This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3515006

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

Interference Mitigation and Collision Avoidance of Dynamic UAVBSs Network via Mobility Control: A Game Theoretic Approach

OMAR ALI THABET[1], (Student Member, IEEE), DIDEM KIVANC TURELI[2] (Member, IEEE), UFUK TURELI[1], (Member, IEEE)

[1] Electronics and Communication Engineering Department, Yildiz Technical University, 34220 Istanbul, Turkey
[2] Mechatronics Engineering Department, Istanbul Okan University, 34959 Istanbul, Turkey

Corresponding author: UFUK TURELI (utureli@yildiz.edu.tr).

ABSTRACT Mobility control of UAVBSs can avoid collision and improve the power efficiency and coverage of the wireless network. In this work UAVBS mobility control is formulated as an exact potential game. Three algorithms are proposed to solve this problem under different connectivity and complexity scenarios. In the first scenario on board computation and power may be limited due to other functions. Under this scenario the UAVBSs-Better Direction Control (UAVBSs-BDC) algorithm works iteratively based only on the UAV utility function with linear time to directly optimize the action selection based on the UAVBS's utility. The Utility-Driven Partial Synchronous Learning (UDPSL) algorithm speeds up convergence by using a learning algorithm. This algorithm is seen to increase incidence of collision when UAVBSs are located close together and requires an additional collision avoidance mechanism. The Neighbor Responsive Adaptive-Partial Synchronous Learning (NRA-PSL) algorithm controls the UAVBS's trajectory via conditioned response to its neighbor UAVBSs to select the action that guides the UAVBS to better direction. This algorithm requires additional information about the interference posed by neighbor UAVBS and their location in the cell, which allows it to design a better trajectory which converges faster to the optimal placement of UAVBSs in the cell.

INDEX TERMS Collision Avoidance, Interference Mitigation, Mobility Control, Potential Game, UAVBSs Network.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have several benefits, such as low cost, safety due to the lack of need for human pilots, and high mobility. That is why they have been favorites of many civil and military applications in the past few years. UAV use in wireless communication networks has increased in the past few years because of the enhancement of wireless networks [1]. UAVs are considered one of the promised solutions for the next generation of cellular communication networks. Thus, these UAVs can be deployed as relays [2] or small-mounted base stations [3], but the reason that motivates the telecom operators to deploy these air nodes is their flexibility, ease, and rapid deployment. Thus, UAVBSs are utilized by the telecom operators to enhance the coverage and capacity of cellular networks, particularly during temporary events such as the rally of people during a national celebration or to enhance coverage for rural areas in addition to using these flying nodes during emergency events such as earthquakes or any natural disaster. Some of the telecom operators have conducted practical experiments to study the influence of UAVBS on the cellular network system, such as Nokia and EE telecom operator, which investigated the impact of single drone base station (DBS) on rural areas to improve coverage [4], in addition to the most recent experiment of AT&T operator that was conducted to study the impact of single cell on wings (COW) on the 5G network [5].

For the deployment of multi-UAVBSs, mobility control is an important challenge. Controlling UAVBS's mobility well can reduce interference for all communication links in the network and avoid physical collisions among UAVBSs. This work proposes algorithms for UAVBS control to reduce interference and avoid collisions between the UAVs. Thus, collision avoidance is considered a critical aspect of guaranteeing safe and efficient operation for moving UAVBSs. There are several methods utilized for collision avoidance, such as 1) the geometric approach, which is utilized to avoid physical collisions among UAVs based on geometric equations. 2) A potential field approach that uses the concept of an attractive or repulsive force field either to attract a UAV towards a target or repel it from an obstacle. 3) path planning approach, which aims to find the shortest path in a map or database graph where obstacle edges are already known. 4) a vision-based approach that depends on visual data captured by onboard cameras to detect and avoid potential obstacles or collisions [6].

Thus, in our work, collisions are avoided via mobility control, which optimizes the trajectories for the UAVBSs. Trajectory optimization as a method fits within the broader umbrella of path planning, focusing on optimizing the trajectories for each UAV to navigate safely and effectively in the given environment. This approach is crucial for controlling the movement of multiple UAVs and ensuring consistency in distancing the paths of UAVBSs from each other, which contributes to advancing collision avoidance and interference mitigation through mobility control in UAVBSs networks. The paper is structured as follows: The literature review is reviewed in Section II. The system model is presented in Section III, followed by a potential game approach in Section IV. We demonstrated the simulation results in Section V. Section VI concludes the paper.

II. LITERATURE REVIEW

Interference and physical collisions among UAVBSs are critical issues that can be addressed through mobility control. This section reviews previous studies related to UAVBS deployment, with a specific concentration on UAVBSs mobility control.

Instead of statically deploying UAVs, the literature explores dynamic UAV movement. Han et. al. [7][8] discuss maintaining mobile ad hoc network (MANET) connectivity by controlling the movement of single UAVs. They employ the minimal spanning tree model, utilizing Delaunay triangulation and candidate UAV placement strategies to enhance connectivity. These works focus on optimizing UAV movement based on node behavior, ensuring improved connectivity for mobile ad hoc ground users. Similarly, Jiang and Swindlehurst [9] also delve into optimizing single UAV movement, but with a different focus. They concentrate on optimizing the heading and trajectory of single UAVs to enhance the user uplink rate. Instead of solely considering connectivity, they employ a position prediction algorithm to optimize the uplink rate for users. Takaishi et. al. [10] could be further explored in the work of Han et. al. [7][8], which investigates enhancing end-to-end connection probability through multi-hop UAV communication instead of single UAV. Thus, their proposed UAV trajectory decision scheme aims to improve connectivity by optimizing the trajectory of UAVs. Zhu et. al. [11] address motion control and collisions among UAVBS chains to boost link capacity between ground mobile nodes using an artificial potential field (APF) model, however, this study does not explicitly consider dynamic obstacles such as other UAVs.

Zeng et. al. [12] have studied the maximization of throughput between a fixed source and destination through mobile UAV relays by enhancing the source and relying on transmit power in addition to optimizing the relay trajectory. Thus, the problem of trajectory optimization given a fixed power allocation is formulated as a non-convex optimization problem. Fotouhi et. al. [13] extend the exploration of optimizing uplink rates by offering specific three algorithms for autonomous drone repositioning in response to user activities. These algorithms aim to optimize the achievable uplink rate for users. The first algorithm maximizes spectral efficiency by selecting movement directions based on the calculated average spectral efficiency. The second allocates bandwidth to the nearest user equipment (UE) and moves towards it, while the third prioritizes users with the least remaining data and allocates full bandwidth accordingly.

Shi et. al. [14] propose a path-planning method for UAV swarms based on generalized potential games, enabling a swarm of UAVs to coordinate trajectories based on high-level team goals in a decentralized manner. Chen et. al. [15] also use a potential game approach, but for interference-aware spectrum access in multi-cluster FANETs. Ruan et. al. [16], focus on energy-efficient multi-UAV coverage deployment by using game-theoretic frameworks. However, this work relies on a central ground controller for UAVs' decision-making, thereby making it impractical to deploy it for emergencies due to the significant amount of information exchanged between the UAVs and the controller. Kim et. al. [17], also minimize energy consumption per downlink rate in UAV networks, using a mean field game approach. Thus, the primary goal is to control the velocities of a massive number of UAVs in a distributed way to decrease the long-term energy consumption per downlink rate to ensure collision avoidance.

Zhong et. al. [18] optimize UAV deployment to collect sensor data using a potential game approach and introduce an online learning algorithm to find the NE state. Charlesworth [19] uses non-cooperative games to coordinate UAV movement to maximize the support of ground users while considering the actions of other UAVs.

Fotouhi et. al. [20] discuss different approaches for optimizing drone mobility management (DMA). The Optimal DMA aims to maximize system spectral efficiency by discretizing turning options and selecting directions that enhance spectral efficiency. Game Theory DMA simplifies direction selection by treating it as a non-cooperative game among interfering drones. The paper defines the signal to leakage ratio (SLR) and the SLR DMA focuses on minimizing interference on active users in neighboring cells by calculating SLR while the SNR DMA maximizes signal to noise ratio (SNR) for active users without requiring inter-DBS communication or knowledge of neighboring active users' locations. Each algorithm offers a unique strategy for improving system performance in DMA. Furthermore, the application of game theory in solving challenges related to

UAV-aided networks is addressed in [21]. This survey explores the potential of game theory in modeling and analyzing problems in UAV-aided networks, providing insights into the classification and application of game-theoretic techniques for addressing challenges in these networks.

Rahmati et. al. [22] tackle interference among UAV relays through joint 3D trajectory design and power allocation. A joint optimization solution for 3D trajectory design and power allocation is proposed based on spectral graph theory and convex optimization. Two algorithms for UAVBS deployment are proposed in [23], one centralized and one distributed, both emphasizing connectivity maintenance. Fadlullah et. al. [24], introduce a dynamic trajectory control algorithm to improve communication throughput in UAV-aided networks. Bayerlein et. al. [25], use reinforcement learning for trajectory optimization of autonomous flying base stations via learning the moving UAVBS the trajectory that maximizes the capacity of transmission rate, while Shen et. al. [26], employ a joint update algorithm for trajectory and power control in UAV networks to enhance the aggregate sum rate. Amer et. al. [27] analyze 3D coverage probability in cellular UAVBS networks by proposing a coordinated multipoint transmission scheme to enhance the system performance that handles the dynamic mobility of UAVs. Sharma and Kim [28] propose random waypoint mobility and uniform mobility models to represent UAV mobility in vertical and spatial directions for controlling the trajectories of these flying nodes.

Mozaffari et. al. [29], explore energy-efficient data collection in IoT applications using UAVs by proposing a framework for deploying UAVs at a suitable position and controlling their movements. Wang et. al. [30] optimize transmit power and trajectory for UAV networks to maximize throughput. Thus, this study introduces a suboptimal algorithm for formulating non-convex optimization with trajectory constraints and a power budget. Wu et. al. [31], study optimal relay UAV positions and mobility control methods for end-to-end and multi-agent-inter communication scenarios by using gradient-based methods, whereas Pérez-Carabaza et. al. [32] utilize the minimum time search (MTS) planner, based on the ant colony optimization algorithm, in order to ensure optimized paths without physical collision among UAVs under communications constraints.

Huang et. al. [33] offer a detailed examination of collision avoidance strategies in multi-UAV systems. They categorize these approaches into conflict resolution, path planning, model predictive control, geometric guidance, potential field, and motion planning. Through a comprehensive classification based on algorithms and frameworks, the paper highlights the diversity of methodologies in this domain and emphasizes their key features. Moreover, it discusses the persistent challenges in collision avoidance for multi-UAVs.

This paper studies the problem of resource allocation in downlink communication in a system with a single cell base station and multiple UAVBSs. The contribution of this work are as follows.

First, the interactional relationship amongst multi UAVBSs is formulated as an exact potential game. The actions of this game control the mobility of the UAVBSs, with the goal to mitigate the interference generated and avoid physical collision via intelligent motion. The algorithms ensure that UAVBSs maintain distance between each other and select the direction that minimizes the interference amongst UAVBSs. The UAVBS motion is completely controlled through the utility function without a predefined safe distance that constrains the mobility of the UAVBS or contributes to avoiding the collision.

Second, three algorithms are proposed to control UAVBS mobility and obtain the convergence of the game. These algorithms differ in computational and communication complexity, leading to different convergence rates, energy efficiencies and network performance. In implementation the UAVBS can choose an algorithm based on the availability of resources, its flight speed and the rate of change of environmental conditions.

The first algorithm proposed is called the UAVBSs Better Direction Control (UAVBSs-BDC) Algorithm. In the UAVBSs-BDC algorithm each UAV performs a random walk by choosing a random direction to fly in with constant velocity in every round of the game. If the direction of flight improves the utility of the UAV, it will continue to fly in the same direction, whereas if its utility is not increased it will choose a different random direction in the next round. This way the UAVBS maintains a balance between exploring new strategies and continuing to use strategies that lead to increased utility. The UAVBSs in this algorithm do not need to communicate with their neighbors, which means each UAVBS selects its action (direction) without having any information about the location or heading of neighbor UAVBSs. Thus, this algorithm has proportional fairness between enhancing the network performance and power consumption of UAVBSs.

The second proposed algorithm is the Utility-Driven Partial Synchronous Learning (UDPSL) algorithm. This algorithm extends UAVBSs-BDC by allowing UAVs to learn from their past actions. The UAVs choose a mixed strategy with action probabilities calculated using a variation of partial synchronous binary log linear learning (PSBLLL) algorithm. Thus, this algorithm works on allowing each UAVBS to select the action that has a higher probability of maximizing the UAVBS's utility. However the algorithm requires keeping track of a large number of past actions and utilities, and requires more computation.

The last proposed algorithm is called the Neighbor Responsive Adaptive-Partial Synchronous Learning (NRA-PSL) Algorithm. In this algorithm the convergence is enhanced using additional information about the location of other UAVBSs. The players choose the direction that guides them to distance themselves from each other via allowing each UAVBS to be responsive to its neighbors and adaptive in its direction selection based on the impact of its neighbors on its own associated users. This algorithm leads the UAVBSs to track the trajectories that contribute to enhancing the whole network performance and guaranteeing the players' utility maxima that eliminate interference amongst the UAVBSs.

III. SYSTEM MODEL

This paper considers the downlink heterogeneous network, which consists of one ground base station (gNB). The gNB serves ground based users (gUE) and provides wireless backhaul service to UAVBSs. gUEs within an inner radius of 500m of the gNB are considered to have a good connection to the network, and are directly assigned to the gNB whereas terrestrial users located 500 to 1000m form gNB do not have a reliable connection to gNB. Thus, the UAVBSs' mission is to grant service to the users that are located at the edge of the network in addition to the users that are located outside the inner radius but do not get good service from the gNB, as illustrated in Fig. 1.

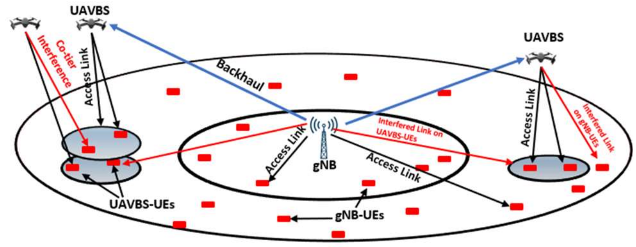

Figure 1. **The cellular heterogeneous network consists of gNB overlaid with UAVBSs in an urban environment.**

The ground users are deployed independently and identically (i.i.d.) according to a Binomial Point Process (BPP). UAVBSs are assumed to move in 2D with fixed altitude h, and the radius of coverage area for each UAVBS is 200m [34]. A dense urban cellular network is modeled where the gUE each have full buffers. This means that throughout each UAVBS serves at least one user for the majority of the simulation.

