This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Joint Coverage, Connectivity, and Charging Strategies for Distributed UAV Networks

Angelo Trotta, Marco Di Felice, Federico Montori, Kaushik R. Chowdhury, *Senior Member, IEEE*, and Luciano Bononi

***Abstract*—This paper proposes deployment strategies for consumer unmanned aerial vehicles (UAVs) to maximize the stationary coverage of a target area and to guarantee the continuity of the service through energy replenishment operations at ground charging stations. The three main contributions of our work are as follows. 1) A centralized optimal solution is proposed for the joint problem of UAV positioning for a target coverage ratio and scheduling the charging operations of the UAVs that involves travel to the ground station. 2) A distributed game-theory-based scheduling strategy is proposed using normal-form games with rigorous analysis on performance bounds. Furthermore, a bio-inspired scheme using attractive/repulsive spring actions are used for distributed positioning of the UAVs. 3) The cost-benefit tradeoffs of different levels of cooperation among the UAVs for the distributed charging operations is analyzed. This paper demonstrates that the distributed deployment using only 1-hop messaging achieves approximation of the centrally computed optimum, in terms of coverage and lifetime.**

***Index Terms*—Aerial robotics, battery recharge scheduling, force control, game theory, networked robots, sensor networks.**

I. Introduction

Unmanned aerial vehicles (UAVs) represent one of the fastest growing technological sectors today, with over 126% annual increase in the market size in 2016 and estimated global revenues of around 3 billion dollars [1]. Such UAVs are being envisaged for a variety of use-cases such as disaster recovery [5] and cellular data offloading [6], which require a connected aerial mesh network that provides continuous spatio-temporal coverage of a target area [2]–[4]. The following two key issues must be addressed to facilitate the deployment of the UAV mesh network: first, multiobjective localization of UAVs that considers the sensing coverage needs of the network, and second, maintaining persistent service considering energy-related interruptions. This paper tackles these issues through a rigorously derived analytical framework that provides both centralized and distribution solutions.

A. Challenges in UAV Localization and Persistent Service

The central concern in multiobjective localization is how to set the static locations of each UAV in a 3-D space, so that both the sensing needs of the application and the aerial mesh connectivity requirements are met [7]. While several works have looked into communication-aware mobility schemes for UAVs, few of them jointly address the problem of static coverage [8], [9], [11].

The second issue of ensuring persistent service stems from the limitations of the on-board battery, which is in the order of fraction of an hour (typically 15–20 mins) for most commercial consumer-grade UAVs [12]. Solutions for terrestrial technology such as cross-layer energy-efficient communication, when mapped to UAVs, are unlikely to make a significant impact [15]. This is because the ratio of the cumulative energy cost of all sensing/communication tasks to that of operating the motors is 20:80 [16]. We believe these practical difficulties can be surmounted by carefully scheduling energy replenishment operations and leveraging charging stations on the ground. To realize this paradigm, an intelligent scheduler policy is needed that directs the UAVs to ground-based charging stations depending on the application requirements and the scenario constraints (e.g., number of UAVs that can charge synchronously). Previous research on the UAV scheduling problem has revealed that the problem is NP-complete [2], [3], [18], and with the additional consideration of ensuring coverage, the complexity of the problem only increases.

B. Proposed Research and Contributions

We approach the combined problem of stationary UAV coverage and energy replenishment scheduling by proposing a combined framework, which ensures that user-defined coverage metrics are met and mesh connectivity is maintained, while maximizing the persistent service requirement. We consider a generalized system model, composed of N_S UAVs and one ground-based charging station to make the following three main research contributions.

1) We devise optimal and heuristic solutions to both the problems of UAV positioning and scheduling the recharging cycles considering also the impact of unique issues related to UAVs, such as their height above the ground, the energy

Manuscript received July 12, 2017; revised February 15, 2018; accepted April 23, 2018. This paper was recommended for publication by Guest Editors S.-J. Chung and A. A. Paranjape upon evaluation of the reviewers' comments. This work was supported by the ALMAIDEA UniBO Grant Senior Project: "BeeDrones: Environmental monitoring systems based on ultralow power sensor nodes and aerial communications." *(Corresponding author: Angelo Trotta.)*

A. Trotta, M. Di Felice, F. Montori, and L. Bononi are with the Department of Computer Science and Engineering, University of Bologna 40126, Italy (e-mail: angelo.trotta5@unibo.it; marco.difelice3@unibo.it; federico.montori2@unibo.it; luciano.bononi@unibo.it).

