



HAL
open science

Drone-assisted cellular networks: a multi-agent reinforcement learning approach

Seif Eddine Hammami, Hossam Afifi, Hassine Moun gla, Ahmed E. Kamel

► **To cite this version:**

Seif Eddine Hammami, Hossam Afifi, Hassine Moun gla, Ahmed E. Kamel. Drone-assisted cellular networks: a multi-agent reinforcement learning approach. ICC 2019: 53rd International Conference on Communications, May 2019, Shanghai, China. pp.1-6, 10.1109/ICC.2019.8762079 . hal-02101394

HAL Id: hal-02101394

<https://hal.science/hal-02101394v1>

Submitted on 10 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Drone-assisted cellular networks: A Multi-Agent Reinforcement Learning Approach

Seif Eddine Hammami^{*}, Hossam Afifi^{*}, Hassine Mouncla^{*†}, Ahmed Kamel[‡]

^{*}UMR 5157, CNRS, Institute Mines Telecom, Telecom SudParis Saclay, France

[†]LIPADE, Paris Descartes University, Sorbonne Paris Cite, Paris, France

[‡]Department of Electrical Engineering, Iowa State University

Abstract—Drone-cell technology is emerging as a solution to support and backup the cellular network architecture. cell-drones are flexible and provide a more dynamic solution for resource allocation in both scales: spatial and geographic. They allow to increase the bandwidth availability anytime and everywhere according to the continuous rate demands. Their fast deployment provide network operators with a reliable solution to face sudden network overload or peak data demands during mass events, without interrupting services and guaranteeing better QoS for users. With these advantages, drone-cell network management is still a complex task. We propose in this paper, a multi-agent reinforcement learning approach for dynamic drones-cells management. Our approach is based on an enhanced joint action selection. Results show that our model speed up network learning and provide better network performance.

I. INTRODUCTION

The continuous growth of data demand and the increase in wireless traffic rates in new mobile networks needs intelligent and dynamic technologies for telecommunication management. Recent studies predict that the new generation cellular standards (like 5G) will rely much more heavily on a dense and less power consuming networks to serve dynamically requested data rates to the user [1]. Small cells mounted on unmanned aerial vehicles (UAVs) or drones (we call them drone-cells hereafter) are proposed, as an alternative to fixed femto-cells, to support existing macro-cell infrastructure. The deployment of these mobile small cells consists on move these small cells toward a target positions (usually within the range of the macro-cell to support) based on the decision made by a mobile network operator. Since drone operation costs money, the drone-cells movement toward adequate positions must be correlated with the amount of data requests, i.e. drone-cells should support overloaded cells (Figure 1). Hence, an intelligent entity should be added to the network that instantly monitors the network state and finds the optimal decision to control drone-cells. Several studies have discussed the theoretical ability of cellular networks to use drone-cells to support their existent macro-cells [2], [3] but the readiness of those networks to integrate such dynamic entities has not been discussed. For example, drone-cells require coordinated insertion to the network infrastructure while serving subscribers and smooth de-association at the end of their activity (low battery, low data demand ...). This requires capabilities for efficient network configuration and management flexibilities. It needs self-organizing capabilities for the networks. Massive

amounts of information about users behavior and network load must be continuously collected and analyzed by intelligent algorithms. This should trigger the usage of UAVs on field. On the other hand, the promising development of artificial intelligence techniques and machine learning tools may speed up the development of cellular networks and lead to develop such dynamic and self-adaptive tools that help the network operators to better manage their infrastructure and integrate new techniques. Machine learning techniques have the advantage of exploiting the plethora of data generated by cellular networks to improve these dynamic deployment techniques [4].

In this work, we propose a dynamic reinforcement learning solution for drone-cells networks that exploits real traces of demand profiles and adapts in real time the deployment of drone-cells according these demands. We choose an efficient machine learning paradigm instead of classical linear programming models. Our solution is based on a multi-agent reinforcement learning (MARL) approach which is applied in two steps: an off-line exploration step and an on-line one. The off-line step consists in a learning phase where the drone-cells of the multi-agent topology learn, in a cooperative reinforcement learning context, how to react and move according to different scenarios of bandwidth demands. Real Traces of network time-series load are used for this phase. In the on-line phase, the pre-learning agents demonstrate how they are adapted instantly on a real scenario of load peak signaled by the STAD framework. The cooperative reinforcement learning is based on a joint action selection that aims to choose the optimal set of agents action. We propose a joint action selection algorithm and we compare it against the classical grid selection algorithm.