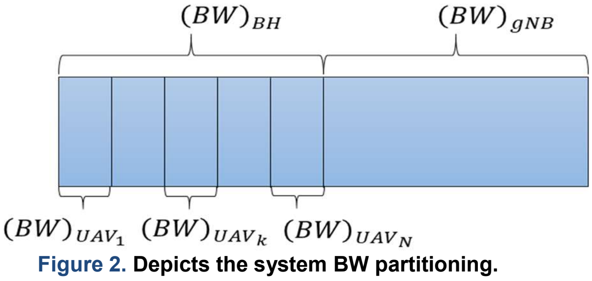

Figure 2. Depicts the system BW partitioning.

The ground users are associated with the nearest BS, either with the UAVBSs or with the gNB. Due to the height of the UAVBS, users that are located in the inner cell inside the coverage range of UAVBS mostly get service from UAVBS. Users located outside that range can access the UAVBSs or ıf no UAVBS is in range they are also connected to the gNB.

The system bandwidth is divided into two parts: the first part is allocated for the UEs of gNB, and the second part is divided equally by the number of UAVBSs within the network to allocate it to its UEs equally, as illustrated in Fig. 2.

A. RECEIVED POWERS

We assume that gNB and UAVBSs transmit with constant powers P_{gNB} and P_{UAVBS}, respectively. For consistency, we assume the transmit antenna between the gNB and the UAVBSs, and between the gNB and UAVBSs toward the ground users (access) is omnidirectional antenna and has a gain $\mathcal{G}$. Since the gNB-UE link is in an NLoS condition, the received power at the UEs from the serving gNB can be written as $P^{Rx}_{gNB-UEs} = P_{gNB}\ G\ f_{gNB}\ d^{-\alpha}_{gNB,UEs}\ \eta^{-1}_{NLOS}$, where f_{gNB} represents small scale power fading, $d^{-\alpha}_{gNB,UE}$ refers to the distance between the serving gNB and the UEs, α is the path-loss exponent, η_{NLOS} is the mean path-loss constant that depends on the wireless environment of the network for NLoS transmission [35].

Since the UAVs are flying at 200m height, the UAVBSs-to-UEs link experiences line of sight (LoS) channel conditions, therefore the received power at UE m from UAVBS p can be expressed as:

$$P_{m,p} = P_p\ G\ f_p d^{-\alpha}_{m,p} \eta^{-1}_{LOS}. \tag{1}$$

The power received at UE m from the gNB is:

$$P_{m,gNB} = P_{gNB} G\ f_{gNB}\ d^{-\alpha}_{gNB-m} \eta^{-1}_{NLOS}\ , \tag{2}$$

where, f_p is the small scale fading power between UAVBS p and gUE m, η_{LOS} represents mean excessive path loss for LoS transmission, and η_{NLOS} represents mean excessive path loss for NLoS transmission. The distance between UAVBS p and gUE m, is $d_{m,p}$. The UAVBSs are assumed to be located at a fixed altitude of h_{UAVBS}, while the gNB and gUE are assumed to be located at ground level.
The received SINR at gUE m that is associated with UAVBS p, can be expressed as follows:

$$SINR_{m,p} = \frac{P_{m,p}}{I_{m,p}+I_{m,gNB}+N_0}\ , \tag{3}$$

where N_0 refers to noise power, so it could be calculated as following $N_0^{Thermal} = B.T$, where $N_0^{Thermal}$ is thermal noise power density, $B = 1.38 \text{ x } 10^{-23}$J/K refers to the Boltzmann `, the temperature is $T = 290$ K indicates to temperature. Thus, $N_0 = N_0^{Thermal}\ BW\ N_f$ where BW is the bandwidth, and N_f is the noise figure. The total interference power received at gUE m associated with UAV p from neighboring UAVs is:

$$I_{m,p} = \sum_{j=1,j\neq p}^{P} P_{m,j}\ . \tag{4}$$

The interference power received at gUE m from gNB is $I_{m,gNB} = P_{m,gNB}$. The throughput of gUE m associated with UAVBS p is calculated from the SINR:

$$R_{\boldsymbol{m,p}} = W\log_2\left(1+ SINR_{\boldsymbol{m,p}}\right). \tag{5}$$

The capacity of each UAVBS is calculated as summation of the throughput of users that are associated with that UAVBS. Let $M_p$be the set of gUE associated with UAVBS p. Thus the capacity of UAVBS p is:

$$Capacity_p = \sum_{m\in M_p} R_{m,p} \ , \tag{6}$$

where the total capacity of UAVBSs network is calculated via summation of the capacities of all UAVBSs are expressed below respectively as:

$$Capacity_{UAVBS} = \sum_{j=1}^{P} Capacity_p \ , \tag{7}$$

where P refers to the number of UAVBSs.

B. FADING MODEL

The Nakagami-m channel model is used in this work as it can be used to model both LOS and Non-LOS propagation channels. We assume the Nakagami-m fading model for both the LoS and NLoS channels because it captures a wide variety of fading environments. Since the square of a Nakagami-m random variable is gamma distributed, the channel fading powers for serving and interfering f_{gNB}, f_{UAVBS}, are all gamma distributed [36]. For mathematical tractability, it is assumed that m is integer and the serving and interfering links have the same m values.

C. FAIRNESS PERFORMANCE

Fairness has been utilized in network engineering to determine whether the users or the applications are receiving a fair share of system resources. Fairness is a very necessary requirement for any cellular network because it is not enough to have only high system throughput without a guarantee of a fair distribution of system resources. Therefore, we have explained the fairness index that is used in our work.

Jain's fairness index is one of the most efficient measurements to determine how each user is being served in a network if they have requested service, as proposed by R. Jain et. al. [37]. The fairness at Nash equilibrium is evaluated for the proposed algorithms. The range of fairness index starts from $j = 1/N$, where $j = 1$ corresponds to maximum fairness when all users receive the same amount of resources.

$$J(t) = \frac{\left(\sum_{n=1}^{N} R_{UE}\right)^2}{N.\sum_{n=I}^{N} R_{UE}{}^2}, \tag{8}$$

where R_{UE} represents the user throughput, and N is the number of users.

D. USER ASSOCIATION AND INTERFERENCE FRAMEWORK

The association of gUE with UAVBS and gNB is assumed to be based on the nearest distance between the gUE and the UAVBSs. Since this work concentrates on downlink communication, the interference is calculated based on distances between the gUE and the base stations. The area of coverage of a UAVBS is assumed to be limited to 400m, thus any gUE which is more than 400m away from all UAVBS is associated with the gNB. Any gUE which is closer than 400m to at least one UAVBS is associated with the closest UAVBS. The power received from other UAVBS appears at the gUE as interference, which is assumed to be cumulative.

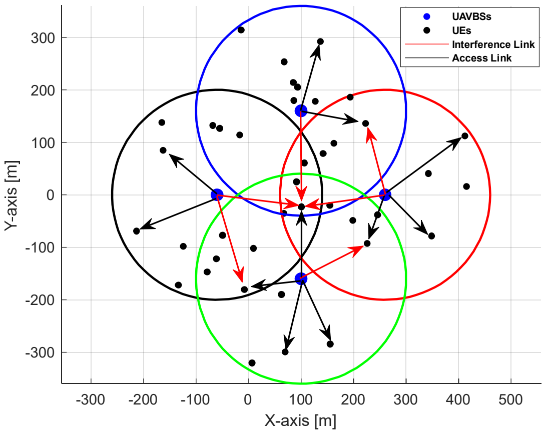

Figure 3. Illustrates the complex interference scenario amongst UAVBSs.

E. USER MOBILITY

The mobility of the ground users in this work is as follows, each user selects a random angle to move toward the selected direction each second and stop for a while, and then continues in this way. However, when these users reach the cell-edge, they return inside the cell. Thus, the users continue in their mobility until the game obtains convergence.

F. FLIGHT TIME AND POWER CONSUMPTION MEASUREMENT FOR THE UAVBSs

In this paper the power consumption of UAVs is calculated for the proposed algorithms in order to investigate the energy efficient algorithms to determine if the proposed algorithms are valid for real-world applications. The UAV used in simulations is based on the commercial quadcopter DJI Phantom 4 Pro, which is a small drone with appropriate specification for this scenario.

UAV energy consumption can be affected by many factors such as the propulsion properties of the UAV such as rotor length and motor, the weight of the UAV and the payload, whether the UAV is moving vertically or horizontally or hovering, the height of the UAV and wind speeds [38]. While in recent years several methods have been developed to model the energy consumption of UAVs, there is still a discrepancy between theoretical predictions and measured results [39]. The average energy consumed per unit distance (meter) can be calculated according to Equation (5) in Kirschtein [40]. The equation itself contains many factors which are outside the scope of this work, based on the physical characteristics of the drone such as its wind resistance and the physical properties of its rotors.

While some of the data required to calculate power consumption for the DJI Phantom 4 Pro was not available, where possible data was completed from extensive aerodynamic simulation studies from the DJI Phantom 3 and from more general data for small drone chases. The specifications used are given in Table 1.

Table 1. Phantom 4 Pro specifications [41].

Product Name	Phantom 4 Pro
Weight of UAV (including battery and propellers)	1.388 kg
Weight of Nokia BS	2 kg
Total Weight	3.4kg
Max speed	20 m/s
Battery capacity	5.870 Ah
Battery Voltage	15.2 V
Battery Energy	89.2 Wh
Rotor Radius	0.12m [42]
Rotor Mean Chord	0.025m [42]
Blade Drag	0.012 [42]
Avionics Power Consumed	0.1W [39]
Blade Lift Coefficient	0.4 [39]
Blade Drag Coefficient	0.012 [39]
Lifting Power Markup	1.15 [39]

For this paper wind conditions are set to zero, and obstacles and aerodynamic interactions between the UAVBS are ignored. The weight of the drone is also fixed. The energy used during takeoff and landing is not investigated, as this energy is fixed for all the algorithms proposed, so the angle of elevation is fixed to zero. The energy required to change direction is also ignored, thus the energy consumed depends only on the speed of the UAVBS.

The power consumption of the UAVBS as a function of velocity is shown in Figure 4 [40]. It can be seen that as velocity can give the UAV lift, more power is spent to keep the UAVBS moving slowly. However fast speeds can lead to collisions particularly when UAVBS are close together.

Given that the battery capacity of the Phantom 4 Pro is 5.87 A-h, according to the above power consumption figures the lifetime of the drone is only 7.0 min when moving at a speed of 2m/s, while it is 25.2 min when moving at a rate of 10 m/s. The manufacturer gives the drone a lifetime of about 30 min, so the power consumption model is slightly higher than advertised by the manufacturer. This difference may be due to the mismatch of the power consumption model used, or due to discrepancies between the parameters obtained from the Phantom 3 and generic UAV and the superior design of the new generation Phantom 4.

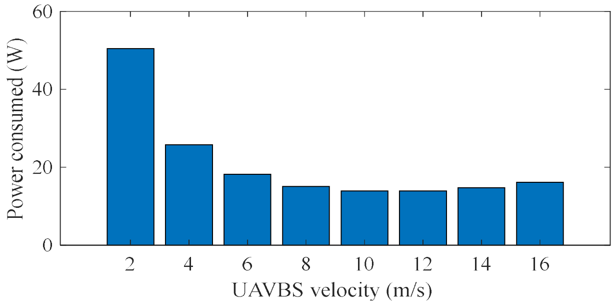

Figure 4. Power consumption of UAVBS as a function of speed.

IV. POTENTIAL GAME APPROACH

Monderer and Shapley introduced the concept of potential game, which is utilized to analyze the properties of equilibrium in games. Thus, the change in the player's action is expressed via an independent function, which is called potential function. The potential function works on determining the player's global preference over the payoffs of their actions. Thus, they have mentioned four types of potential games such as ordinal potential games, weighted potential games, exact potential games (EPG), and generalized ordinal potential games [43]. In addition, other types exist in the literature such as the best-response potential game that is proposed by Voorneveld [44], and pseudo-potential games that proposed by Dubey [45]. In our work we concentrate on EPG due to guarantee a solution which is the steady state so, we formulate the UAVBSs mobility control problem as an EPG.

Definition 1: The game $\mathcal{G}$ is considered an exact potential game if and only if a potential function $\Phi(A): A \rightarrow \mathbb{R}$ exists such that, $\forall\, p \in P_p$ [43]:

$$\begin{aligned} u_p\left(a_2^p, a^{-p}\right) - u_p\left(a_1^p, a^{-p}\right) \\ = \Phi\left(a_2^p, a^{-p}\right) - \Phi\left(a_1^p, a^{-p}\right), \\ \forall\, a_1^p, a_2^p \in A_p, \forall\, a^{-p} \in A_{-p}, \end{aligned} \tag{9}$$

where u_p represents the player's p utility function, a_1^p represents the action taken by the player p before the change of player's action, a_2^p represents the action taken by player p after the change of player's action, and a^{-p} refers to the opponent players' actions except player p. Whereas Φ represents the potential function as a function of actions of player p and opponent players' actions. Thus, we can conclude from the definition 1 that if the player p changes its action from action 1 to action 2, this change leads to a change in its utility u_p. Thus, that change in utility function u_p must be equivalent to the change in potential function Φ. In other words, the change in a single player's utility due to its own

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3515006

action deviation leads to the exact same amount of change in the potential function.

A. GAME MODELLING

The problem of UAVBSs mobility control could be represented as $\mathcal{G} = (P, \{A_p = \theta_p\}, u_{p \in P})$ where P is the set of players that refers to number of UAVBSs, A_p refers to the actions set of each player p that represent the UAVBSs' angles or directions as assumed in this work, and u_p refers to the utility function of each player p. Each player p controls its mobility via changing its directions θ_p, these directions are actions set for each player.

The actions are discretized as a set of N_θ angles $[-\theta_{max}, \dots, \theta_{max}]$ in order to make the actions finite and guarantee the existence of NE as proved in [46]. The directions are defined individually for each UAVBSs, giving the angle of motion relative to the heading of the UAVBS. Thus, the angle step options g are calculated according to the following equation $g = (2\theta_{max})/(N_\theta - 1)$. If $N_\theta$is an odd integer, $g = 0$ will be one of the options, which allows the UAVBS to continue in a fixed direction. The maximum turning angle is calculated as $\theta_{max}= (A_{max}\ t)/v, A_{max}$ is the maximum centripetal acceleration, which is adjusted in this work as 2 m/s^2, v is the speed of the player p, and t is time [47][48]. A larger value of N_θ produces a wider range of turning angles. This enhances maneuverability, which allows UAVBSs to navigate through complex environments and adjust their trajectories more precisely by enabling them to reach areas that might be challenging for UAVBSs with limited range. Selection from a broad range of angles allows UAVBSs to explore various directions, which may assist in separating each UAVBS faster from its neighbors particularly if the UAVBS is surrounded by other UAVBS. Thus, the mobility of UAVBSs will be more flexible in tight environments, making them interact with the environment more agilely. However having a large number of potential actions can slow the learning process for the UAVBS, since each direction must be experienced to determine if it generates an improvement in utility. The calculation of a player's p action takes into account the physical parameters to find the suitable turning angles that make them appropriate for the practical application.

