state of the UAV HetNet with their action. Our chosen reward function is the difference in 5pSE between two states.

As conventional Q learning algorithms maintain a Q table, with states as rows and actions as columns and with each cell representing the Q value of a specific action in a specific state, they cannot handle infinite state spaces, as the size of the Q table would become infinite. The Q table would take an extremely long duration to converge. Conventional Q learning suffers from issues of memory complexity, computational complexity, and sample complexity [12]. As illustrated in Fig. 4, the parameter space for our scenario can get extremely large and therefore, a conventional Q learning approach is not feasible. Deep learning, relying on the powerful *functional approximation* and *representational learning* properties of deep neural networks, provides us with the tools for overcoming these problems [13].

B. Deep Q Learning

Deep Q learning (DQN) came to the forefront of machine learning when it was used by DeepMind and Google to train an agent to achieve professional level scores on 49 different Atari 2600 games [14]. DQN extends Q learning by using a neural network to model the Q function, instead of using the simple Q table. One approach is to design the network to accept the state and action as the inputs, and provide the corresponding Q value as the output. Another approach is for the network to accept the current state as the input and provide the Q value of each possible action as the output. Fig. 6 depicts these 2 approaches. The latter approach, used in [14], is found to be better as only a single forward pass through the network is needed when we want to do a Q value update or pick the action with the highest Q value [15]. This is the architecture that we use as well.

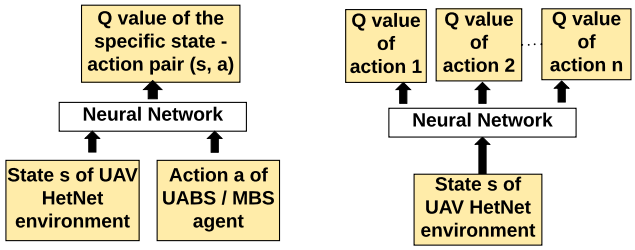

Fig. 6. (a) Naive formulation of deep Q network. (b) Evolved architecture of deep Q network

TABLE II. Neural network architecture.

Layer	Number of neurons	Activation
Input layer	Size of state space	ReLU
Fully connected 1	24	ReLU
Fully connected 2	24	ReLU
Output Layer	Size of action space	Linear

This neural network can handle infinite state spaces, and also recognize common patterns between similar states. Another advantage of deep Q learning is experience replay - the neural network is retrained after each action step, enabling the agent to adapt to changes in the environment. Thus, the deep Q learning agent can learn from its experience continuously.

In our simulations, we use a neural network with an input layer, two hidden layers and an output layer, as summarized in Table II. We update the network weights using the Adam optimization algorithm [16]. As explained in Algorithm 3, we allow the agents to explore the environment for a given number of steps and then choose the best state encountered so far.

Algorithm 3 Deep Q learning for UAV HetNets

```
while (Realization ≤ numRealizations) do
    Intialize a new realization of the environment
    while (NumberOfSteps ≤ numSteps) do
        Each UABS agent acts, observes the reward and the new state.
        Each MBS agent acts, observes the reward and new state
        Each agent learns by updating its neural network.
    end while
    Choose the best state encountered so far
    Preserve the neural network weights
end while
```

V. Simulation Results

In this section, using Matlab and python-based computer simulations, we compare the 5pSE observation of the wireless network, while using the three algorithms presented in Section IV. The system parameters used in these simulations are summarized in Table III.

Fig. 7 shows the CDF of 5pSE, obtained by using the three algorithms in 30 different realizations, with two UABSs. In each realization, UEs and MBSs were placed randomly. We observe that the CDF of the sequential approach is close to the CDF of the brute force approach, implying that the

TABLE III. Simulation parameters.

Parameter	Value
MBS and UE intensity	8 per km^2 and 100 per km^2
MBS and UABS transmit powers	46 dBm and 30 dBm
Path-loss exponent	4
Altitude of UABSs	121.92 m (400 feet)
Simulation area	0.5×0.5 km^2
Range expansion bias in dB	0 to 15
Power reduction factor for MBS during (α)	0 to 1
Duty cycle for the transmission of USF (β)	0 to 1
Scheduling threshold for UEs served by MBSs (ρ)	35 dB or 45 dB
Scheduling threshold for UEs served by UABSs ($\rho\prime$)	-20 dB to -5 dB
Downlink frequency	763 MHz

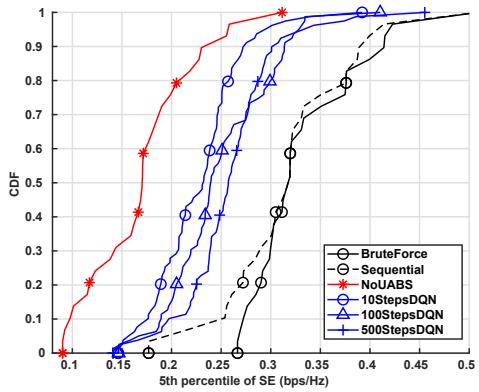

Fig. 7. CDF of 5pSE with sequential algorithm, brute force algorithm, and DQN algorithm with different number of steps.

performance of sequential approach is close to the optimal. The sequential approach, moreover, is much faster and, on an average, takes under a second. The brute force approach, on the other hand, takes 4-4.5 days to complete when run as a single process, i.e. without multi-threading. Both algorithms show improved performance compared to the case without any UABS in the system - showing that using UABSs improved the 5pSE.