K. R. Chowdhury is with the Northeastern University, Boston, MA 02115 USA (e-mail: krc@ece.neu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2018.2839087

overhead of making the ascent/descent, and the beaconing frequency.

2) We develop a distributed UAV deployment algorithm building on virtual spring mechanisms [9], which relies on the local information (e.g., residual energy), and the information received by other UAVs forming the aerial mesh. We compare this approach with the global optimal found through centralized knowledge.
3) We investigate the cost-benefit tradeoff of information exchange among UAVs for ensuring persistence service that maximizes network lifetime. Then, we formulate three variants of game-theory-based UAV energy cycling schedulers with varying levels of information exchange performed by the UAVs and compute the mixed strategies guaranteeing the Nash equilibrium for them. Importantly, we show that with only local 1-hop knowledge our distributed scheduler performs close to that with global multihop knowledge with a centralized scheduler, which has many practical deployment advantages.

The rest of the paper is structured as follows. We review the state-of-the-art addressing coverage and energy-efficiency issues in Section II. The system model and problem formulation are described in Section III. The optimal solution through global coordination is presented in Section IV. The distributed approaches for charging scheduling and UAV positioning are detailed in Section V. A rigorous performance evaluation is presented in Section VI, and Section VII concludes the paper.

II. Related Works

We review existing works separately for the dual topics of UAV positioning for stationary coverage and UAV lifetime/energy management.

A. UAV Positioning for Stationary Coverage

Compared to classical multirobot coverage problems, a major novelty in centralized coverage schemes is given by the possibility to control the altitude of each UAV. In [21], the authors investigate the optimal 3-D placement of UAVs, so that the number of connected users on the ground is maximized, while the transmitting power of the UAVs is minimized. Similarly, the study in [8] derives fundamental results about 3-D coverage, clarifying the relationship between altitude, beamwidth of the antenna, and coverage probability. The tradeoff between coverage and connectivity is investigated in [22], where the authors show that guaranteeing both these goals can be challenging in highly sparse networks. Considering distributed approaches, [9] and [24] aim to achieve maximum stationary coverage of a target scenario while preserving connectivity among the UAVs. More specifically, [9] proposes a mobility scheme based on the virtual spring model [11], such that all the aerial links experience the same quality of service regardless of the propagation conditions. A channel-aware swarm mobility scheme is proposed in [24], based on the cluster-breathing technique and on the utilization of the receiver signal strength metric as proxy of the link quality. The problem of minimizing the total distance travelled by the mobile robots in order to visit a set of target locations, denoted as target assignment in robotic networks, is address by [10]. For the link communications, the authors used both a simple circular range-based model and a region-based model- in which all robots within the same region can communicate with each other.

B. UAV Lifetime/Energy Management

UAV network lifetime maximization can be achieved through different approaches, i.e., energy-aware control protocols, energy-aware network protocols, and external energy replenishment through ground-based charging stations. Energy-aware control protocols minimize the unnecessary maneuvers of mobile devices [7], or devise an energy-aware path plan meeting constraints on minimal coverage [23]. Energy-aware network protocols for UAVs mitigate the impact of wireless network operations on the battery consumption, with a comprehensive survey available in [15].

For the solutions involving energy replenishment through scheduling of UAVs, we further distinguish between two classes, i.e., 1) path planning oriented and 2) stationary coverage oriented. In 1), the UAVs keep flying over a set of sites, and the goal of the scheduler is to determine the optimal tour of the UAVs, so that each site is visited with a given frequency. Examples are described in [2], [18], [25], [26], [28], and [29]. More specifically, Vasile and Belta [26] formulate the scheduling problem by means of temporal logic, while the scheduler in [18] computes the itinerary of each UAVs, so that the presence of an energy-feasible path toward the replenishment station is always guaranteed. In [2], the paths of UAVs are computed in order to fully cover the trajectory of a mobile user; to this aim, the scheduler assigns each UAV to a space-time segment minimizing the travel distance. The charging slots are allocated via mixed linear integer programming techniques. In [28] and [29], the authors investigate the utilization of unmanned ground vehicles (UGVs) as mobile recharging stations and they develop path planning algorithms for both the UAVs and the UGVs, by considering a modified version of the traveling salesman problem.