B. FORMULATION THE GAME AS EPG

As with many congestion problems, this game $\mathcal{G}$ can also be formulated as an EPG. This is investigated by using the forward method: first the utility function is derived for the players. A potential function is derived from these utility functions, to explore whether this game satisfies one of the properties of an EPG [49].

The utility function of each player p is based on the interference generated by UAV p which impacts other UAVs and the interference generated by other UAVs which impacts UAV p. The utility of player p is sum of the interference generated by player p at all gUEs not associated with player p and the interference powers from opponents $-p$ that impact gUEs associated with player p. Let $M_p$be the set of gUE associated with UAVBS p and let M_{-p} be the set of gUE not associated with UAVBS p. Then the utility of player p is:

$$u_p(a_p, a_{-p}) = -\left(\sum_{n \in M_{-p}} P_{n,p} + \sum_{m \in M_p} I_{m,p}\right). \tag{10}$$

The utility is exactly expressed in this work as:

$$u_p(a_p, a_{-p}) = -\begin{pmatrix} \sum_{n \in M_{-p}} P_p\ \mathcal{G}\ f_p d_{n,p}^{-\alpha} \eta_{LOS}^{-1} \\ + \sum_{m \in M_p} P_{gNB} G\ f_{gNB}\ d_{gNB-m}^{-\alpha} \eta_{NLOS}^{-1} \\ + \sum_{m \in M_p} \sum_{j=1, j \neq p}^{P} P_j\ G\ f_j d_{m,j}^{-\alpha} \eta_{LOS}^{-1} \end{pmatrix}. \tag{11}$$

The first term represents the interference from player p to associated users of all players $-p$ and the gNB. P_p is the transmit power of player p, and $d_{n,p}$ refers to the distance between player p and gUE $n \in M_{-p}$ not associated to player p. When player p moves, it changes the distance $d_{n,p}$ thus changing term 1. The second term represents the interference from gNB on the associated users of player p. This term does not depend on the play and is constant as long as sets set M_p and M_{-p} are constant. The third term represents the interference from all players other than player p to gUE associated with player p. P_j is the transmit power of the opponent player $j \in -p$, and $d_{m,j}$ refers to the distance between the opponent player $j \in -p$ and gUE $m \in M_p$ associated to player p. The movement of opponent players of player p will change the distances Term 2 and term 3 are affected by the action of player p only if due to the motion of player p some gUE moves from set M_p to set M_{-p}.

The utility function shows that there is an interdependent interaction relationship among the players. This interdependence arises from the fact that players' utilities are not solely determined by their own actions but are also influenced by the actions of others. The interdependent nature of the interaction relationship highlights the complexity and strategic depth of the game, as players must navigate the interplay between their own objectives and the actions of their opponents. Thus, interdependent interactions may involve strategic elements, but they can also involve non-strategic dependencies where players' actions affect each other directly or indirectly. This is similar to the concept of strategic separability of EPG which is explained as follows:

Definition 2. The game G is strategically separable if $\forall p$, $\exists\ \mathcal{P}_p: A_p \to \mathbb{R}$ and $\exists\ \mathcal{Q}_p: A_p \to \mathbb{R}$ such that [49]:

$$u_p(a_p, a_{-p}) = \mathcal{P}_p(a_p) + \mathcal{Q}_{-p}(a_{-p}). \tag{12}$$

From Definition 2, the utility function in (11) satisfies the separability property, which is one of the EPG properties. That is, the player's utility function can be decomposed into the summation of a term affected solely by one's own action, and another term influenced only by the opponents' joint action.

$$\mathcal{P}_p(a_p) = -\sum_{n\in M_{-p}} P_{n,p} = -\sum_{n\in M_{-p}} P_p\ G\ f_p d_{n,p}^{-\alpha} \eta_{LOS}^{-1} \tag{13}$$

$$\mathcal{Q}_{-p}(a_{-p}) = -\sum_{m\in M_p} I_{m,p} = -\left(\sum_{m\in M_p} \sum_{j=1,j\neq p}^{P} P_j\ G\ f_j d_{m,j}^{-\alpha} \eta_{LOS}^{-1} + \sum_{m\in M_p} P_{gNB} G\ f_{gNB}\ d_{gNB-m}^{-\alpha} \eta_{NLOS}^{-1}\right). \tag{14}$$

Using the above expressions the utility in (11) becomes:

$$u_p(a_p, a_{-p}) = \mathcal{P}_p(a_p) + \mathcal{Q}_{-p}(a_{-p}). \tag{15}$$

Theorem 1. If the game $\mathcal{G}$ is strategically separable, then it is also an exact potential game with the following potential function [49].

$$\Phi(a) = -\sum_{p\in P} \mathcal{P}_p(a_p) \tag{16}$$

The potential function could be expressed as the sum of the interference generated by player p, as:

$$\Phi(a) = -\sum_{p\in P} \sum_{n\in M_{-p}} P_p G f_p d_{n,p}^{-\alpha} \eta_{LOS}^{-1} \tag{17}$$

It can be shown that when channel coefficients and gUE allocations between players are constant, Definition 1 is satisfied for this potential function since following (15):

$$u_p(a_2^p, a^{-p}) - u_p(a_1^p, a^{-p}) = \mathcal{P}_p(a_2^p) - \mathcal{P}_p(a_1^p). \tag{18}$$

Similarly, following (16):

$$\Phi_p(a_2^p, a^{-p}) - \Phi_p(a_1^p, a^{-p}) = \mathcal{P}_p(a_2^p) + \sum_{j\neq p} \mathcal{P}_{-p}(a^{-p}) - \mathcal{P}_p(a_1^p) - \sum_{j\neq p} \mathcal{P}_{-p}(a^{-p}). \tag{19}$$

Thus,

$$\Phi_p(a_2^p, a^{-p}) - \Phi_p(a_1^p, a^{-p}) = \mathcal{P}_p(a_2^p) - \mathcal{P}_p(a_1^p). \tag{20}$$

Hence, $\Phi(a)$ is a potential function for our game $\mathcal{G}$, and the game is an exact potential game (EPG). This means that the game is guaranteed to have at least one Nash equilibrium (NE) which is located at the optimum of its potential function.

The above derivation only holds if the association of gUE to UAVBS and gNB remains constant. However in a cellular network where the goal is to maximize the capacity of the network, gUE must be connected to the nearest BS to maximize their throughput and minimize interference. This means that the game being studied can be considered to be a dynamic game. As a gUE connects or disconnects from a UAVBS, this will change the game and the potential function. To ensure steady convergence of the game, handover of gUE occurs every five rounds of the game. This corresponds to 5 seconds in simulations, and serves to prevent frequent reassignment of gUE.

Definition 3. The action profile $(a_1^*, a_2^*, a_3^*, a_4^*)$ obtain NE if and only if [49]:

$$u_p(a_p^*, a_{-p}^*) \geq u_p(a_p, a_{-p}^*)\ \forall a_p \in A_p, \forall p \in P. \tag{21}$$

In fact it is possible to find one pure strategy and one mixed strategy NE for this game.

Lemma 1. Consider a single cell centered around a gNB with P UAVBSs associated with the network. The gUE are distributed in a uniform Poisson Point Process across the cell. Every gUE is associated with the BS from which it receives the strongest signal, so that

$$M_p = \left\{\forall m\colon \arg\max_{j\epsilon P} P_{m,j} = p\right\}. \tag{22}$$

The potential function in (17) is maximized if the UAVBS are located along the edge of the cell.

Proof. The proof is outlined in Appendix A.

When UAVBS are located at the edge of the cell, the interference from the gNB is less than interference from other UAVBS.

Lemma 2. Let L_p be the set of all gUE that are close enough to be assigned to UAVBS p rather than to the gNB.

$$L_p = \{\forall m\colon P_{m,p} \geq P_{m,gNB}\}. \tag{23}$$

In general $M_p \subseteq L_p$. The expected value of the potential function in (17), averaged over all possible placement of gUE in the cell is maximized if the sets L_j, $\forall j \in P$ are mutually exclusive.

Proof. The proof is outlined in Appendix A.

Theorem 2. Consider a single cell centered around a gNB with P UAVBSs associated with the network. The potential function in (17) is maximized if the UAVBS are located along the edge of the cell so that the coverage areas of the UAVBS L_p are mutually exclusive.

Proof. Follows directly from Lemma 1 and Lemma 2.

Theorem 3. In game $\mathcal{G}$ suppose the UAVBS are located on the edge of the cell so that the coverage areas of the UAVBS $\boldsymbol{L_p}$ are mutually exclusive. In this game, two NE strategies for the UAVBS are:

(a) Every UAVBS moves around the perimeter of the cell in the same direction (clockwise or counterclockwise).

(b) Every UAVBS moves with equal probability in every direction so that they preserve their location on the cell boundary.

Proof. The proof is outlined in Appendix A.

Corollary 1. In the limit as $N_\theta \to \infty$ Strategy (a) in Theorem 3 is a pure strategy equilibrium.

Proof. The proof is outlined in Appendix A.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3515006

C. DYNAMIC EQUILIBRIUM

The distributed algorithm involves dynamic decision-making, where players continuously adapt their actions in response to changes in the game environment. For example, in the UDPSL and NRA-PSL algorithms, when a player selects (explores) his action probabilistically and the player's utility decreases, each player adjusts his action by making a deterministic choice to improve his performance and achieve better outcomes. Thus, based on the player's actions selection process, the equilibrium of the game or the convergence may not fit neatly into traditional equilibrium concepts such as Nash Equilibrium, ε-Equilibrium, or Correlated Equilibrium. However, we could propose an equilibrium way suitable to the hybrid behavior of players in selecting their actions. The algorithm incorporates mechanisms for online learning and adaptation, allowing players to update their actions iteratively as the game progresses. This dynamic adaptation occurs in real time based on observed utilities and interactions with other players. By learning from the consequences of their actions, players refine their strategies to optimize their performance over time. This creates an adaptive equilibrium because the dynamic adaptation to real-time utility feedback is a key aspect of the algorithms proposed and distinguish them from traditional equilibrium concepts where players may not have access to such immediate feedback.

In addition, the adaptive equilibrium can be a powerful concept for analyzing and modeling strategic interactions in exact potential games. Incorporating adaptive decision-making mechanisms at the UAVBS, enhances the systems' ability to achieve optimal or near-optimal solutions in complex and dynamic game environments.

Thus, the proposed distributed algorithms exhibit robustness to uncertainty and variability in the game environment due to its adaptive nature. Players' actions evolve dynamically to cope with unexpected events or changes in the strategic landscape, ensuring stability and convergence even in dynamic and uncertain conditions. Despite the dynamic and evolving nature of the game, players' actions converge over time to achieve optimal or satisfactory outcomes, reflecting the adaptive equilibrium reached by the algorithm.

V. THE PROPOSED ALGORITHMS

A. UAVBSS BETTER DIRECTION CONTROL (UAVBSs-BDC) Algorithm

This heuristic algorithm iterates with linear time to obtain the solution for UAVBSs' mobility control problem, which is formulated above as a potential game. The UAVBSs in the UAVBSs-BDC algorithm do not need to communicate with their neighbors, which reduces the communication and computational overhead required. The players P are deployed randomly, where each player p makes its decision sequentially, and each player selects its action a_p randomly from its action set A_p .

Each player p begins its motion with a move in a random direction. It then calculates its utility function value. When the player's current utility $u_P(t)$ is better than the previous one $u_P(t-1)$, the player p continues to play the same action in the next iteration $a_p(t+1) = a_p(t)$. The aim is to continue playing an action which may lead the player in a better direction in the next iteration. This may move the UAV away from the gNB and from its opponents, resulting in mitigation of interference among UAVBSs and avoiding physical collision at the same time. However, when the player's current utility $u_P(t)$ is less than the previous one $u_P(t-1)$, that means the players may be getting closer to each other or the gNB as demonstrated in Fig. 4 so the player p should change its action in the next iteration and select $a_p(t+1)$ randomly from its action set A_p in order to increase the chance of exploring a different action that may obtain a better utility.

Algorithm 1 UAVBSs-BDC Algorithm

```
Deploy players $P$ randomly
Initialize the probability distribution of actions for each player $p$ to uniform distribution.
Initialize $a_p(0), u_p(0)$.
Each player $p \epsilon P$ asynchronously executes the following.
while $(t < N_h)$ or $(S(t) > \delta)$
    if the $u_p(t-1) \geq u_p(t-2)$
        $a_p(t) \leftarrow a_p(t-1) \begin{pmatrix} \text{UAVBS continue} \\ \text{flying in same direction} \end{pmatrix}$
    else
        $a_p(t) \leftarrow A_p(rnd)$.
    end if
    Player $p \epsilon P$ location is updated.
    If player collision occurs, game ends.
    $u_p(t)$ is updated according to (11).
    $\Phi_p(a_p(t))$ is updated according to (17).
    $S(t)$ is updated according to (25).
    if link to gNB is available
        $p$ sends interference for all connected UE.
        $p$ receives sum interference all other UE.
    end if
end while
```

To decide whether the game has converged, the changes in user utility are monitored. The algorithm controls the change in utility over the past N_c moves.

$$\overline{u_p(t-1)} = \frac{1}{N_c} \sum_{\tau=t-N_c}^{t-1} u_p\left(a_p(\tau), a_{-p}(\tau)\right). \qquad (24)$$

$$S(t) = \prod_{p\in P} \max_{t-N_c\leq\tau\leq t} \left| u_p\left(a_p(\tau), a_{-p}(\tau)\right) - \overline{u_p(t-1)} \right|. \tag{25}$$

When $S(t) < \delta$ then the game is considered to be converged. The goal of this convergence algorithm is to allow fluctuations in utility caused by exploratory actions of the UAVs while waiting for the strategies of the users and the utility to converge. Appropriate values for the parameters δ and N_c were found to depend closely on the simulation conditions. For this work $\delta = 0.02$ and $N_c = 100$ were used.

B. UTILITY-DRIVEN PARTIAL SYNCHRONOUS LEARNING (UDPSL) ALGORITHM

In the UAVBSs-BDC algorithm, when utility is not improved by the previous action, the next action is selected randomly from the set of possible actions. It is of interest to find if adding a learning algorithm to the UAVBS can improve the convergence of the system.