The paper is organized as follows. In section 2 we present related work, while in section 3 we present a highlight of multi-agent reinforcement learning approach. Section 4 develops our MARL-based model. Section 5 presents simulations uses-cases and results and we conclude in section 6.

II. RELATED WORK

Smart deployment of drone-cells has been a topic of recent research; examples of such works include [5], where the authors use binary integer linear programming (BILP) to selectively place drone-cells in networks and integrate them to an existent cellular network infrastructure while temporal increase of user demands occurs. They design the air-ground channel for drone-cells as the combination of two

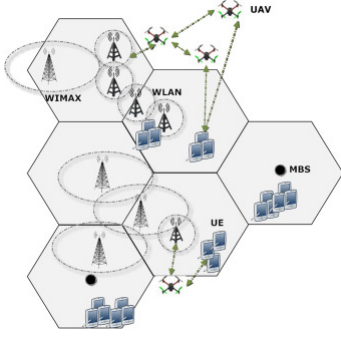


Figure 1: Graphical illustration of a Drone-assisted network

components: a non-line-of-sight (NLoS) and one for a line-of-sight (LoS) communication. Authors in [3] investigate the adequate number and the 3D placement topology for drone-cells formation. The study aims to optimize the geographic coordinates of drone-cells and their total number while serving users. In [6], authors propose a framework for drone-cells formation. The framework aims to optimize the coverage (by increasing or decreasing the drone altitude) in order to remove dead zones and reach network capacity goals.

In contrary to previous studies that aim to optimize the placement of drone-cells before deployment, [7] investigates path optimization issues during drone-cells formation.

Our proposed contribution develops new methods not only in optimal network deployment for drone-cells, but also in how to dynamically manage this deployment in order to enhance the existent infrastructure using a multi-agent reinforcement learning approach.

III. REINFORCEMENT LEARNING

In this section, we introduce the background on “multi-agent” reinforcement learning. We also discuss using multi-agent approach in our contribution and we present its benefits/advantages against a simple single-agent.

A. Multi-agent reinforcement learning approach

A typical agent has its set of states, actions and rewards. As the wireless topology is composed of multiple drone-cells, and since modeling a single reinforcement learning model for each single drone-cell leads to an exponential growth in the $(state, action, reward)$ space, we propose to employ a multi-agent reinforcement learning model. It will reduce the space and have a better global federated goal.

The multi-agent reinforcement learning approach [8], [4], is a generalization of the single model and is modeled by a stochastic game. The multi-agent RL employs a joint action, which is the combination of actions to be executed by each agent at state k .

The stochastic game is represented by the tuple $\langle X, A_1, \dots, A_n, f, r^1, \dots, r^n \rangle$ with n is the total number of agents. A_i with $1 \leq i \leq n$ is the set of possible actions of agent i . So we can define the joint action as $A : A_1 \times A_2 \times \dots \times A_n$. $r^i : X \times A \times X \rightarrow \mathbb{R}$ is the reward function of agent i that

is assumed to be bounded. $f : X \times A \times X \rightarrow [0, 1]$ is the transition probability function. In the multi-agent approach, the transitions between states are the result of a joint action of all the agents which is expressed as: $ja_k = [a_{1,k}, \dots, a_{n,k}]$ where $a_{i,k} \in A_n$. The reward $r_{i,k+1}$ ($r^i(x_k) = r_{i,k+1}$) resulting from executing the action $a_{i,k}$ is computed according to the joint policy jh and the joint expected return is expressed as follows:

$$R_i^{jh}(x) = E \left\{ \sum_{i=1}^{\infty} (\gamma^i r_{k+1} | x_0 = x, jh) \right\} \quad (1)$$

The Q -learning function depends also on the joint action and the joint policy and it is expressed as follows:

$$Q_{k+1}^{jh}(x_k, ja_k) = Q_k(x_k, ja_k) + \alpha_k [r_{k+1} + \gamma \max_{ja'} Q_k(x_{k+1}, ja') - Q_k(x_k, ja_k)] \quad (2)$$