Stationary coverage-oriented schedulers assume the presence of UAVs at fixed positions, and aim to guarantee the continuous coverage of a target area. Shakhatreh *et al.* [3] prove that the problem of determining the minimum number of UAVs guaranteeing the persistent coverage of a target area is NP-complete. The study in [13] describes a network architecture composed of fixed battery-powered access points (APs) and of UAVs, which carry full-charged batteries back and forth between the APs and an energy-supplying station. In this case, the static coverage is provided by the wireless ground mesh formed by the APs, while the UAVs perform the energy replenishment operations. Similar to [3], we address stationary coverage. However, the number of UAVs is assumed as an input to the problem in our case, and the goal of the scheduler is to determine the maximum lifetime while guaranteeing a minimum coverage. Hence, the problem becomes computationally tractable and closer to the characteristics of a real scenario, where the available resources (e.g., the number of the UAVs) are typically known in advance.

On a more technology development front, the study in [17] demonstrates a guidance system enabling the UAV to land on a

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
TABLE OF VARIABLES/SYMBOLS

D	Area to cover
N_S	Numbers of UAVs
$A = \{a_1, \ldots, a_{N_s}\}$	Set of the UAVs
$T = \{t_0, t_1, \ldots\}$	Slotted time
t_{slot}	Length of a time slot
S_E	Charging station
h	Flight altitude
θ	Angle of the sensing cone
α_t	Per-slot Loss of energy while flying
β_t	Per-slot Gain of energy while recharging
γ_h	Loss of energy for the descending operation
δ_h	Loss of energy for the ascending operation
$OP(h) = \gamma_h + \delta_h$	Energy overhead for the to-and-fro journey
$\phi : T \to \{0, 1\}$	Availability function for the station S_E
$S = \{s_{\text{fly}}, s_{\text{rec}}\}$	Set of possible UAV states
$E : A, T \to \mathbb{R}$	Residual energy function
$s : A, T \to \{0, 1\}$	Recharge scheduling function
G_{OK}	Action of attempting to access the station S_E
G_{NO}	Action of remaining in flight state
R_{OK}	Action of releasing the station S_E
R_{NO}	Action of not releasing the station S_E
$p_G^{i,j}$	Probability that UAV a_i at time slot t_j executes the action G_{OK}
$p_R^{k,j}$	Probability that UAV a_k at time slot t_j executes the action R_{OK}
$p_B^{i,j}$	Probability that UAV a_i at time slot t_j finds the station S_E busy
$p_{T_0}^{k,j}$	Probability that no UAV, except for a_k, at time slot t_j executes the action G_{OK}

charging station. The problem of a reliable recharging process is addressed in [14], where the authors design a ground recharge station for UAVs, and propose a charging scheduler that assign priorities to UAVs in proportion to their battery level. Swieringa *et al.* [27] describe a battery replacement system for UAVs, evaluating it on a small test bed. In [30], the authors describe the design of an autonomous battery change/recharge station and demonstrates the possibility to support persistent missions of more than 3 h duration with small UAVs.

This paper extends our previous conference paper [4], in which we introduced a preliminary version of the problem and the centralized deployment. Here, we considerably enhance the system model and problem formulation, and also design the distributed deployment. The performance evaluation has also been completely revised to reflect these new contributions.

III. SYSTEM MODEL

We now introduce the system model and the assumptions adopted in the rest of the paper, followed by a mathematical definition of the research problem. Table I list the symbols and variables for ease of reference.

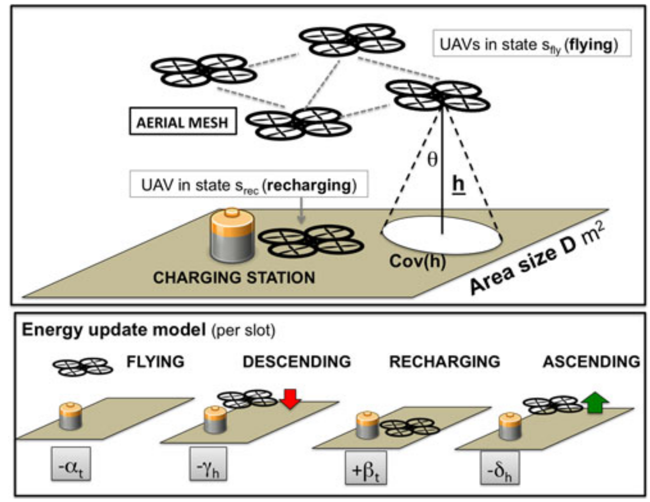

Fig. 1. Aerial mesh network with the charging station on the ground.