The UDPSL is inspired by the binary log linear learning (BLLL) that was introduced by [50], which was originally proposed [51][52]. Partial synchronous binary log linear learning was introduced by [53]. The motivation for using the UDPSL algorithm in our work comes from the positive contribution of this algorithm in relaxing the completeness and the asynchronicity that introduced by [53], where each player could explore various actions via constraining the action set, in addition to get benefits from allowing a subset of players to compete synchronously, which assists in increasing the convergence rate.

Thus, in [51] a subset of players are selected based on probability, whereas in our work, we adjust the selection of participated player p in the game based on the players' utilities, where we choose randomly only the players that have the worst utility to play the game, with a condition that the randomly selected players k must be more than one player and less than the total number of players P in order to not meet the condition of traditional BLLL and the condition of the whole synchronicity of players.

In this algorithm each player keeps a history of the N_h past uses of every action in its action space, the resulting change in the UAVBS utility and when this action was used.

So if at time t_1 player p takes action $a_p(t_1) = a_1$ it will store the time t_1 and the improvement in utility obtained after moving to this action:

$$\begin{aligned} u'_p(a_1, t_1) = {} & u_p\big(a_p(t_1) = a_1, a_{-p}(t_1)\big) \\ & - u_p\big(a_p(t_1 - 1), a_{-p}(t_1 - 1)\big) \end{aligned} \tag{26}$$

At time t_0 when player p will choose a new action, it will choose this action $a_1 \epsilon A_p$ with probability:

$$prob_p(a_1) = \frac{w_p(a_1)}{\sum_{a_n \in A, a_n \neq a_1} w_p(a_n)} \tag{27}$$

where the weight of each action is calculated as:

$$w_p(a_1) = \frac{1}{N_h}\sum_{n=1}^{N_h} \exp\left(\frac{\beta\ u'_p(a_1, t_n)}{t_0 - t_n}\right) \tag{28}$$

where β is an adjustable scaling constant and since time increases, $t_0 - t_n$ is always positive.

Algorithm 2 UDPSL Algorithm

```
Deploy players $P$ randomly
Initialize the weights $\boldsymbol{w_p(a_1)}$ of actions to 1 and the probability distribution of actions $prob_p(\boldsymbol{a_1})$ to uniform distribution for each player $p$.
Initialize $a_p(0)$, $u_p(0)$.
while $(t < N_h)$ or $(\boldsymbol{S(t)} > \boldsymbol{\delta})$
    M players with lowest utility are selected to be active player set $M$.
    for each passive player $p \in P\backslash M$
        $a_p(t) \leftarrow a_p(t-1)$ $\left(\begin{smallmatrix}\text{UAVBS continue}\\ \text{flying in same direction}\end{smallmatrix}\right)$
    end for
    for each active player $p \in M$
        if the $u_p(t-1) \geq u_p(t-2)$
            $a_p(t) \leftarrow a_p(t-1)$ $\left(\begin{smallmatrix}\text{UAVBS continue}\\ \text{flying in same direction}\end{smallmatrix}\right)$
        else
            $a_p(t) \leftarrow A_p(rnd)$.
                with action probabilities $prob_p(a_p)$
        end if
    end for
    for each player $p \in P$
        Player $p \epsilon P$ location is updated.
        If player collision occurs, game ends.
        $u_p(t), \Phi_p(a_p(t)), S(t)$ updated by (11), (17), (25).
        $w_p(a_p(t)), prob_p(\cdot)$ updated by (28),(27).
        if link to gNB is available
            $p$ sends interference for all connected UE.
            $p$ receives sum interference all other UE.
        end if
    end for
end while
```

This weight term is inspired by BLLL. When an action has never been used, its prior utility values are all zero, and its weight will be 1. When it has been used and its use has led to an improvement in utility, $u'_p(a_1, t_n) > 0$ and its weight is greater than 1. When it has been used and its use led to a decrease in utility, $u'_p(a_1, t_n) < 0$ so the weight of that term will be less than 1. The denominator term $t_0 - t_n$ causes the effect of the change in utility to be forgotten over time. If this term is large, it will reduce the effect of both gains and losses in utility, bringing the weight of the action closer to the default of 1. This policy requires the UAVBS to store $2N_hN_\theta$ numbers and perform $\mathcal{O}(N_hN_\theta)$ operations to choose a new action.

The BLLL algorithm [53] uses a temperature term $\beta = 1/T$ to balance the exploration of the search space with convergence. In the present work, following Theorem 3, Nash Equilibrium strategies may not be described using pure or even mixed strategies. As a result, the users must continue to explore and converge as the algorithm runs.

In the first iteration of the UDPSL algorithm, each player p selects random action to estimate its utility. In subsequent iterations, a subset of M players with lowest utility are selected as active players who will choose a new action according to their probability mass function $prob_p$, the remaining players play the same action as in the previous round. As in the previous algorithm, the game is said to converge when $S(t) < \delta$, where $S(t)$ is given in (25).

C. NEIGHBOUR RESPONSIVE ADAPTIVE-PARTIAL SYNCHRONOUS LEARNING (NRA-PSL) ALGORITHM

While the UDPSL algorithm converges fast, it can lead to higher probability of collision among UAVBSs. This is due mainly to the quasi-synchronous nature of the algorithm, where not every user updates its action in every round of the game. The NRA-PSL algorithm was developed to study whether additional information about the position of UAVBSs could be used to improve the collision and convergence of the game.

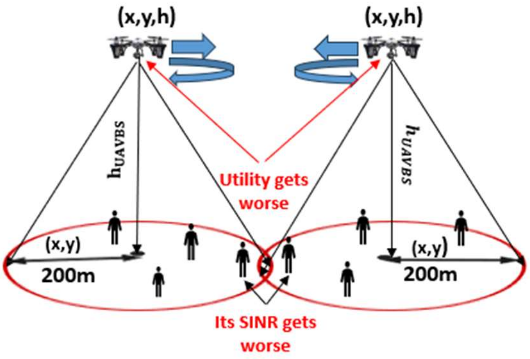

Figure 5. Depicts the changing direction of UAVBSs.

In the NRA-PSL algorithm, as with the UDPSL algorithm, in each round a subset of the M players with lowest utility will choose a new action. The action is chosen randomly with probability $prob_p$, which is updated in each round of the game based on (27) and (28). After committing the new action, the active player will compute its utility again. If there is no improvement in utility, the player p will enter its avoidant mode. In avoidant mode, the player p will identify the player j that is causing the most interference to its associated gUE. At this time it is assumed that the player p knows the location of player j. Then player p will chose an action which will move it in a direction away from player j.

The convergence criterion for this game is also given by $S(t) < \delta$, where $S(t)$ is given in (25).

Algorithm 3 NRA-PSL Algorithm

```
Deploy players $P$ randomly
Initialize the weights $w_p(a_1)$ of actions to 1 and the probability distribution of actions $prob_p(a_1)$ to uniform distribution for each player $p$.
Initialize $a_p(0)$, $u_p(0)$.
while $(t < N_h)$ or $(S(t) > \delta)$
    M players with lowest utility are selected to be active player set $M$.
    for each passive player $p \in P\backslash M$
        $a_p(t) \leftarrow a_p(t-1)$ $\binom{\text{UAVBS continue}}{\text{flying in same direction}}$
    end for
    for each active player $p \in M$
        if the $u_p(t-1) \geq u_p(t-2)$
            $a_p(t) \leftarrow a_p(t-1)$ $\binom{\text{UAVBS continue}}{\text{flying in same direction}}$
        else
            $a_p(t) \leftarrow A_p(rnd)$.
                with action probabilities $prob_p(a_p)$
        end if
        Player $p \epsilon P$ location is updated.
        If player collision occurs, game ends
        $u_p(t)$ is updated according to (11).
        if the $u_p(t) < u_p(t-1)$
            Find the player $q \in P$ which generates
                the most interference to $p$'s users.
            Player $p$ picks action $a_p(t)$ which moves
                player $p$ away from player $q$.
        end if
    end for
    for each player $p \in P$
        Player $p \epsilon P$ location is updated.
        If player collision occurs, game ends
        $u_p(t), \Phi_p(a_p(t)), S(t)$ updated by (11), (17), (25).
        $w_p(a_p(t))$, $prob_p(\cdot)$ updated by (28),(27).
        if link to gNB is available
            $p$ sends interference for all connected UE.
            $p$ receives sum interference all other UE.
        end if
    end for
end while
```

In the NRA-PSL algorithm, in addition to excess storage of $2N_hN_\theta$ numbers and $\mathcal{O}(N_hN_\theta)$ operations required for the learning algorithm, there is additional communication, computation to locate and move away from the highest interfering neighbor. Given that there are P UAVBS, the player needs to receive $P-1$ values reflecting interference from the gNB, rather than one value reflecting total interference. Then the rival with maximum interference is chosen, which requires $\mathcal{O}(P)$ computations. The UAVBS also needs to know the location of the rival UAVBS to be able to move away, which requires either monitoring of the environment to locate UAVBS or communicating with the gNB to obtain location information, and calculations to determine the strategy needed to move away from the UAVBS.

VI. CONVERGENCE ANALYSIS FOR THE PROPOSED ALGORITHMS

A. CONVERGENCE ANALYSIS of UAVBSS-BDC ALGORITHM:

The convergence of the UAVBSs-BDC algorithm is a pivotal aspect ensuring the attainment of dynamic adaptive equilibrium in the mobility control problem for UAVBSs. Due to players in this algorithm have also adaptive behavior in the process of making their decisions, where each player either makes random or deterministic choice depending on the online observation of its utility feedback. The algorithm iterates sequentially, with each iteration aimed at optimizing the players' actions towards a stable state. A thorough analysis of the convergence properties of the algorithm provides insights into its effectiveness in reaching the adaptive equilibrium.

Convergence Criterion: At the core of the UAVBSs-BDC algorithm lies a random walk. If the available actions for each player are discretized as a set of N_θ angles $[-\theta_{max}, \dots, \theta_{max}]$ where N_θ is even, this constitutes a discrete random walk in space. In each move, the player moves on one of these axes in the positive or negative direction. In the well known result from Markov random theory that when $N_\theta = 4$, the UAVBS which executes a random walk in two dimensions will reach every point along this axis with probability 1 (as time goes to infinity) and every point on the grid formed by its moves is reachable [54]. The additional directions obtained $N_\theta > 4$ move the UAVBS onto parallel grids, but these grids are offset from each other by less than $vt/2$ meters. This means that if $N_\theta > 4$ a UAVBS will reach a location within $vt/2$ meters of any point in the cell with probability 1. Since it is established that there is a NE position and strategy by Theorem 3, once the UAVBS reach the edge of the cell and their coverage areas are disjoint, the game becomes a potential game with a NE, this algorithm is guaranteed to converge.

System Stabiliy: In the context of this algorithm, the main threat to stability is UAVBS collision. Collisions can be avoided by having separate collision avoidance protocols followed by the UAVBS, however any algorithm which cause the UAVBS to be in close proximity to each other can cause dangerous flight conditions and should be avoided. Collisions can occur due to random motion of the UAVBS, but there is one scenario which the UAVBSs-BDC algorithm causes collision. This is illustrated in Fig. 6. When two UAVBS are located on one side of a gUE, they both have an incentive to pass the gUE on to the other. This is because the utility cost of gUE to UAVBS 1 when it is associated with UAVBS 1:

$$\Delta u_1{}^{\mathcal{A}} = -P_{m,gNB} - \sum_{j=2}^{P} P_{m,j} \,, \tag{29}$$

whereas when gUE is associated with UAVBS 2 its cost to UAVBS 1 is:

$$\Delta u_1{}^{\mathcal{NA}} = -P_{m,1}. \tag{30}$$

Since UAVBS 1 and UAVBS 2 are located very close to each other, it is likely that $P_{m,1} \approx P_{m,2}$ which means that

$$\Delta u_1{}^{\mathcal{A}} = -P_{m,gNB} - \sum_{j=2}^{P} P_{m,j} \approx -P_{m,1} - P_{m,gNB} - \sum_{j=2}^{P} P_{m,j} < -P_{m,1} = \Delta u_1{}^{\mathcal{NA}}. \tag{31}$$

So both UAVBS 1 and UAVBS 2 can improve their utility by handing the gUE over to the other UAV. Unfortunately in some cases the movement of the UAVs can be constrained by the cell edges, proximity to the gNB and other UAVBS so that the only way to hand the gUE over to the other UAV is to fly directly in the direction of UAVBS 2 and pass it. UAVBS 2 in turn moves towards UAV-1 in order to hand over the gUE to the other UAV. These actions lead to moving in close proximity and can lead to collision between UAVBS 1 and UAVBS 2. This condition occurs if the area area the UAVBS move towards does not have other gUE. Since the placement of gUE in the cell is dense, it is a rare event. This problem is addressed by slowing down the rate at which user handovers occur. In particular limiting handovers of gUE between UAVs to occur after several rounds of the game allows UAVBS 1 and UAVBS 2 to move past each other and into regions with more dense locations of gUE before handover occurs.

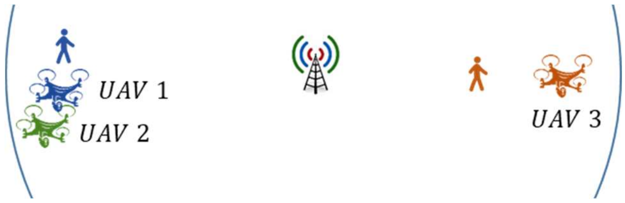

Figure 6. Depicts configurations which may cause collision between UAVBS 1 and UAVBS 2 and a quasi-equilibrium for UAVBS 3.

Convergence Rate: The convergence rate of the UAVBSs-BDC algorithm reflects the speed at which steady states are attained. The convergence of the game theoretic algorithms described in this paper are based on users' utility. In each iteration, players assess their current utility function value and compare it with the maximum value obtained in previous iterations. Convergence is achieved when no user has observed a significant change in their utility over some time. After this point, players are assumed to have achieved adaptive equilibrium, where players continuously select actions probabilistically and adjust these probabilities based on observed outcomes.

By analyzing the number of iterations required for convergence and the convergence behavior under various conditions, insights into the algorithm's convergence rate are obtained. Understanding the convergence rate informs the optimization of algorithm parameters and enhances its efficiency in reaching equilibrium states. In simulations it is found that the convergence rate depends on the random location of gUE in the cell and the random sequence of actions taken by the UAVBS in each game. Simulation results show that there are several quasi-equilibrium states for the system where one or more UAVBS are trapped in a location with a local maximum of their utility function, leading the configuration of UAVBS to remain constant over many rounds of the game. Two of these quasi-equilibria are described below.