In our model, we are adopting the fully coordinated multi-agent RL, and in this case all reward functions are the same for all agents, $r^1 = \dots = r^n = r$

B. Motivations of using MARL approach

Along with advantages due to the distributed feature of the multi-agent systems [9], as the acceleration become possible by parallelizing the computation, MARL approaches emerge with the benefit of sharing experiences between agents. Experience sharing helps agents to learn faster and reach better performances. Hence, the agents can communicate between each other to share experience and learning so that the better learned agents may speed up the learning phase of other agents. Moreover, multi-agent systems facilitate the insertion of new agents into the system which leads to came up with scalability issues affecting classic method such as linear programming.

IV. DRONE-CELLS NETWORK AGENT MODEL

In this section we present our proposed model for drone-cells optimization and management and we explain the mapping between our model parameters and the theoretical model presented in the previous section.

A. Model Framework Description

Our drone-assisted network model is based on a centralized multi-agent reinforcement learning approach. The framework consists of a multi-agent system implementation based on MESA package. It is composed of entities that describe the system parameters and manage the interaction between the different agents. The framework is described in figure 2.

Since the approach is centralized, we need a coordinator agent that collects information from different agents and intervenes in the actions selection process. In our framework, this coordinator agent is represented by the network operator agent (which can be also a centralized entity in the network architecture). The framework is composed also of a set other agents. We distinguish two types of agents: Active agents executing the model actions and passive agent that participate

in the system interaction but without executing any action. The active agents are the drone-cells that are represented by the drone entity in our framework. The action to be executed by a drone is either to move to a cell location in order to serve it or to go back to the facility location (in case of battery running out or no cell left to serve). This entity defines the battery level $BT_{d,t}$ of a drone d at time-stamp t , the bandwidth offer level $BW_{d,t}$, the status of drone d (whether is serving, idle or charging) and other parameters.

The passive agents are the cell (represented by Cells entity) and the facility (represented by Facility entity) agents. The cell agent entity defines the bandwidth demand rate at each time-stamp, the geographic location of the cell and the status of the cell (whether is served or not). The facility agent constitutes the initial location of all drones and where drones can be charged. If a drone is not serving a cell must move to the facility location. We suppose that the facility location is situated in the middle of the service area in order to optimize the flight time toward cells.

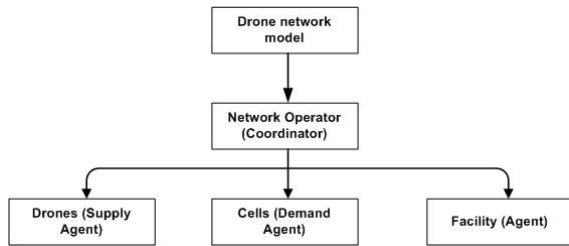


Figure 2: Graphical illustration of the MARL Drone-assisted network Framework and the relationship between the Agents

B. Model States

The model states provide information about the network status at each time-stamp t such as drones parameters, Facility actual capacity or cells demand rates. In our MARL model, each drone agent d has its own state S_t which describes the actual parameters at time t . These parameters are the drone battery level $BT_{d,t}$, the bandwidth offer capacity level $BW_{d,t}$.

Moreover, each cell agent C state is represented by the aggregated throughput demand $Td_{C,t}$ at time t that exceeds the maximum capacity of the macro base station that covers it, and that should be assisted by drones. The facility agent state is also described by the current number of drones that are placed in it and either they are charged or not. All these information are gathered by the network operator agent (The coordinator agent) in order to choose the joint action to take for all drones in order to maximize the total network's QoS.

C. Model Actions

The action that a drone agent can execute, at each state, is either to move and serve a cell and serve it or to move to the facility (to charge its battery or in case of no cell left to serve). Unlike single agent models, the action of each drone agent is taken by the coordinator (centralized approach) to maximize the global network QoS. Hence, it is not always

straightforward that the chosen agent action maximizes its own reward. Otherwise, the coordinator agent chooses a joint action (set of action for all drones) that maximizes the global network performance.