A. Scenario Modeling

We consider a square area of size $D\ m^2$ and a set $A = \{a_1, a_2, \ldots, a_{N_s}\}$ of N_s UAVs. Each UAV is able to sense the environment, communicate wirelessly with other peers, and recharge its battery at the replenishment station S_E that is located on the ground at the center of the scenario. We assume that the station can dispense energy at a speed of C_{S_E} [J/s], while each UAV a_i has a maximum battery capacity equal to E^{MAX}. The UAVs form a connected network at h meters from the ground. Depending on h, each UAV is able to sense an area $\text{Cov}(h)$ equal to

$$\text{Cov}(h) = \pi \cdot \left(h \cdot \tan\left(\frac{\theta}{2}\right)\right)^2 \quad (1)$$

where θ is the angle of the sensing cone depicted in Fig. 1.

Without loss of generality, we assume that the time is divided into consecutive time slots $T = \{t_0, t_1, \ldots\}$ of length equal to t_{slot}. We denote with $E(a_i, t_j)$ the residual energy of agent $a_i \in A$ at time slot $t_j \in T$. We assume that all the UAVs start with the same energy amount, equal to E_{init}, i.e., $E(a_i, t_0) = E_{\text{init}} \leq E^{\text{MAX}}, \forall a_i \in A$.

Next, we introduce the function $\phi : T \to \{0, 1\}$ indicating the availability of the replenishment station at a given time slot. More specifically, $\phi(t_j) = 0$ indicates that the station is occupied by one UAV at slot t_j, while $\phi(t_j) = 1$ indicates that the station is currently available. At each slot t_j, each UAV $a_i \in A$ can be in one of the following two states.

1) State s_{fly} (flying): the UAV a_i does not use the station S_E, losing a perslot constant amount of energy while flying (denoted as α_t in the following).
2) State s_{rec} (recharging): the UAV a_i recharges its battery on the ground, gaining a perslot constant amount of energy (denoted as β_t in the following).

Let $S = \{s_{\text{fly}}, s_{\text{rec}}\}$ denote the UAV state set. Based on its state at time slot t_{j-1}, each UAV can execute different actions at slot t_j. More specifically, if UAV a_i is in state s_{fly} at time slot t_{j-1}, then one between the following two actions can be selected.

1) *Go* (G_{OK})—The UAV a_i attempts to access the replenishment station. If the station S_E is free then the UAV changes its state to s_{rec}, otherwise it remains in state s_{fly}.
2) *Stay* (G_{NO})—The UAV a_i remains in state s_{fly}.

Similarly, if UAV a_i is in state s_{rec} at time slot t_{j-1}, then one of the following two actions is selected.

1) *Release* (R_{OK})—The UAV a_i releases the replenishment station and changes its state to s_{fly}.
2) *Keep* (R_{NO})—The UAV a_i remains in state s_{rec} and keeps recharging for another slot.

The action of trying to acquire the replenishment station has a cost of $\gamma_h = \gamma \cdot h$, which models the energy overhead for the descent operations, i.e., from landing to the ground from an initial height of h. Similarly, the action of releasing the station, and of flying back to the aerial mesh, has a cost of $\delta_h = \delta \cdot h$. We assume that the power consumed during horizontal flight and hover is approximately equivalent [31] and modeled by the parameter α_t. Furthermore, since a single replenishment station is assumed, we do not expect to see large distance between the station and the peripheral UAVs, and hence, we assume that values of γ_h and δ_h will depend on the flight altitude h only: the γ and δ parameters will then correspond to the average of the energy power consumption during the descending and ascending operations, regardless of the length of the path that the UAVs need to travel before reaching the target position. In the following, we indicate with $OP(h) = \gamma_h + \delta_h$ the total energy overhead for completing the to-and-fro journey. We define $\alpha_t = \alpha \cdot t_{\mathrm{slot}}$ as the amount of energy loss during an entire time slot while being in state s_{fly} and $\beta_t = \beta \cdot t_{\mathrm{slot}}$ as the amount of energy gained during an entire time slot while being in state s_{rec}. Selection of α, β, γ, and δ parameters depends on the specifications of the hardware in use, and we list later their quantitative representations for our use-case.