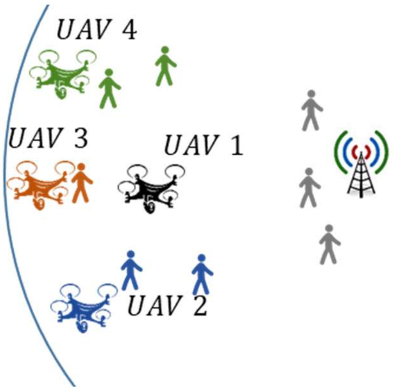

Figure 7. **Depicts a quasi-equilibrium of the UAVBSs-BDC algorithm**

One quasi-equilibrium occurs when a UAV such as UAVBS 3 in Fig 6 is located far from the other UAVBS and there is an area that has few gUEs close to the cell edge in that area. To prevent frequent handovers, the gUE can only associate with a UAVBS if the distance between them is less than some d_{min}. UAVBS 3 in Fig. 6 will move towards the cell edge since this decreases its interference to the gUE connected to the gNB and increases its utility. However at some point the gUE between UAVBS 3 and gNB will cease to associate with UAVBS3 and will associate with the gNB instead. When the gUE is handed over to the gNB, due to distance constraints $P_{m,gNB} < P_{m,UAVBS-3}$ so the interference at the gUE will increase. If the other UAVBS are very far from the gUE, then it is possible that that the utility of UAVBS 3 will decrease when the gUE is not associated with it:

$$\Delta u_3{}^{\mathcal{A}} = -P_{m,gNB} - \sum_{j=1,j\neq 3}^{P} P_{m,j} < -P_{m,3}. \qquad (32)$$

This stops UAV-3 from moving towards the edge of the cell, conflicting with Theorem 2. Theorem 2 does not hold because gUE are not perfectly uniformly distributed across the cell and the position of individual gUE can affect the convergence of the algorithm.

A second quasi-equilibrium state that emerged multiple times during simulations is depicted in Fig. 7. Here, UAV-1 is unable to increase its utility because a move in any direction increases its interference to a gUE connected to another UAVBS or the gNB. Over time, the random motion of the other UAVBS will move them out of this configuration and allow a gap to form between UAV-1 and the edge of the cell, where UAV-1 can move to maximize its utility. Since UAV-4 and UAV-2 may not have a utility increase in moving away, this can take a very long time. This configuration is not a static equilibrium for the dynamic game, but it is a frequently occurring quasi-equilibrium, which significantly affects the convergence rate, and can cause the algorithm to terminate before the potential function is maximized. Reducing the rate of handover for gUE for the system does not reduce the probability that this condition occurs and in fact may make it harder for UAV-1 to find a direction to break away from this quasi-equilibrium state.

Potential Function Maximization: A key mechanism driving convergence in the UAVBSs-BDC algorithm is the maximization of the potential function. The potential function captures the overall utility of the players' actions and serves as a metric for assessing convergence. As the algorithm iterates, potential function values are compared, and actions are adjusted to maximize utility. Convergence is achieved when the users' utility does not change significantly over a long time period. When UAVBS collide, system potential will not be maximized. Also when the system is stuck in quasi-equilibrium for a long time, the convergence test in (25) may be satisfied and the system will be assumed to have reached its final equilibrium. So the scenarios in the previous section all prevent the potential function from being maximized and interference from being minimized.

Even when the algorithm arrives at the NE described in Theorem 3, since the gUE are not perfectly uniformly distributed across the cell, the cell is not perfectly symmetric. A UAVBS can have a slightly lower utility at some locations on the edge of the cell than at others. After the UAVBS reaches the edge of the cell, it will generally move around the cell leading to variations and sometimes slight decrease in the utility of the UAVBS and thus the equilibrium potential.

While these effects are all observed for static distributions of the gUE, for systems with mobile gUE the ground distribution of gUE will be changing over time, and the utility at any position will also vary over time. In addition, due to observation that the UAVBS consume more energy by hovering in place and moving slowly, the loss in capacity due to the continuous motion of the UAVBS may not be the determining factor in the performance of the system.

Simulation results: The simulation results provide further evidence of the convergence properties of the UAVBSs-BDC algorithm. By testing the algorithm with diverse scenarios and parameter settings, its convergence behavior is evaluated under real-world conditions. Empirical testing offers practical insights into the algorithm's performance and its effectiveness in achieving adaptive equilibrium.

B. CONVERGENCE ANALYSIS OF THE UDPSL ALGORITHM

The UDPSL algorithm offers a methodical approach to achieving convergence in the context of UAVBS mobility control. The algorithm modifies the selection of a new action by using a learning algorithm based on the prior utility obtained using each action. Convergence analysis entails a detailed examination of the algorithm's iterative process and its ultimate objective of reaching a stable equilibrium state.

The UDPSL algorithm begins with the random deployment of players (UAVBSs) and initializing the probability of each action to be equal. In subsequent iterations, a subset of players with the worst utility update their actions in each round of the game. This implements partially synchronous decision-making in the game. When a player p takes an action a_p, it will update the probability of taking that action again based on whether the action led to an increase in utility or a decrease in utility using (27). The fact that the players remember the results of their past actions and that they do not change their action in every round of the game mean that players are more likely to persist in an action even if it does not increase their utility.

System Stability: The collision effect described in Fig. 6 is still equally likely to occur because players can persist in an action even when it stops increasing their utility. It is observed in simulation that the UDPSL algorithm leads to a slightly higher probability of collision.

Convergence Criterion: The persistence of actions by users means that the quasi-equilibrium states are less likely to occur. The UAVBS often move straight past such quasi-equilibria, and sometimes take shortcuts when going around the cell edge. In this way the UAVBS are much more mobile and are more likely to end up in the NE described in Theorem 3, thereby maximizing the potential function for the game. By leveraging utility-driven learning and partial synchronous decision-making, the algorithm navigates the complex dynamics of UAVBS mobility control, ultimately achieving convergence and enhancing overall system performance as shown in the simulation results.

C. CONVERGENCE ANALYSIS OF NRA-PSL ALGORITHM

The UDPSL algorithm introduces learning into the game. As observed previously this can lead to faster convergence, but can also lead to system stability issues and UAVBS collisions. The NRA-PSL algorithm modifies the previous algorithm by introducing a second phase to the game where users move away from the largest source of interference to their gUE. This leads to actions which allow the UAVBS to disperse quickly if they are trapped in a limited area of the cell and thus avoid collision. The quasi-equilibrium states described in analyzing UAVBSs-BDC algorithm also occur with the NRA-PSL algorithm. However in both the states there is a UAVBS which is trapped in a local utility maximum. In the NRA-PSL multiple movements with no improvement in utility trigger the second phase of motion of the UAVBS in a set direction, away from the largest source of interference. This can provide a way out for the trapped UAVBS, causing a swift end to the deadlock between the UAVBSs.

Ideally, the strategies were considered for the movement of a UAV to avoid collision would require simultaneous action of the UAVs [47]. However in this work the actions of each user are directions which are defined in relation to the current heading of the UAV. So for instance if the action of a UAV is to move at zero degrees, the motion is to continue in the current direction of the UAV. This means that for coordinated motion, information about the action and the direction of motion of the UAV would need to be shared and a joint decision reached between the UAV. To simplify the process, a strategy is proposed which allows acting UAV to move away from the player UAV that generates the most interference to the gUE associated with itself when the chosen strategy does not lead to an increase in utility. Short term, this reduces the received signal SINR at the gUE closest to the rival UAV. After gUE reassignment, the UAVs will be spaced further apart, increasing SINR for gUE associated with both UAVs as described in Lemma 2.

System Stability: The strategy of moving away from the most UAVBS causing the largest interference reduces the probability of collision as it increases the distance between UAVBSs.

Convergence Criterion: As with the UDPSL algorithm, the persistence of actions by users means that the quasi-equilibrium states are less likely to occur. The fact that UAVBS quickly distance themselves from each other also means that they quickly explore a larger area of the cell, making it much less likely that the quasi-equilibria situations occur.

Simulation results provide evidence of the convergence characteristics of the NRA-PSL algorithm. Thus, the algorithm guaranteed reaching a form of optimal outcome, or (players' utility maxima) for game $\mathcal{G}$. Consequently, achieving this outcome, which eliminates interference among the players (UAVBSs) can be interpreted as a successful convergence of the NRA-PSL algorithm towards an optimal state in our scenario.

VII. COMPLEXITY COMPARISON OF THE PROPOSED ALGORITHMS

The three algorithms are compared based on computational complexity, storage space and signaling complexity. Computational complexity description emphasizes the evaluation of action combinations in addition to the general computational complexities. Signaling complexity refers to the communication overhead required for UAVBSs to exchange information and coordinate their actions effectively within a dynamic mobility environment. Below, the complexities of each algorithm are discussed, highlighting the specific features within each algorithm that contribute to complexity reduction.

The UAVBSs-BDC algorithm has the lowest computational complexity. In each round of the algorithm each UAV will choose a direction at random. In the next round, motion continues in this direction if the utility improvement is positive, the UAV chooses a new direction if the previous step did not lead to an improvement in utility. The algorithm computational complexity is constant and does not scale with the number of UAVBS in the cell because of the simplified decision-making process. This leads to faster reaction time and lower power consumption.

The algorithm requires only the cumulative interference at all nodes connected to itself and the cumulative interference from it to all nodes not connected to itself. This data can be collected between UAVBS in one-to-one communication. However it can also be collected by a central system, which could be the gNB, one of the UAVBS or a separate central coordinator. For this paper, it is assumed that the gNB collects and disseminates this information. The players do not communicate directly with each other. This reduces the signaling burden and simplifies coordination, enhancing the algorithm's scalability and real-world applicability.

Every UAVBS collects interference data during its communication with the gUE and transmits the sum of this data to the gUE. Obtaining the association of each gUE and interference levels at all gUE will require communications from gUE to the BS it is attached to, totaling $|M_P|$ numbers, where $|M_P|$ denotes the number of gUE associated with player P. Since the system has N_{UE} gUE, this corresponds to a signaling overhead of $\mathcal{O}(N_{UE}/P)$, although this data may already be collected for other purposes in the system. These numbers also need to be summed, leading to $\mathcal{O}(N_{UE}/P)$ processing cost at the UAVBS. Finally every UAVBS will transmit the total interference data for its associated gUE as one number to the gNB. In short the UAVBSs using the UAVBSs-BDC algorithm have computational complexity and signaling overhead of $\mathcal{O}(N_{UE}/P+1)$.

The UDPSL algorithm has higher computational complexity since it implements a learning algorithm at each UAVBS. Each UAVBS must store of $2N_hN_\theta$ numbers as historical data to use on the learning algorithm and $\mathcal{O}(N_hN_\theta)$ operations required to calculate the probability distribution used to select the next action for each UAVBS. In simulations, the number of available actions N_θ was set to a smaller number to hasten the convergence of the algorithm. If $N_\theta$is chosen to be larger, then it may be necessary to use every action to update probabilities not just of the action taken but of closer directions to hasten the convergence of the algorithm. This is not explored in the current work.

In UDPSL in each round of the game only a subset of players are actively updating their strategy. This effectively reduces the computational load per user. However since an inactive player is still moving, its interference to gUE not associated with it will change even in rounds of the game where the player does not update. This means that the signaling overhead is the same as the UAVBSs-BDC algorithm. Thus the computational complexity of the UDPSL algorithm is $\mathcal{O}\left(p_{active}(N_{UE}/P+N_hN_\theta+1)\right)$ where p_{active} is the portion of UAVBS which are active in each round of the game. The signaling overhead is $\mathcal{O}(N_{UE}/P+1)$ as with the UAVBSs-BDC algorithm.

The NRA-PSL algorithm requires the highest computational and communication complexity of the three algorithms because the design and construction of this algorithm involve evaluating action combinations, making informed decisions based on the current situation, and interactions with neighboring players. Since it implements the same learning algorithm as UDPSL, in addition to the computational complexity requirement of $\mathcal{O}(N_hN_\theta)$ calculations to evaluate probability, it must receive $P-1$ values reflecting interference from the other UAVBS, and perform an $\mathcal{O}(P)$ ranking algorithm to determine thei highest interference. Subsequently the UAVBS also needs to know the location of the rival UAVBS to be able to move away. This is an additional number transmitted in many rounds of the algorithm. The computational complexity of the NRA-PSL algorithm is $\mathcal{O}\left(p_{active}(N_{UE}/P+N_hN_\theta+P+1)\right)$ where p_{active} is the portion of UAVBS which are active in each round of the game. The signaling complexity is $\mathcal{O}(N_{UE}/P+P+1)$. However the excess computational load required may be even more, as calculating the correct direction to move away from another UAVBS may not be very easy.

The NRA-PSL algorithm has higher computational and communication requirements than the UDPSL algorithm, however it leads to fewer collisions between UAVBS and faster convergence. The UAVBSs-BDC algorithm requires fewer communication and computational resources, but takes longer to converge than either UDPSL or NRA-PSL. Its rate of collision is between the two algorithms. Each algorithm offers unique advantages in addressing mobility control problems and has the ability to evaluate combinations of actions and adapt to dynamic environments. Thus, understanding the features of the proposed algorithms that contribute to reducing their complexities makes these algorithms align with their applicability in the real world.

VIII. SIMULATION RESULTS

The simulation in our work is carried out by a computer with 32 GB RAM and Intel Core i7 2.60GHz processor, and 64-bits Windows 11 operating system and by using the MATLAB 2019a as a platform. Thus, all the proposed algorithms in this work have been applied on 4 UAVBSs network. The users are deployed randomly uniformly inside the whole network, these users are updating their directions each time step to make stochastic dynamic, which may create a mobility similar to the real-world users' mobility. Simulation parameters are summarized in Table 1. Fig. 8-Fig.13 illustrate the performance of the algorithms for a single scenario with fixed gUE and UAVBS starting locations. The remainder of the section discusses aggregate results from simulations. Simulations were performed using 100000 different gUE starting positions, but with the same initial positions for the UAVBS to provide a basis to compare the results. Since the UAVBS may be launched from a central platform, a starting position where the UAVBS are closely spaced was considered realistic.

Fig. 8 shows the initial deployment of the UAVBS in the top left and the trajectory the four UAVBS for the three algorithms when UAVBS velocity is 2m/s. For all algorithms the final locations for the UAVBSs are on the cell boundary, moving counterclockwise.

Both the UAVBSs-BDC and the UDPSL algorithm terminate in similar last positions, but the UDPSL algorithm terminates after 1590 rounds of the game, whereas as the UAVBSs-BDC algorithm achieves the same result after 4438 epochs. In the UAVBSs-BDC algorithm, the UAVBSs have moved around more before reaching their final positions. For the NRA-PSL algorithm, the UAVBS have migrated to almost opposite corners of the cell, following the same path. It is not clear from the figure that the blue UAVBS has completed a complete circular navigation of the cell along the cell boundary, while the other UE follow it to preserve the Nash equilibrium.