D. Reward function

The reward function for each drone agent is measured at each state and after choosing the action to execute. The reward function measures the local network service ratio (NSR) and it is the fraction of the served throughput by the drone d at time t and the throughput request of the cell to be served by d at the same time. Its expression is as follows:

$$r_{d,t} = \frac{BandwidthServed_{d,t}}{BandwidthRequest_{C,t}} \quad (3)$$

E. Coordinated multi-agent RL

Choosing the optimal joint action is a critical step for the multi-agent RL model. The criteria of choosing a better joint action helps to reduce the computational cost of searching the joint actions, which is exponentially proportional to the agent number and to provide a better performance. We present in this section the hill climbing search algorithm (HCS) which is proposed in [10]. Then, we propose an enhancement of the Hill climbing search algorithm for an optimal joint action selection (eHCS) algorithm that speeds up the action search. The model is then compared to the HCS algorithm.

1) *Hill climbing search algorithm*: The main objective of the hill climbing algorithm is to examine the neighboring agents one by one and to select the first neighboring agent action which optimizes the current reward as the next node. The algorithm is resumed by the following steps:

- 1- Initialization step: Select an initial joint action JA by picking the action of each agent that maximizes the reward.
- 2- Choosing randomly an agent and its action in the previous JA is changed by a neighboring action.
- 3- The new formed action is evaluated. If it provides a better network service ratio (NSR) so it is stored as optimal JA.
- 4- Repeat steps 2 and 3 until testing all combinations.

2) *Enhanced Joint optimal action selection algorithm*: Our multi-agent reinforcement learning model is based on a semi-centralized cooperative solution. The centralized part consists of adding a coordinator agent (which is the network operator agent here) which is able to collect all drone agents and other model agents information in order to select the "joint optimal action" to be executed by drones. The centralized approach reduces then the complexity of the system, speeds up the selection process and alleviates the system information sharing among all agents. Unlike the distributed approach which consists on forwarding all information between all agents, that may be a time-consuming process. Plenty of centralized joint action selection algorithms exist in the literature such as hill climbing search [10], Stackelberg Q-Learning [11] etc. In our model we propose a modified version of hill climbing search algorithm which fits our drone actions selection problem.

The main objective of this algorithm is described as follows: At first, each drone agent, at a state S_t at time-stamp t , choose

its own optimal action to execute without taking into consideration other drones configuration. Then, the coordinator agent collected all drone information (Battery level, remaining bandwidth) as well as their chosen actions. It collects also cells and facility agents information. After that, the coordinator sorts decreasingly the set of cell demands (respectively the set of drone offers). Then, it tries, at each iteration, to find a match between the drone actions (the position to move toward) and the demand positions (cell positions) and assign the adequate drone to the cell. Then, if the cell is totally served, the coordinator looks for drone agents whose action is identical to the served cell position and re-select a new action for the drone according a new Q-table. The new Q-table is formed with the remained possible actions (except the selected priorly).

The coordinator repeat the same process over all drones. This selection method avoid assigning many drones to the cells with higher demand and try to share drones all over the network.

The semi-centralized joint action selection algorithm is presents in algorithm 1.

Algorithm 1 Centralized Joint action selection

- 1: **Input:** (Action,State) space, Q-tables, ServedCell_list
 - 2: **Output:** Joint Action JA_t
 - 3: *Collect_system_information*
 - 4: **for** d in Drones **do**:
 - 5: **if** $Action_{d,t} \in ServedCell_list$ **then**
 - 6: *ExtractNewQ - table(d)*
 - 7: *ActionSelection(NewQ_table(d))*
 - 8: Verify NewAction(d,t)
 - 9: Find MatchAction(Cell)
 - 10: **if** Matched Action **then**
 - 11: Actualize ServedCell_list
 - 12: Add NewAction(d) to JAction_List
- return** JAction_List
-

Note that the joint action selection algorithms are used only during the exploration phase (training phase). Once the Q-tables are calculated and their entries are the optimal values, they are used for the online exploitation phase.