B. Problem Formulation

We denote with $s(a_i, t_j) : A, T \rightarrow \{0, 1\}$ the scheduling function that defines the state for each UAV $a_i \in A$ at each time slot $t_j \in T$. More specifically, if $s(a_i, t_j) = 1$, then the UAV a_i is in state s_{fly} at time slot t_j; vice versa, if $s(a_i, t_j) = 0$, then the UAV a_i is in state s_{rec} at time slot t_j.

Let $g(a_i, t_j) : A, T \rightarrow \{0, 1\}$ be the function indicating whether the UAV a_i executes or not the action G_{OK} at the beginning of time slot t_j. Clearly, $g(a_i, t_j) = 1$ requires that $s(a_i, t_{j-1}) = 1$. Let $r(a_i, t_j) : A, T \rightarrow \{0, 1\}$ be the function indicating whether the agent a_i executes or not the action R_{OK} at the beginning of time slot t_j. Again, $r(a_i, t_j) = 1$ requires that $s(a_i, t_{j-1}) = 0$. The $E(a_i, t_j)$ function is updated according to the following:

$$\begin{aligned} E(a_i, t_j) = E(a_i, t_{j-1}) - s(a_i, t_j) \cdot \alpha_t + (1 - s(a_i, t_j)) \cdot \beta_t \\ - g(a_i, t_j) \cdot (\gamma_h + \delta_h \cdot (1 - \phi(t_j))) - r(a_i, t_j) \cdot \delta_h. \end{aligned} \tag{2}$$

Here, $E(a_i, t_{j-1})$ is the residual energy for agent a_i at the previous time slot. The second term refers to the energy consumption for flying; the third term refers to the energy gained while recharging; the last two terms refer to the energy lost when attempting to change the current state.

We are interested in deploying an energy-efficient UAV network, with constraints in terms of the area covered and of the persistence in service. Let κ be a system threshold on the fraction of the area covered by the aerial network. Our constrained coverage and persistence aerial network deployment (CCPANP) problem can be informally defined as: how to determine an optimal charging scheduling function $s(a_i, t_j), \forall a_i \in A, \forall t_j \in T$, so that the following are true. First, the fraction of area covered by the UAV network at slot t_j is always greater than κ. Second, the network lifetime is prolonged as much as possible. Formally, the CCPANP problem is defined as follows.

Definition 1 (CCPANP problem): Let t_{final} be the lifetime of the system defined by the smallest time slot $t_j \in T$ where $\exists a_i \in A$ such that $E(a_i, t_j) = 0$, i.e., the UAV a_i runs out of battery. Given the set of UAVs $A = \{a_1, a_2, \ldots, a_{N_s}\}$ and the factors $\alpha, \beta, \gamma, \delta$, we want to determine the optimal $s(a_i, t_j)$ function $\forall a_i \in A, \forall t_j \in T$ such that the network lifetime t_{final} is maximized and the following constraints are met $\forall a_i \in A, \forall t_j \in T$:

$$\sum_{a_i \in A} (1 - s(a_i, t_j)) = 1 \qquad \forall t_j \in T \tag{3}$$

$$E(a_i, t_j) > 0, E(a_i, t_j) \leq E_{a_i}^{\mathrm{MAX}} \quad \forall a_i \in A, \forall t_j < t_{\mathrm{final}} \tag{4}$$

$$\rho_j = \frac{C(t_j)}{D} \geq \kappa \qquad \forall t_j \in T. \tag{5}$$

Here, the first condition asserts that at each time slot t_j only one UAV can utilize the station S_E. The second condition asserts that no UAV can run out of battery till t_{final}. The last constraint asserts that the ratio ρ_j between the area covered at time t_j by the aerial network, i.e., $C(t_j)$ and the total area to cover, i.e., D is greater than a given system threshold κ. $C(t_j)$ is defined as

$$C(t_j) = \bigcup_{i=1}^{N_s} (\mathrm{Surf}(t_j, a_i, h, \theta) \cdot s(a_i, t_j)) \tag{6}$$

where $\mathrm{Surf}(t_j, a_i, h, \theta)$ is the surface covered at time slot t_j by the UAV a_i flying at altitude h with the sensor angle θ. Clearly, $\mathrm{Surf}(t_j, a_i, h, \theta) = \emptyset$ if $s(a_i, t_j) = 0$, i.e., the UAV a_i is in state s_{rec} at time slot t_j.