Fig. 7 also shows that, if the deep Q learning agent is allowed to take more steps and explore the environment thoroughly, its performance improves. With fewer steps, the agent is unable to explore the environment properly and learn patterns. We also note that allowing the agent to run for more steps increases the execution time. Hence, a trade-off exists between the execution time and performance. We also note an interesting observation that, for our scenario, an algorithm based on a simple heuristic outperforms the intelligent DQN algorithm, possibly due to overfitting, as a result of the small sample space. To validate this, we compared the performance of the DQN and the sequential algorithm with more UABSs, and hence, with a larger sample space. Fig. 8 shows that the simple heuristic based algorithm consistently performs better than the DQN algorithm, for larger parameter spaces as well. This result implies that the particular optimization problem for maximizing 5pSE of FeICIC has a large number of local optima, that result in close performance to the global optima, and can be solved using the proposed sequential heuristic-based algorithm.

VI. Conclusion and Future Work

We quantified and presented three algorithms for interference management in UAV HetNets, compared their performance and provided some interesting insights. We observed that a simple heuristic based ICIC technique performs better

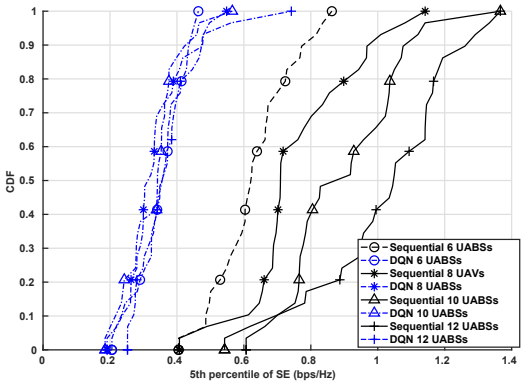

Fig. 8. CDF of 5pSE with sequential algorithm and DQN algorithm, for different number of UABSs.

than a more complex DQN based ICIC technique, while still being close to the optimal brute force ICIC technique. As a future work, we are investigating the performance of other multi-agent techniques like game theory, as well as developing more heuristics for ICIC. We will also explore the use of DQN for *updating* the parameter space after some disturbance to an initial FeICIC configuration.

References

[1] Technopedia, "Cell On Wheels (COW))," Jan. 2018. [Online]. Available: https://www.techopedia.com/definition/26001/cell-on-wheels-cow

[2] C. Geib, "An ATT Drone Is Connecting Puerto Ricans to Wireless Service," Nov. 2017. [Online]. Available: https://futurism.com/att-drone-connecting-puerto-ricans-wireless-service/

[3] J. Lyu, Y. Zeng, and R. Zhang, "Spectrum Sharing and Cyclical Multiple Access in UAV-Aided Cellular Offloading," *arXiv preprint arXiv:1705.09024*, 2017.

[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] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for cell association in UAV-enabled cellular networks," *IEEE Communications Letters*, vol. 21, no. 9, pp. 2053–2056, 2017.

[6] V. Sharma, M. Bennis, and R. Kumar, "UAV-assisted heterogeneous networks for capacity enhancement," *IEEE Commun. Let.*, vol. 20, no. 6, pp. 1207–1210, 2016.

[7] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 137–150, 2014.

[8] A. Merwaday, S. Mukherjee, and I. Güvenç, "Capacity analysis of LTE-Advanced HetNets with reduced power subframes and range expansion," *EURASIP J. Wireless Commun. Netw.*, no. 1, pp. 1–19, Nov. 2014.

[9] S. Mukherjee and I. Güvenç, "Effects of range expansion and interference coordination on capacity and fairness in heterogeneous networks," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, CA,USA, Nov. 2011, pp. 1855–1859.

[10] A. Kumbhar, I. Guvenc, S. Singh, and A. Tuncer, "Exploiting LTE-Advanced HetNets and FeICIC for UAV-assisted Public Safety Communications," *IEEE Access*, no. 99, pp. 1–13, Oct. 2017.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[12] A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman, "PAC model-free reinforcement learning," in *Proc. ACM Intl. Conf. Machine learning*, Pittsburgh, PA, USA, Jun. 2006, pp. 881–888.

[13] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint arXiv:1708.05866*, 2017.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] T. Matiisen, "Guest Post (Part I): Demystifying Deep Reinforcement Learning," Dec. 2015. [Online]. Available: https://ai.intel.com/demystifying-deep-reinforcement-learning/

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Repr. (ICLR)*, Ithaca, NY, 2015, pp. 1–13.