V. SIMULATION AND RESULTS

A. Use-case Scenario

We validate our framework by using real datasets of cell demand time-series extracted from the Milan CDRs dataset. We reproduce in this contribution a mass event where macro-cells are assisted by drone-cells managed by our framework in order to cover the unusual peaks of cellular user demands. As shown and analyzed aforementioned, we detected some unusual behavior of demand around the SanSiro stadium during football match compared to workdays. Figure 3 represents the cells segmentation of the Sansiro stadium and areas around. Figure 4 depicts the daily demands time-series of these cells and we can notice that the SanSiro stadium cells have a high peak of user demand around 8pm (football match time) while

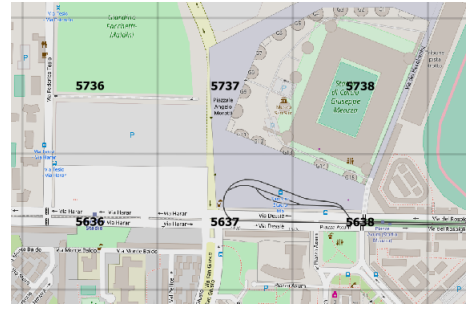


Figure 3: Graphical illustration of cell segmentation of SanSiro area

nearby cells have two peaks: one before the match and another after it. This behavior may correspond to the arrival time of

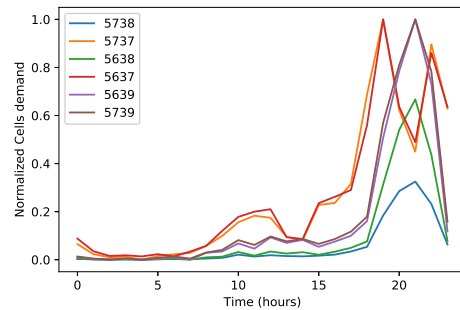


Figure 4: Example of demand time-series of SanSiro areas cells

supporters and their departure time after the match. We can notice that these cells are covering a parking area and some metro stations which make our assumptions believable. We apply our framework to these demand time-series in order to manage the abnormal increase of users demand during this period of the day and we show results in the next part.

B. Results

The framework is applied on cell demands between 6pm and 11pm, a period of time when the abnormal demand peak are occurring. During this period, drone-cells will attempt to assist the existent macro-cell and serve the additional demand. Table 1 resumes the simulation parameters.

Table I: Simulation Parameters and Values

Parameters	Values
Drone-cells max battery	100
Drone-cells max rate	50 Mb/s
Cell max rate demand at peak hours	120 Mb/s
Battery life-time factor	0.5
Number of cells	6
Learning rate	0.75
Learning period (exploration)	300
exploitation period	50

As performance metric, we choose to use the network service ratio (NSR) which is defined as follows:

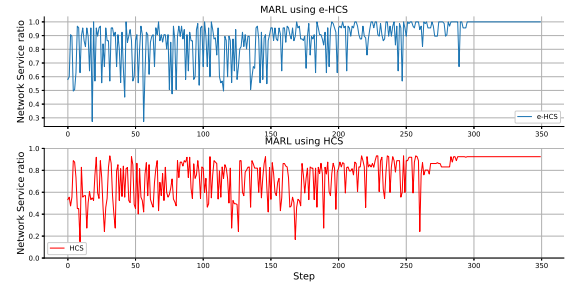
$$NSR = \frac{TotalBandwidthserved}{Totalbandwidthdemand} \quad (4)$$

where $TotalBandwidthserved$ is the sum of the served bandwidth by all drones and $Totalbandwidthdemand$ is the sum of all cell demands.