IV. Centralized Optimal Approach

In this Section, we provide the optimal solution to the CCPANP research problem defined above. First, we observe from [34] that the optimal coverage is achieved when placing the UAVs in regular hexagon patterns with side length equals $R = h \cdot \tan(\frac{\theta}{2})$, where R is the area coverage radius. The separation between the UAVs is, hence, equal to $R \cdot \sqrt{3}$. Let $N_{\min}$ be the minimum number of UAVs that is required to guarantee at time slot t_j that $\rho_j \geq \kappa$. Using results in [34], we compute $N_{\min}$ as follows:

$$N_{\min} = \left\lceil \frac{\kappa \cdot D}{\left(h \cdot \tan(\frac{\theta}{2})\right)^2 \cdot \frac{3 \cdot \sqrt{3}}{2}} \right\rceil. \tag{7}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

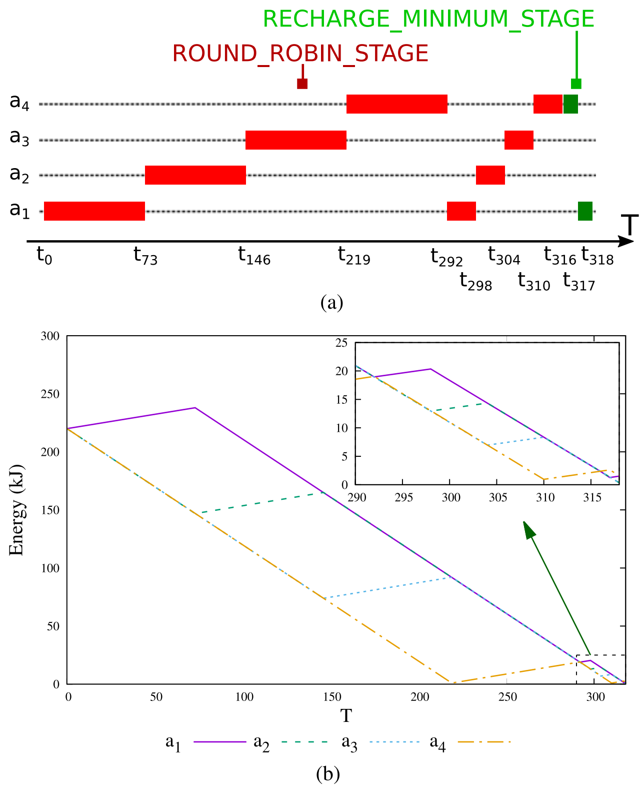

Fig. 2. Two stages of Algorithm 1 with $N_S = 4$, $E_{\text{init}} = 220$ kJ, $OP(h) = 200$ J, $\alpha_t = 1000$ J, $\beta_t = 250$ J. (a) Bars define the time spent by the UAV a_i on the recharging station S_E. The ROUND_ROBIN_STAGE is drawn in red, while the RECHARGE_MINIMUM_STAGE is drawn in green. (b) We show the UAV energy levels during the charging operations. (a) Charging sequence. (b) UAVs energy levels.

Given (7), we assume that $N_S \geq N_{\min} + 1$ in order to be able to cover the requested area of size $\kappa \cdot D$ m^2.

Algorithm 1 shows the pseudocode for solving the CCPANP problem. We assume that all the UAVs have the same initial amount of energy, indicated as E_{init}. At each time-slot, the `schedule` method is executed, and the UAV with id equal to currentCharge is charged. Moreover, we check that all UAVs have energy greater than zero, otherwise the algorithm ends (at line 1). The algorithm can work in two stages, respectively, the ROUND_ROBIN_STAGE, which can be repeated over several iterations, and the RECHARGE_MINIMUM_STAGE. In Fig. 2(a) and (b), we show a graphical example of these two stages.

In the ROUND_ROBIN_STAGE, we let each UAV a_i recharge of the maximum number of sequential slots, denoted as roundSize$[i]$. The exact value of roundSize$[i]$ is computed by the `allocateRoundCharge` method by the following:

1) considering the UAV with maximum residual energy (see line 1);
2) computing the maximum number of slots before such node will drain its energy (numRoundsPerUAV);
3) assigning numRoundsPerUAV to all the UAVs (see lines 1–1);
4) in the case of residuals, allocate the extraRounds slots to the UAVs with minimum energy (see lines 1–1).