Fig. 9 shows the changes in the potential function and the changes in total network capacity as a function of the iterations of the game. The potential function reflects the interference in the network, a higher potential value signifies less interference. Although it appears based on the potential function that the final results of the algorithms are close, the network capacity shows that for small levels of interference even a small decrease can increase network capacity.

Fig. 10 shows the changes in the utility for the four UAVBS during the simulation. The utility function as defined in (10) refers to the summation of the experienced and generated interference by each player.

Table 2. Simulation Parameters

Param.	Description	Value
P	Number of UAVBS	4
N_{UE}	Number of gUEs inside the whole network	500
R_{BS}	gNB cell radius	1km
P_{gNB}	gNB transmit power	46 dBm [55]
P_{UAVBS}	UAVBS transmit power	30 dBm [56]
v_{UAV}	UAVBS's Velocity	[2,4,6,8] m/s
v_{UE}	User's Velocity	1.35 m/s
Nf	UE Noise figure	9 dB [55]
BW	System bandwidth	20 MHz
h	Altitude of UAVBS	100 m
η_{NLOS}	Mean excessive path-loss for NLoS	20 [35]
η_{LOS}	Mean excessive path-loss for LoS	1 [35]
t	Time step	1 s
m	Shape factor	1 for Rayleigh and 3 for Rician
α	Path-loss exponent	2 for LoS and 3 for NLoS
β	Scaling factor for learning algorithms	175000
N_{θ}	Max candidate directions	8

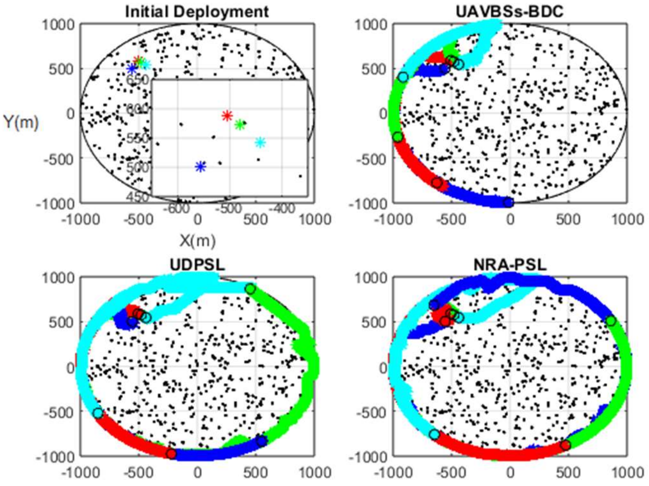

Figure 8. Mobility of UAVBSs for the Initial Deployment and Final Deployment (reaching Convergence) at Predetermined Fixed Positions at 2m/s.

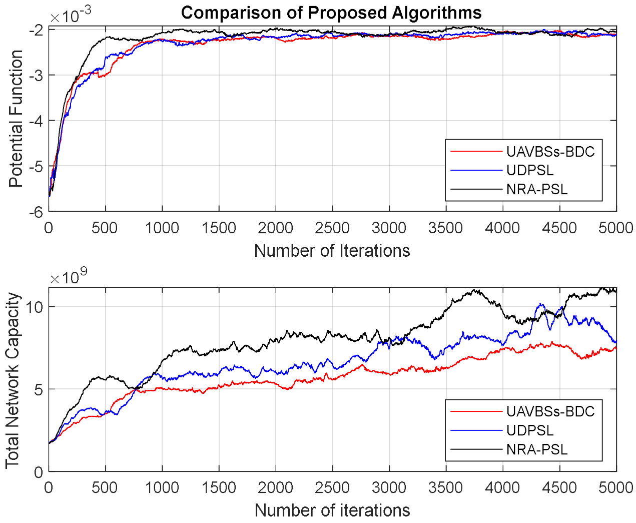

Figure 9. **Potential function and network capacity for the proposed algorithms at 2 m/s.**

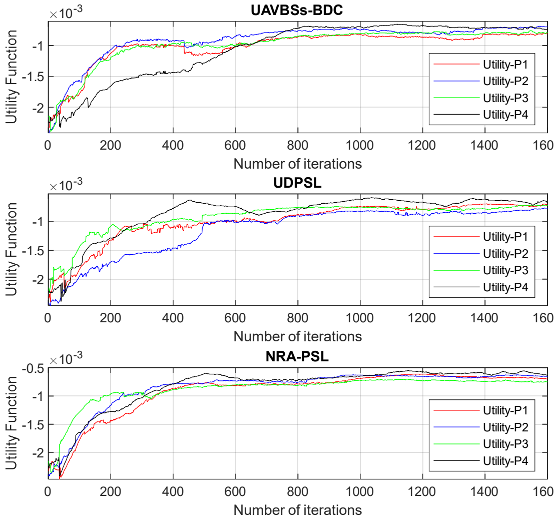

Figure 10. **The utility function of UAVBSs for all proposed algorithms at 2 m/s**

As demonstrated in these figures, the utility increases as the simulation runs, which indicates that the trajectories of UAVBSs are optimized by the dynamic motion of these UAVBSs via controlling the selected actions that lead each UAVBS to choose the direction that assists in mitigating the interference among these flying nodes. It can be seen that at steady state there is greater variance between the utility of different UAVBS when the UAVBSs-BDC algorithm is used, compared to the UDPSL and NRA-PSL algorithms.

Fig. 11 tracks the motion of the UAVBSs when the velocity is increased to 8 m/s. It can be seen that there is more random movement for all the algorithms, but all three algorithms lead to better final locations for the three algorithms.

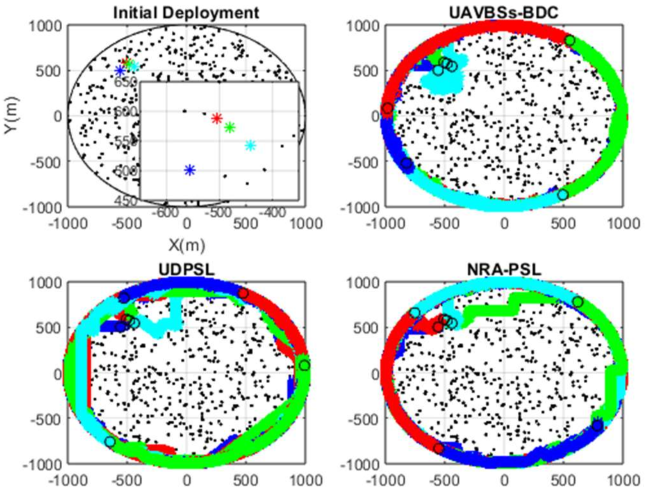

Figure 11. **Mobility of UAVBSs from the Initial Deployment until Final Deployment (reaching Convergence) at Random Positions**

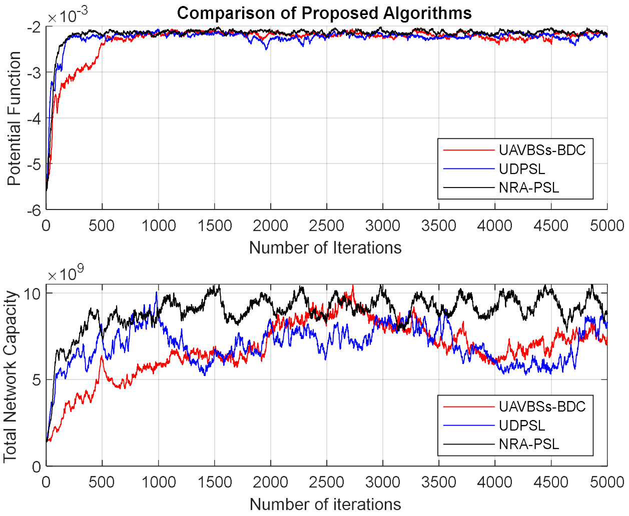

Figure 12. **Potential function and network capacity for the proposed algorithms at 8 m/s.**

Fig. 12 shows the potential function and the total network capacity for the duration of the simulation where UAVBS speed is 8m/s. Again it is clear that even small fluctuations in the potential function can lead to significant differences in the SINR and thus the network capacity. It is clear that the UDPSL and NRA-PSL algorithms converge to higher throughput faster than the UAVBSs-BDC, and as the system reaches steady state the highest throughput is in the network achieved by NRA-PSL, then UDPSL. The UAVBSs-BDC seems to achieve lower throughput in most iterations of the game.

Fig. 13 shows how each UAVBS' utility changes during the simulation. The utility for UAVBS p is the negative of the sum of the interference from the UAVBS to gUE associated with other BS and the interference from other UAVBS to gUE associated with UAVBS p. As with the lower velocity, the variance between the UAVBS's utility is higher for the UAVBSs-BDC algorithm.

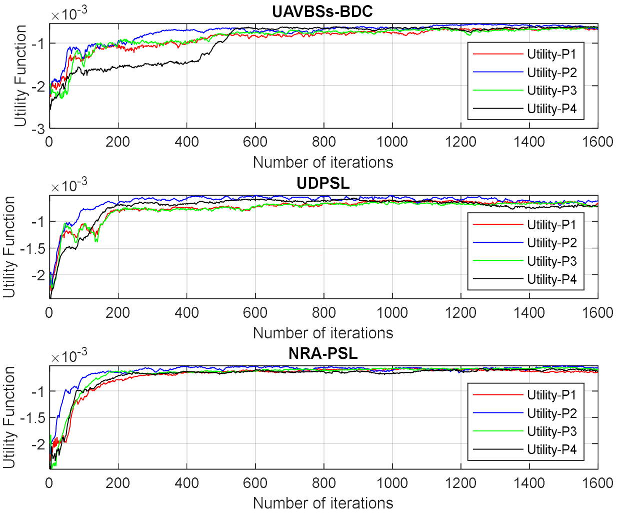

Figure 13. The utility function of UAVBSs for all proposed algorithms at 8 m/s

The results indicate that once the system reaches NE, it may be advantageous for the UAVBS to slow down to reduce the fluctuations in capacity and to reduce the number of handovers required of the gUE.

After studying these simulations in detail, to arrive at more general conclusions about the performance of the algorithms the algorithms were run for many different initial placements of the gUE. The starting positions of the UAVBSs were kept fixed, since in a real system the UAVBS are likely to be launched within a small area and the UAVBS are highly mobile during the execution of the algorithms. This placement is also challenging since the UAVBS are close together so they have high levels of interference and a high possibility of coming within collision range.

An initial 100 simulations led to an estimation of the mean and variance of the final value potential function to be $\mu = -2.50 \times 10^{-3}$ and $\mu = 1.66 \times 10^{-3}$. Thus to achieve statistically meaningful results, it was determined that the final value potential function should be averaged over 10000 different topologies. The following data was collected over 10000 runs of every algorithm for every speed. Unless otherwise specified, scenarios which resulted in UAVBS collision were not included in the data.

Fig. 14 shows that for all speeds of the UAVBS the NRA-PSL algorithm, which allows UAVBS to actively avoid each other has the lowest probability of collision, while the UDPSL algorithm which implements learning but has no collision avoidance mechanism has the highest probability. From the results in Fig. 15, the majority of collisions occur in the first 500 iterations of the algorithm, with UDPSL registering collisions after 200 iterations of the game. The later collisions in UDPSL can occur due to conflicts described in Section VI. Collisions can only occur if UAVBS are close, this suggest that an adaptive move from NRA-PSL when the UAVBS are clustered close together to UDPSL when they are spaced far apart may prevent collision while reducing the computational overhead of the system. In a real system, there will be collision avoidance mechanisms operating outside of these algorithms, but it is better both for collision avoidance and to reduce interference for UAVBS not to approach each other during the allocation algorithm.

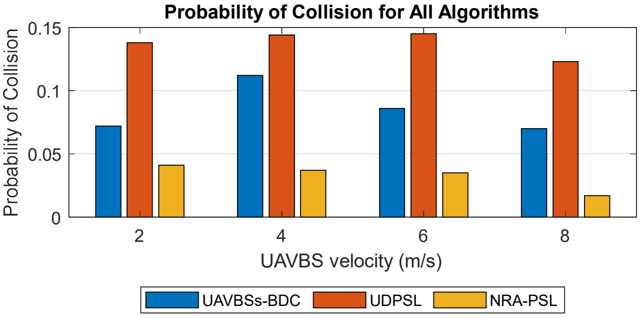

Figure 14 The comparison of the UAVBSs collision probability for all proposed algorithms.

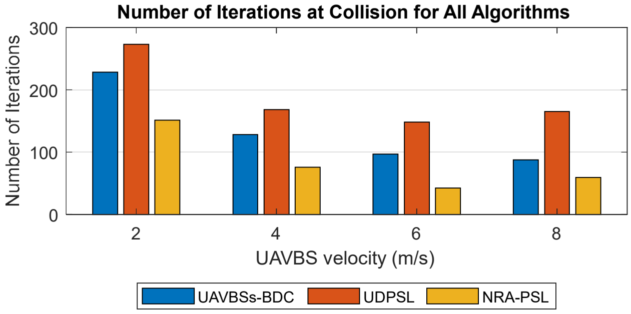

Figure 15 The average iteration at which collision occurs for for all proposed algorithms.

Fig. 16 shows the average user SINR and Fig. 17 the network capacity of the UAVBSs for the algorithms at the end of simulation. The final average capacity is shown to be lowest for the UAVBS-BDC algorithm and highest for the NRA-PSL algorithm, which verifies the results in Fig. 9 and Fig. 12. The network capacity also seems to increase slightly with speed, perhaps because it is more difficult for the UAVBS to break out of quasi-stable states at low speeds.