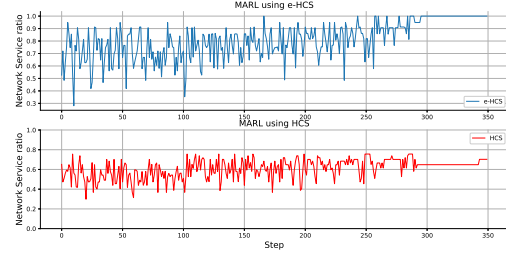
Our MARL model based on the enhanced hill climbing algorithm is compared also to a MARL model with the classic hill algorithm. Figures 5-10 shows the simulations and comparison results between both models. Figure 5 depicts the scenario simulation at 6pm (Fig 5a) and 7 pm (Fig 5b), the time of supporters arrival before starting the football match. We use in this scenario only 8 drones to cover the area for both models. We can notice that after finishing the learning period (after 300 steps) the network service ratio converges to 1. During the learning period, the drones are exploring several options of possible actions and communicating their decisions to the coordinator. At the end of this period, the coordinator selects the optimal joint action for all drones. All cells are fully served. We also notice that 8 drones are sufficient for our eHCS-based model, to serve the network. Whereas, the concurrent model based on HCS performs worse than our model. We notice that in this scenario, and after the exploration phase, the model fails to converge to 1 and the network service ratio converges to 0.9 and 0.6 at 6pm and 7pm respectively. Hence, the HCS based model performs 10% lower than our model at 6pm and %40 lower at 7pm, when demand is much higher. So, we can say that our model performs much better than the other model during peak hours. Moreover, the exploration period was not sufficient for the HCS based model to achieve the optimality and we conclude that our model is faster with 8 drones for the period between 6pm and 7pm. Figures 6-10 show simulation and comparison results using 14 drones. We notice that for all time periods, the network service ratio converges to 1 after the learning period for our model (the learning is very fast in this situation). Whereas, the HCS based model success to converge to 1 only for three time periods; at 6pm, 7pm and 11pm while its network service ratio is lower than 1 for the rest of periods. Furthermore, for scenarios at 6pm, 7pm and 11pm, even that the HCS based model converges to 1 after the exploration period, it is slower that our model.

Globally, we notice that the HCS model fails to reach an optimal steady state contrary to our model. The exploration period was not sufficient to learn and to find the optimal joint actions within these periods. The other model does not manage battery in a performant way. Most of drone batteries are lost in an unefficient way (Battery life-time is 2 slots and they need one time slot to recharge). That is why the other model reaches optimality at 11pm again (after recharge). Hence, in order to satisfy the global demand during the exploration period with the same period size, the HCS based model needs more drones to serve all demands. MARL learns these constraints and succeeds to efficiently serve the network. With this configuration, our enhanced model only needs 11 drones at total to cover the whole area during the football match event. These 11 drones alternate to serve the requested amount of data according to their battery life level.

Figure 11 depicts the deployment of drone-cells using our



(a) Network QoS evolution 6pm



(b) Network QoS evolution 7pm

Figure 5: Network service ratio evolution in function of execution steps using 8 drones

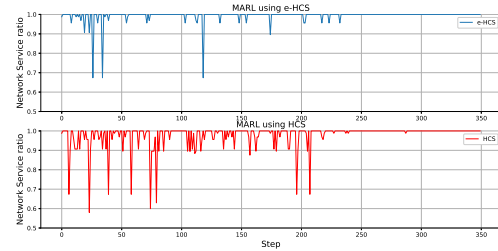


Figure 6: Network QoS evolution in function of execution steps with 14 drones and at 6pm

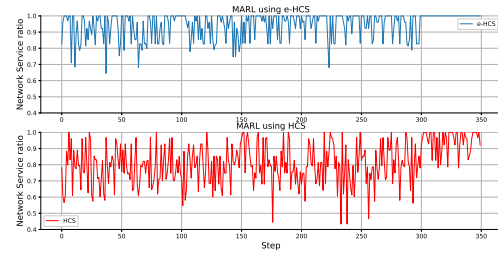


Figure 7: Network QoS evolution in function of execution steps with 14 drones and at 8pm

enhanced model at 9pm, time-slot corresponding to a heavy demand behavior. The colors represent the demand intensity at each cell. We can notice that 9 drones of 14 are deployed at cell locations that guarantee serving the total required bandwidth. Note that at this time 4 drones are out of charge while 1 drone is not serving at all.

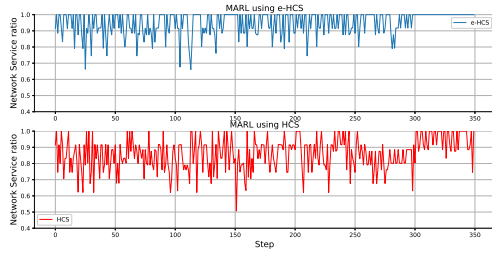


Figure 8: QoS, 14 drones, 9pm

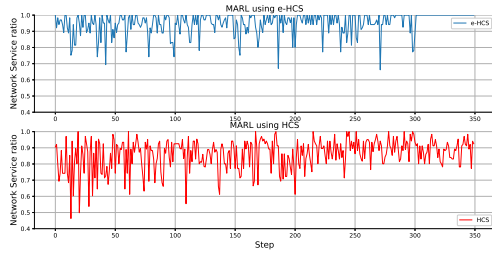


Figure 9: Network QoS evolution in function of execution steps with 14 drones and at 10pm