Algorithm 1: CCPANP centralized algorithm.

1: **procedure** `schedule`(slotNumber j)
2: Invoke *decideStatus*(j)
3: **for** $i = 1$ to N_s **do**
4: **if** $E(a_i, t_j) == 0$ **then**
5: **return** //End of network lifetime
6: **end if**
7: **if** currentCharge == i **then**
8: $s(a_i, t_j) \leftarrow 0$ //Node a_i is *charging*
9: **if** $s(a_i, t_{j-1}) == 1$ **then**
10: Execute action G_{OK}
11: **else**
12: Execute action R_{NO}
13: **end if**
14: **else**
15: $s(a_i, t_j) \leftarrow 1$ //Node a_i is *flying*
16: **if** $s(a_i, t_{j-1}) == 1$ **then**
17: Execute action G_{NO}
18: **else**
19: Execute action R_{OK}
20: **end if**
21: **end if**
22: **end for**
23:
24: **procedure** `decideStatus`(slotNumber j)
25: **if** status == **ROUND_ROBIN_STAGE** **then**
26: currentRoundCharge $\leftarrow$ currentRoundCharge + 1
27: **if** currentRoundCharge $>$ roundSize[currentCharge] **then**
28: **if** $S_{\text{round}} = \emptyset$ **and not** isRRPhasePossible(j) **then**
29: status $\leftarrow$ RECHARGE_MINIMUM_STAGE
30: **else**
31: **if** $S_{\text{round}} = \emptyset$ **then**
32: Invoke allocateRoundCharge(j)
33: **end if**
34: currentCharge $\leftarrow$ removeFirst(S_{round})
35: currentRoundCharge $\leftarrow$ 1
36: **end if**
37: **else**
38: currentRoundCharge $\leftarrow$ currentRoundCharge +1
39: **end if**
40: **end if**
41: **if** status == **RECHARGE_MINIMUM_STAGE then**
42: currentCharge $\leftarrow$ getMinEnergyNode()
43: **end if**
44:
45: **procedure** `isRRPhasePossible`(slotNumber j)
46: maxNode $\leftarrow$ getMaxEnergyNode()
47: numRounds $\leftarrow \lfloor \frac{E(maxNode, t_j) - OP(h)}{\alpha_t} \rfloor$
48: **if** numRounds $> N_s - 1$ **then**
49: **return True**
50: **else**
51: **return False**
52: **end if**
53:
54: **procedure** `allocateRoundCharge`(slotNumber j)
55: maxNode $\leftarrow$ getMaxEnergyNode()
56: numRounds $\leftarrow \lfloor \frac{E(maxNode, t_j) - OP(h)}{\alpha_t} \rfloor$
57: numRoundsPerUAV $\leftarrow \lfloor \frac{numRounds}{N_s - 1} \rfloor$
58: extraRounds $\leftarrow numRounds \% (N_s - 1)$
59: **for** $i = 1$ to N_s **do**
60: roundSize[i] $\leftarrow$ numRoundsPerUAV
61: **end for**
62: **for** $k = 1$ to extraRounds **do**
63: minNode $\leftarrow$ getMinEnergyNode()
64: roundSize[minNode] $\leftarrow$ roundSize[minNode] + 1
65: **end for**
66: $S_{\text{round}} \leftarrow \{1, ..., N_s\}$
67: Order S_{round} based on roundSize in ascendent order

We then order all the UAVs based on their energy in an ascending order (let S_{round} be this set at lines 1–1), and in turn we extract the first element from S_{round} (see line 1), recharging it of roundSize$[i]$ consecutive slots (see lines 1–1). Once all the UAVs in S_{round} have been charged once, we check whether the ROUND_ROBIN_STAGE can be iterated again through the `isRRPhasePossible` method (see lines 1–1); if so, we compute the new roundSize vector and we use a round robin fashion as explained before. Otherwise, the algorithm enters into the RECHARGE_MINIMUM_STAGE (check at line 1), where at each slot the UAV with the minimum energy is charged (see lines 1–1). It is easy to notice that the complexity of Algorithm 1 is bound by the `allocateRoundCharge` method, which is executed in $O(N_s)$, hence is linear with the number of UAVs. In the following, we provide numerical results about Algorithm 1.