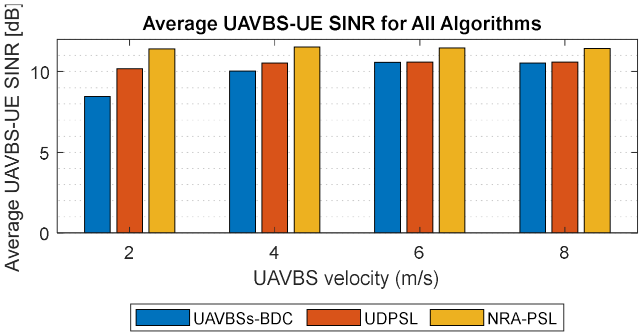

Figure 16 **The user SINR for the proposed algorithms and random deployment corresponds UAVBS's velocity.**

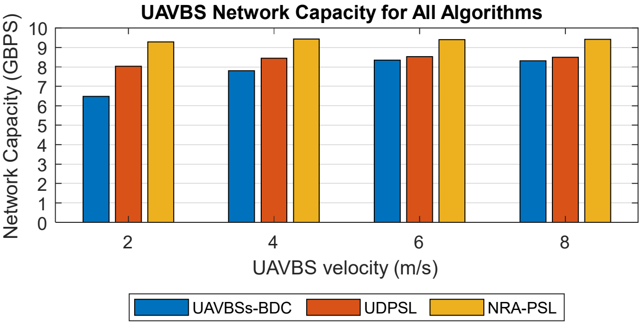

Figure 17 **The comparison of the UAVBSs network capacity for all proposed algorithms.**

Fig. 18 shows the number of iterations per simulation before the algorithms reach final capacity. It was difficult in simulations to find a suitable value threshold δ for $S(t)$ in (25) which prevented simulations from terminating in a quasi-equilibrium state defined in Section VI while allowing the game to terminate. In fact, since the number of directions is limited, Corollary 1 does not apply and there is no pure strategy NE that the system could converge to. For practical reasons the number of iterations was limited to 5000.

It can be seen from Fig. 9 and Fig. 12 that even when the potential value becomes small and fluctuations in this value become smaller, there can still be small increases in the average potential function, which can lead to significant improvements in capacity. It was also observed in simulations that this was not true for every scenario and for some scenarios the potential function could reach a peak and slowly decrease over time, particularly for the UDPSL and NRA-PSL algorithms, depending on the gUE distribution. To compare the three algorithms in terms of the number of iterations, the data was processed after simulations by finding the first iteration at which the potential function arrived within 2% of the final value. For the slowest speed of 2m/s it has already been observed that the network capacity is lowest, and this was achieved with the largest number of iterations per game as the UAVBSs are slow. The system converges on average with a smaller number of iterations when the speed is higher.

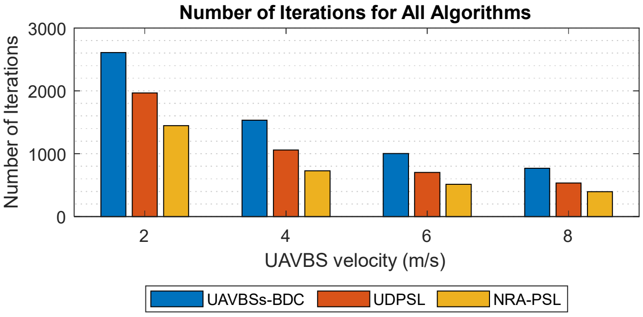

Figure 18 **The average number of iterations for all proposed algorithms.**

Fig. 19 demonstrates a comparison of the average energy consumption of the UAVBSs network at the convergence iteration, given in Fig. 18. The total power consumption for each UAVBS at each time is calculated according to equation (5) in Kirschtein [40]. Since flying at higher speeds reduces both the power required for the flight and the number of iterations required to converge, higher speeds up to 8 m/s lead to lower power consumption. Since at slower speed convergence is also slower makes the energy consumed at slower speeds even larger. In fact convergence when the UAVBS speed is 8m/s requires 1/5th of the power for the NRA-PSL algorithm and 1/10th the power for the UAVBSs-BDC algorithm as convergence when the UAVBS speed is 2m/s.

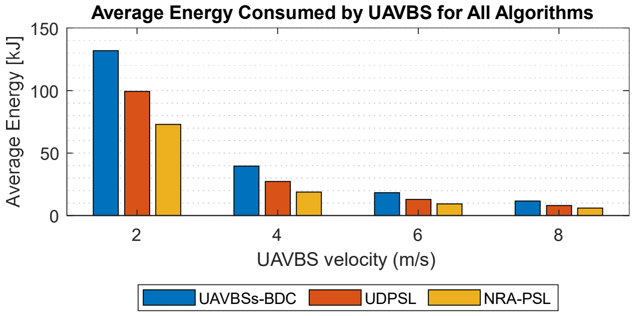

Figure 19 **The average energy consumptions of UAVBSs for all algorithms corresponding to UAVBS's velocity.**

Between the three algorithms, NRA-PSL is the most energy efficient due to its fast convergence and UAVBSs-BDC is the least energy efficient. For this result, the energy required to perform computations for the three algorithms is considered to be the same. The energy consumption of the NRA-PSL algorithm for computation will be higher than that required for the other algorithms, however the energy required for flight and motor control is likely to be significantly higher than the additional computational cost of the algorithm.

Fig. 20 depicts the average fairness index of the UAVBSs network for all algorithms corresponding to the UAVBS's velocity. The fairness index compares the achieved bitrate for each user with bitrates assigned to other users. If the values are almost the same, the fairness index is closes to 1. In this case the fairness index for the algorithms are all above 90%. Varying velocity has no direct impact on the fairness index. The NRA-PSL algorithm provides the highest fairness among the users of the UAVBSs network compared to the other algorithms. In general fairness is higher when the throughput of the network is higher. Algorithms provide more fairness as long as the performance of the network is enhanced, while in case of any degradation in network performance, the average fairness index deteriorates.

The fairness index is calculated only for gUE associated with the UAVBS. The gUE associated with the gNB have vastly lower throughput, due to the NLOS transmission from the gNB to the gUE, the fact that even users at the edge of the cell will connect to the gNB if they are not in the coverage range for the UAVBS and because while the bandwidth is shared equally between the UAVBS and the gNB the total number of users assigned to the UAVBSs is much fewer than the number of gUE assigned to the gNB.

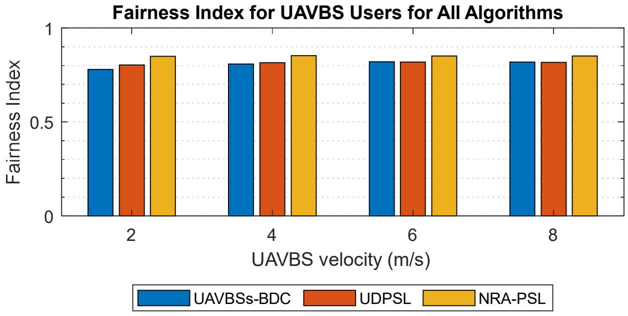

Figure 20 **Fairness index for all proposed algorithms**

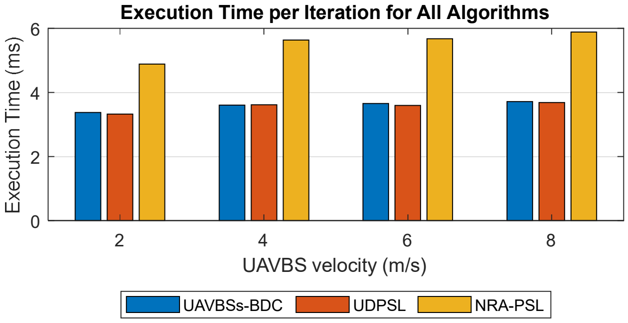

Figure 21 **The real-time computation for all algorithms.**

Fig. 21 shows us the real-time computation per iteration for each algorithm. The figures were obtained from an implementation in Matlab. They also contain processing tasks that would not be required in a real implementation on UAV, for instance the system has to calculate the next location for each UAVBS whereas in real life the next location is automatically obtained when the UAVBS moves. However this data can still give some idea about the computational complexity of the algorithms.

Fig 21 shows that the computational complexity of the UAVs-BDC and UDPSL algorithms does not change significantly with velocity, the fluctuations seen appear to show random changes. UDPSL is approximately the same complexity as UAVBSs-BDC, which is not expected as it implements a learning algorithm. This may be due to the base processing involved in the simulations. NRA-PSL however is significantly more expensive, as computing the opposite direction to the largest interferer takes some calculations. The computational requirements for NRA-PSL also seem to increase with UAVBS velocity, perhaps because the distancing mode is called more often. The total computational cost may be found by multiplying the number of iterations until steady state, shown in Fig. 18 by the execution time per iteration in Fig. 21.

IX. CONCLUSION

This work addresses the problem of communication interference and physical collision amongst dynamic UAVBSs network by controlling their mobility while maximizing communication throughput in the network. Three algorithms have been proposed: UAVBSs-BDC, UDPSL and the NRA-PSL algorithms, aimed at addressing the UAVBSs mobility control problem with various levels of mobility and communication abilities. These algorithms complement each other by offering different perspectives and strategies for solving the same problem. The UAVBSs-BDC algorithm concentrates on better direction control with minimal computational and communication overhead. The UDPSL algorithm combines elements of binary log linear learning with partial synchronous learning. The NRA-PSL algorithm adds neighbor responsiveness and allowing players to enhance their actions, which allows the algorithm to converge faster, but at the cost of greater communication and computation overhead. Thus, proposing these algorithms together, contributes to exploring different avenues for addressing the mobility control problem.

Based on the insights gained from this study, upcoming works may explore and investigate this work further by using machine learning techniques to improve decision-making; for instance, reinforcement learning with Q-learning could be applied to assist UAVBSs, learn the optimal direction over time, or adapt to dynamic environments.

APPENDIX A

Proof of Lemma 1 Suppose a UAVBS-1 is located a distance r_b from the cell center. Without loss of generality, assume that the UAV is located along the x coordinate axis and assume that all gUE located within a distance r_0 of the drone are assigned to the drone. This is represented as the red circle in Figure 8. The interference The distance between the drone and a gUE at location $(r_m \cos\theta_m, r_m \sin\theta_m)$ is $|\vec{d}| =$

$\sqrt{{r_m}^2 + {r_b}^2 - 2 r_b r_m \cos\theta_m}$. Suppose the UAV then moves a distance Δ towards the closest point of the cell edge. The line AB is a line which passes through the drone and instersects the cell edges at A and B. For gUE that are between line AB and the cell edge, $|\vec{d}|$ will decrease, resulting in more interference. However for all gUE located on the gNB side of line AB interference will decrease. Since gUE are distributed uniformly across the cell, and interference is inversely related to distance when all other factors are constant, the total expected interference from this UAV to all users in the cell not connected to the UAV will decrease as the UAV moves towards the edge of the cell. Thus interference is minimized when UAVs are located on the edge of the cell.

Proof of Lemma 2 Suppose UAV 2 and UAV 3 are located so that their coverage areas overlap. Both UAVs are located some distance away from all other UAVBS and the gNB. When UAV 3 moves away so that the overlap in coverage area is eliminated, the net interference from UAV 3 to all users outside its coverage area does not change. However consider the gUE k such that $k \in L_{UA}$ and $k \in L_{UAV3}$ but $P_{m,UAV2} \geq P_{m,UAV3}$ so that gUE k is associated with UAV 2, $k \in M_{UAV2}$. If UAV 3 moves away from UAV 2 and thus gUE k, its interference to gUE k will decrease. Its interference to other gUE will not increase by the same amount, since any gUE receiving the same amount of power from UAV 3 will be associated with UAV 3. This means that moving away from the neighbor UAV will decrease the net interference generated by UAV 3. Similarly consider the gUE l such that $l \in L_{UA}$ and $l \in L_{UAV3}$ but $P_{m,UAV3} \geq P_{m,UAV2}$ so that gUE l is associated with UAV 3, $l \in M_{UAV3}$. If UAV 3 moves away from UAV 2, gUE l will be reassigned to UAV 2. Previously the interference to gUE l from UAV 2 was $P_{m,UAV2}$. When UAV 3 moves away the interference will be $P'_{m,UA} \leq P_{m,UAV2}$ decreasing the net interference on this gUE. Since the interference of some gUE is decreased after the move, and the interference does not increase for any UAV, moving the UAV to prevent overlap of the coverage regions always increases the potential function in (17) and the function is maximized if the sets $L_j,\ \forall j \in P$ are mutually exclusive.

Proof of Theorem 2: The motion of any UAV can be broken down into two components, motion towards or away from the center of the cell and motion along a fixed distance to the cell. From Lemma 1 motion towards the gNB will only decrease utility for each UAVBs. The UAVBS cannot move outside the cell boundaries, so motion away from the gNB is not possible. From Lemma 2, when UAVBS preserve the distance between them the utility for all UAVBS will be constant. The first strategy will preserve the distance between the UAVBS. It is possible for one or more UAVBS to deviate from the strategy to move slower or faster, but this will not increase their utility. It may decrease their utility if their position relative to the other UAVBS changes so that their coverage regions overlap, breaking Lemma 2. In the second strategy the UAVBS will remain in the same position relative to each other and the gNB. Similarly to the previous case, the UAVBS can move tangentially along the cell edge but this will lead to constant utility.

Proof of Corollary 1: A UAVBS will be in NE as described in Theorem 2 (a) if in every move it moves from the cell edge to another point on the cell edge. Thus its moves describe chords of length vt on the cell boundary, where v is the velocity of the UAVBS and t is the between changes in direction. Geometrically this requires all UAVBS to use the strategy $a_p = \tilde{\theta} = 2\,\mathrm{asin}\big(vt/(2R_{BS})\big)$ to move clockwise around the cell edge and $a_p = -\tilde{\theta}$ to move counterclockwise. In the limit as $N_\theta \to \infty$ all strategies such that $\pm\tilde{\theta}$ will be part of the set of permissible strategies, meaning that the pure strategy where all UAVBS take action $a_p = \tilde{\theta} = 2\,\mathrm{asin}\big(vt/(2R_{BS})\big)$ and one where UAVBS take the action $a_p = -\tilde{\theta} = -2\,\mathrm{asin}\big(vt/(2R_{BS})\big)$ in every round of the game form pure strategy NE of the game, since these strategies satisfy the conditions for Theorem 3.

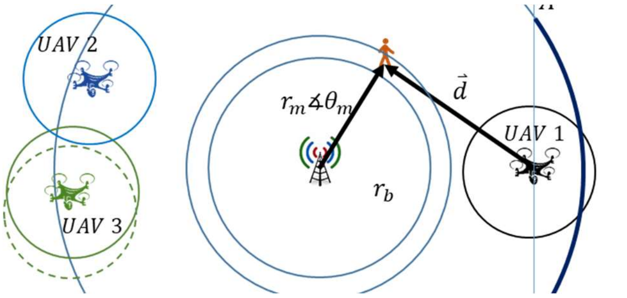

Figure 22. Illustrates the complex interference scenario amongst UAVBSs.

REFERENCES

[1] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019, doi: 10.1109/COMST.2019.2902862.

[2] D. Liu *et al.*, “Task-Driven Relay Assignment in Distributed UAV Communication Networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11003–11017, 2019, doi: 10.1109/TVT.2019.2942095.

[3] Q. Wu, Y. Zeng, and R. Zhang, “Joint trajectory and communication design for multi-UAV enabled wireless networks,” *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 2109–2121, 2018, doi: 10.1109/TWC.2017.2789293.