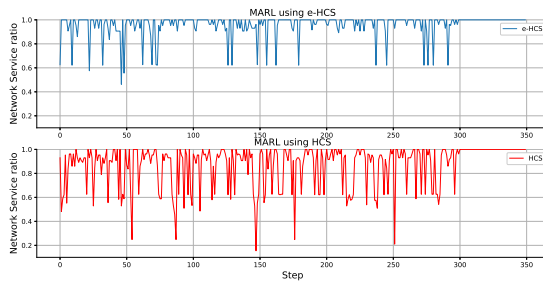


Figure 10: Network QoS evolution in function of execution steps with 14 drones and at 11pm



Figure 11: Illustration of drone-cells deployment at 9pm

VI. CONCLUSION

Drones-cells management is still complex and needs advanced and intelligent algorithms. Even with their fast deployment, drone-cells networks suffer from coordination issues. On the other hand, battery technology is still limited and since it

is split between navigation and antenna beaming/transmission, energy presents a major constraints for drones-cells deployment. We present in this paper a solution based on drone-cells to support macro cells of the classic cellular network during mass events when data rate demand explodes. We solve this complex navigation/coverage management by a multi-agent reinforcement learning approach for this dynamic network deployment. We also propose an enhanced joint action selection algorithm to alleviate the coordination complexity between drone-cells agents and also speed up the search phase of the optimal joint action. Our model takes into consideration the battery life constraints while aiming to maximize the network service ratio. Our solution is validated with real network traces and we provide a bench marking analysis. Our model based on the enhanced joint action selection that we propose is compared with a model based on hill climbing search algorithm. Results show that our model outperforms the second model not only when the rate demand is lower but especially at peak time service.

Our model offers hence a better solution for network operators to dynamically manage their network and to provide efficiently a better QoS for users during mass events. A future perspective will introduce the deep aspect that can add classification and memory of early situations to the multi agent system.

REFERENCES

- [1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.
- [2] A. Alsharoa, H. Ghazzai, A. Kadri, and A. E. Kamal, "Energy management in cellular hetnets assisted by solar powered drone small cells," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE, 2017, pp. 1–6.
- [3] E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu, "On the number and 3d placement of drone base stations in wireless cellular networks," in *Vehicular Technology Conference (VTC-Fall)*. IEEE, 2016, pp. 1–6.
- [4] S. E. Hammami, H. Afifi, M. Marot, and V. Gauthier, "Network planning tool based on network classification and load prediction," in *Wireless Communications and Networking Conference*. IEEE, 2016, pp. 1–6.
- [5] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3d drone placement in 5g+ wireless networks," *arXiv preprint arXiv:1702.08395*, 2017.
- [6] S. Park, H. Kim, K. Kim, and H. Kim, "Drone formation algorithm on 3d space for a drone-based network infrastructure," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*. IEEE, 2016, pp. 1–6.
- [7] M. Jun and R. DAndrea, "Path planning for unmanned aerial vehicles in uncertain and adversarial environments," in *Cooperative control: models, applications and algorithms*. Springer, 2003, pp. 95–110.
- [8] A. Souza and H. Afifi, "Adaptive data collection protocol using reinforcement learning for vanets," in *9th International Wireless Communications and Mobile Computing Conference*. IEEE, 2013, pp. 1040–1045.
- [9] B. Nour, K. Sharif, F. Li, H. Mounsla, and Y. Liu, "M2hav: A standardized icon naming scheme for wireless devices in internet of things," in *Data Driven Control and Learning Systems (DDCLS), 2017 6th*. Springer, 2017, pp. 727–732.
- [10] S. Proper and P. Tadepalli, "Scaling model-based average-reward reinforcement learning for product delivery," in *European Conference on Machine Learning*. Springer, 2006, pp. 735–742.
- [11] C. Cheng, Z. Zhu, B. Xin, and C. Chen, "A multi-agent reinforcement learning algorithm based on stackelberg game," in *International Conference on Wireless Algorithms, Systems, and Applications*. IEEE, 2017, pp. 289–301.