[4] M. Russon, “Nokia and EE trial mobile base stations floating on drones to revolutionise rural 4G coverage,” *International Business Times*, pp. 7–10, 2016.

[5] T. Rains, “Take a look at AT&T’s ‘Flying COWs’ — drones that returned cell service to Hurricane Ida-hit Louisiana,” *Business Insider*, 2021. [Online]. Available: https://www.businessinsider.com/flying-cows-bring-cell-service-to-areas-impacted-natural-disasters-2021-8

[6] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, “Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches,” *IEEE Access*, vol. 8, pp. 105139–105155, 2020, doi: 10.1109/ACCESS.2020.3000064.

[7] Z. Han, A. L. Swindlehurst, and K. J. R. Liu, “Smart deployment/movement of unmanned air vehicle to improve connectivity in MANET,” in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, IEEE, 2006, pp. 252–257. doi: 10.1109/WCNC.2006.1683473.

[8] Zhu Han, A. L. Swindlehurst, and K. Liu, “Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3533–3546, Sep. 2009, doi: 10.1109/TVT.2009.2015953.

[9] F. Jiang and A. L. Swindlehurst, “Optimization of UAV Heading for the Ground-to-Air Uplink,” *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 993–1005, Jun. 2012, doi: 10.1109/JSAC.2012.120614.

[10] D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, “A Dynamic Trajectory Control Algorithm for Improving the Probability of End-to-End Link Connection in Unmanned Aerial Vehicle Networks,” in *Personal Satellite Services. Next-Generation Satellite Networking and Communication Systems. PSATS 2016.*, vol. 148, I. Bisio, Ed., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 148. , Cham: Springer International Publishing, 2016, pp. 94–105. doi: 10.1007/978-3-319-47081-8_9.

[11] M. Zhu, Y. Chen, Z. Cai, and M. Xu, “Using Unmanned Aerial Vehicle chain to improve link capacity of two mobile nodes,” in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, IEEE, Aug. 2015, pp. 494–499. doi: 10.1109/ICMA.2015.7237535.

[12] Y. Zeng, R. Zhang, and T. J. Lim, “Throughput Maximization for UAV-Enabled Mobile Relaying Systems,” *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016, doi: 10.1109/TCOMM.2016.2611512.

[13] A. Fotouhi, M. Ding, and M. Hassan, “Dynamic base station repositioning to improve spectral efficiency of drone small cells,” in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, IEEE, Jun. 2017, pp. 1–9. doi: 10.1109/WoWMoM.2017.7974285.

[14] L. Shi, N. J. Hernandez Marcano, and R. H. Jacobsen, “Multi-UAV Path Coordination through Generalized Potential Games,” in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*, IEEE, Feb. 2022, pp. 125–129. doi: 10.1109/ICARA55094.2022.9738578.

[15] J. Chen, Y. Xu, Q. Wu, Y. Zhang, X. Chen, and N. Qi, “Interference-Aware Online Distributed Channel Selection for Multicluster FANET: A Potential Game Approach,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3792–3804, Apr. 2019, doi: 10.1109/TVT.2019.2902177.

[16] L. Ruan *et al.*, “Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework,” *China Commun.*, vol. 15, no. 10, pp. 194–209, Oct. 2018, doi: 10.1109/CC.2018.8485481.

[17] H. Kim, J. Park, M. Bennis, and S.-L. Kim, “Massive UAV-to-Ground Communication and its Stable Movement Control: A Mean-Field Approach,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, Jun. 2018, pp. 1–5. doi: 10.1109/SPAWC.2018.8445906.

[18] X. Zhong, Y. Guo, N. Li, and S. Li, “Deployment Optimization of UAV Relays for Collecting Data From Sensors: A Potential Game Approach,” *IEEE Access*, vol. 7, pp. 182962–182973, 2019, doi: 10.1109/ACCESS.2019.2960314.

[19] P. B. Charlesworth, “Using non-cooperative games to coordinate communications UAVs,” in *2014 IEEE Globecom Workshops (GC Wkshps)*, IEEE, Dec. 2014, pp. 1463–1468. doi: 10.1109/GLOCOMW.2014.7063640.

[20] A. Fotouhi, M. Ding, and M. Hassan, “DroneCells: Improving spectral efficiency using drone-mounted flying base stations,” *J. Netw. Comput. Appl.*, vol. 174, no. March 2020, p. 102895, Jan. 2021, doi: 10.1016/j.jnca.2020.102895.

[21] M. E. Mkiramweni, C. Yang, J. Li, and W. Zhang, “A Survey of Game Theory in Unmanned Aerial Vehicles Communications,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 4, pp. 3386–3416, 2019, doi: 10.1109/COMST.2019.2919613.

[22] A. Rahmati *et al.*, “Interference Avoidance in UAV-Assisted Networks: Joint 3D Trajectory Design and Power Allocation,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9013532.

[23] H. Zhao, H. Wang, W. Wu, and J. Wei, “Deployment Algorithms for UAV Airborne Networks Toward On-Demand Coverage,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, Sep. 2018, doi: 10.1109/JSAC.2018.2864376.

[24] 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 Netw.*, vol. 30, no. 1, pp. 100–105, Jan. 2016, doi: 10.1109/MNET.2016.7389838.

[25] H. Bayerlein, P. De Kerret, and D. Gesbert, “Trajectory Optimization for Autonomous Flying Base Station via Reinforcement Learning,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, Jun. 2018, pp. 1–5. doi: 10.1109/SPAWC.2018.8445768.

[26] C. Shen, T.-H. Chang, J. Gong, Y. Zeng, and R. Zhang, “Multi-UAV Interference Coordination via Joint Trajectory and Power Control,” *IEEE Trans. Signal Process.*, vol. 68, pp. 843–858, 2020, doi: 10.1109/TSP.2020.2967146.

[27] R. Amer, W. Saad, and N. Marchetti, “Mobility in the Sky: Performance and Mobility Analysis for Cellular-Connected UAVs,” *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3229–3246, May 2020, doi: 10.1109/TCOMM.2020.2973629.

[28] P. K. Sharma and D. I. Kim, “Random 3D Mobile UAV Networks: Mobility Modeling and Coverage Probability,” *IEEE Trans. Wirel. Commun.*, vol. 18, no. 5, pp. 2527–2538, May 2019, doi: 10.1109/TWC.2019.2904564.

[29] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications,” *IEEE Trans. Wirel. Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017, doi: 10.1109/TWC.2017.2751045.

[30] H. Wang, G. Ren, J. Chen, G. Ding, and Y. Yang, “Unmanned Aerial Vehicle-Aided Communications: Joint Transmit Power and Trajectory Optimization,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 4, pp. 522–525, Aug. 2018, doi: 10.1109/LWC.2018.2792435.

[31] G. Wu, X. Gao, and K. Wan, “Mobility Control of Unmanned Aerial Vehicle as Communication Relay to Optimize Ground-to-Air Uplinks,” *Sensors*, vol. 20, no. 8, p. 2332, Apr. 2020, doi: 10.3390/s20082332.

[32] S. Pérez-Carabaza, J. Scherer, B. Rinner, J. A. López-Orozco, and E. Besada-Portas, “UAV trajectory optimization for Minimum Time Search with communication constraints and collision avoidance,” *Eng. Appl. Artif. Intell.*, vol. 85, no. July, pp. 357–371, Oct. 2019, doi: 10.1016/j.engappai.2019.06.002.

[33] S. Huang, R. S. H. Teo, and K. K. Tan, “Collision avoidance of multi unmanned aerial vehicles: A review,” *Annu. Rev. Control*, vol. 48, pp. 147–164, 2019, doi: 10.1016/j.arcontrol.2019.10.001.

[34] A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal LAP Altitude for Maximum Coverage,” *IEEE Wirel. Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014, doi: 10.1109/LWC.2014.2342736.

[35] Y. Chen, W. Feng, and G. Zheng, “Optimum Placement of UAV as Relays,” *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 248–251, Feb. 2018, doi: 10.1109/LCOMM.2017.2776215.

[36] M. Banagar and H. S. Dhillon, “3D Two-Hop Cellular Networks With Wireless Backhauled UAVs: Modeling and Fundamentals,” *IEEE Trans. Wirel. Commun.*, vol. 21, no. 8, pp. 6417–6433, Aug. 2022, doi: 10.1109/TWC.2022.3149213.

[37] R. Jain, D. Chiu, and W. Hawe, “A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems,” Sep. 1984. [Online]. Available: http://arxiv.org/abs/cs/9809099

[38] Y. Zeng and R. Zhang, “Energy-Efficient UAV Communication With Trajectory Optimization,” *IEEE Trans. Wirel. Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017, doi: 10.1109/TWC.2017.2688328.

[39] J. Zhang, J. F. Campbell, D. C. Sweeney II, and A. C. Hupman,

"Energy consumption models for delivery drones: A comparison and assessment," *Transp. Res. Part D Transp. Environ.*, vol. 90, p. 102668, Jan. 2021, doi: 10.1016/j.trd.2020.102668.

[40] T. Kirschstein, "Comparison of energy demands of drone-based and ground-based parcel delivery services," *Transp. Res. Part D Transp. Environ.*, vol. 78, no. December 2019, p. 102209, Jan. 2020, doi: 10.1016/j.trd.2019.102209.

[41] DJI, "DJI Phantom drones," dji.com. Accessed: Mar. 30, 2023. [Online]. Available: https://www.dji.com/global/phantom-4-pro/info#faq

[42] A. Thai, S. Grace, and R. Jain, "CFD Validation of Small Quadrotor Performance using CREATE - AV Helios," in *Proceedings of the Vertical Flight Society 75th Annual Forum*, The Vertical Flight Society, May 2019, pp. 1–15. doi: 10.4050/F-0075-2019-14483.

[43] D. Monderer and L. S. Shapley, "Potential Games," *Games Econ. Behav.*, vol. 14, no. 1, pp. 124–143, May 1996, doi: 10.1006/game.1996.0044.

[44] M. Voorneveld, "Best-response potential games," *Econ. Lett.*, vol. 66, no. 3, pp. 289–295, Mar. 2000, doi: 10.1016/S0165-1765(99)00196-2.

[45] P. Dubey, O. Haimanko, and A. Zapechelnyuk, "Strategic complements and substitutes, and potential games," *Games Econ. Behav.*, vol. 54, no. 1, pp. 77–94, Jan. 2006, doi: 10.1016/j.geb.2004.10.007.

[46] M. Manea and M. Yildiz, "Non-cooperative games," MIT OCW Archived Courses: Economics: 14.126 Game Theory, Spring 2010. [Online]. Available: http://dspace.mit.edu/bitstream/handle/1721.1/106995/14-126-spring-2010/contents/lecture-notes/MIT14_126S10_lec14.pdf

[47] M. Shanmugavel, A. Tsourdos, B. White, and R. Żbikowski, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Eng. Pract.*, vol. 18, no. 9, pp. 1084–1092, Sep. 2010, doi: 10.1016/j.conengprac.2009.02.010.

[48] G. Avanzini, G. de Matteis, and L. M. de Socio, "Analysis of Aircraft Agility on Maximum Performance Maneuvers," *J. Aircr.*, vol. 35, no. 4, pp. 529–535, Jul. 1998, doi: 10.2514/2.2356.

[49] Q. D. Lã, Y. H. Chew, and B.-H. Soong, *Potential Game Theory: Applications in Radio Resource Allocation*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-30869-2.

[50] J. R. Marden and J. S. Shamma, "Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation," *Games Econ. Behav.*, vol. 75, no. 2, pp. 788–808, Jul. 2012, doi: 10.1016/j.geb.2012.03.006.

[51] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous Vehicle-Target Assignment: A Game-Theoretical Formulation," *J. Dyn. Syst. Meas. Control*, vol. 129, no. 5, pp. 584–596, Sep. 2007, doi: 10.1115/1.2766722.

[52] J. R. Marden, G. Arslan, and J. S. Shamma, "Connections between cooperative control and potential games illustrated on the consensus problem," in *2007 European Control Conference (ECC)*, IEEE, Jul. 2007, pp. 4604–4611. doi: 10.23919/ECC.2007.7069001.

[53] M. Hasanbeig and L. Pavel, "From Game-theoretic Multi-agent Log Linear Learning to Reinforcement Learning," no. 2011, pp. 1–34, Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.02277

[54] S. Popov, "Two-Dimensional Random Walk: From Path Counting to Random Interlacements," *Two-Dimensional Random Walk From Path Count. to Random Interlacements*, pp. 1–209, 2021, doi: 10.1017/9781108680134.

[55] 3GPP, "Study on International Mobile Telecommunications (IMT) parameters for 6.425-7.025GHz, 7.025-7.125GHz and 10.0-10.5 GHz (Release 17) TR 38.921 V17.1.0 (2022-03)," 2022.

[56] 3GPP, "Further enhancements to LTE Time Division Duplex (TDD) for Downlink-Uplink (DL-UL) interference management and traffic adaptation (Release 11) TR 36.828 V11.0.0 (2012-06)," 2012.

OMAR ALI THABET (Student Member, IEEE) received the B.Sc. degree in communications engineering from University of Diyala, Diyala, Iraq, in 2013, and the M.Sc. degree in electronics and communications engineering from Yildiz Technical University, Istanbul, Turkey, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Electrical and Electronics Engineering, in telecommunications branch, Yildiz Technical University, Istanbul, Turkey. His main research interests include mobile networks, femtocells, interference mitigation, game theory, UAV communication networks.

DIDEM KIVANC TURELI (Member, IEEE) received the B.S. degree from Boğaziçi University, Istanbul, Turkey, and the M.S. degree from the University of Surrey, U.K. and the Ph.D. degree in electrical engineering from the University of Washington, Seattle. She is an Assistant Professor in the Department of Mechatronics Engineering in Istanbul Okan University, Istanbul. Her research interests are in wireless cellular communication, sensor networks, adaptive control algorithms.

UFUK TURELI (Member, IEEE) received the B.S. degree from Boğaziçi University, Istanbul, Turkey, and the M.S. and Ph.D. degrees in electrical engineering from the University of Virginia, Charlottesville, in 1998 and 2000, respectively. Since June 2015, he has been a Professor and the Director of the Telecommunications Branch, Department of Electronics and Communication Engineering, Yildiz Technical University, Istanbul. He has published numerous journal and conference papers on scalable and robust wireless networks from the physical layer to network layers. He is a member of the IEEE Communications Society. He was on technical program committees (TPC) of the IEEE Globecom, Milcom, ICC, and WCNC conferences. He was also on the IEEE Communications Society (COMSOC) Radio Communication. He served on the Editorial Board of Physical Communication (PHYCOM) Journal (Elsevier) and an Associate Editor. He also served as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.