Lemma 1 (Number of iterations): In the ROUND_ROBIN_STAGE, Algorithm 1 performs a number of iterations[1] equal to

$$K = \log_\psi \left\lceil \frac{\alpha_t \cdot (N_s - 1) + \frac{OP(h)\cdot\psi}{1-\psi}}{E_{\text{init}} - OP(h) + \frac{OP(h)}{1-\psi}} \right\rceil \tag{8}$$

where $\psi = \frac{\beta_t}{\alpha_t \cdot (N_s - 1)}$.

Proof: The proof is reported in Appendix A. ■

Theorem 1 (Network Lifetime): The network lifetime T of Algorithm 1 is in range: $\{T_{\text{RR}} \cdot T_{\text{RR}} + N_s - 1\}$. We denote with T_{RR} and T_{MIN} the number of steps executed by Algorithm 1 while being in ROUND_ROBIN_STAGE and RECHARGE_MINIMUM_STAGE, respectively. Clearly, $T = T_{\text{RR}} + T_{\text{MIN}}$ and $0 \leq T_{\text{MIN}} < N_s - 1$.

Proof: The proof is reported in Appendix B. ■

Corollary 1 (Number of swaps): The maximum number of charge swaps, i.e., of the number of changes of the UAV currently under charge, is in range $\{N_s \cdot K \ldots {}_s \cdot K + N_s - 1\}$ for the Algorithm 1 [K is the value given by (8)].

Proof: The proof is reported in Appendix C. ■

We now prove the optimality of the proposed algorithm in terms of network lifetime maximization, using a two-step approach. First, we prove the optimality of Algorithm 1 when $\gamma_h = 0$ and $\delta_h = 0$, i.e., with no penalties for the swap operations. Then, we prove that Algorithm 1 minimizes the number of swap operations with $\gamma_h, \delta_h > 0$.

Theorem 2 (Optimality1): If $\gamma_h = 0$ and $\delta_h = 0$, then Algorithm 1 guarantees the maximum lifetime, i.e., t_{final} is maximum.

Proof: The proof is reported in Appendix D. ■

Theorem 3 (Optimality2): If $\gamma_h > 0, \delta_h > 0,$ Algorithm 1 minimizes the number of charge swaps.

Proof: The proof is reported in Appendix E. ■

Theorem 4 (Optimality3): The Algorithm 1 is able to satisfy the $\rho_j \geq \kappa$ constraint for every $t_j < t_{\text{final}}$.

Proof: The proof is reported in Appendix F. ■

[1] An iteration of the ROUND_ROBIN_STAGE mode is completed when all the UAV nodes have been recharged. Each UAV a_i charges for a number of slots given by roundSize(i).

V. Distributed Game-Theory-Based Approach

Despite its optimality, Algorithm 1 is not conducive for easy implementation since it assumes strict coordination among the UAVs. We now describe an alternate distributed approach including mechanisms for the scheduling of recharging operations and for the distributed positioning of the UAVs. The first component (i.e., charge scheduling) is modeled via game-theory techniques: Section V-A introduces the formulation and the computation of the mixed strategies meeting the Nash equilibrium. Based on it, three different schedulers are proposed in Section V-B with varying levels of knowledge sharing among the UAVs. The second component of positioning the UAVs involves a distributed bio-inspired algorithm, presented in Section V-D.

Although we are aware that the distributed charging scheduling problem can be addressed also via other techniques, such as gossiping mechanisms [32] or distributed network leader elections [33], the choice of the game-theoretical formulation provides the following two main advantages. First, it guarantees convergence to a coordinated solution within a decentralized environment, hence also maximizing reliance in the presence of hardware/software failures of the UAVs. Second, it allows decoupling the strategy played by the UAVs from the information dissemination process, i.e., from the amount of knowledge available at each UAV, as better detailed in the following. Moreover, we remark that in this paper, we are considering the scheduling process as a set of consecutive and different static games at each time slot $t_j \in T$. Our modeling aims to cope with the unpredictable and unknown dynamics of the environment: at each instant, all the UAVs will adapt their behaviors to the actual internal/external conditions, e.g., their residual energy, and take proper decisions. The proposed solutions can also deal with dynamic network scenarios in which the number of UAVs can change over time. Given the requirements to adapt the system response to varying environmental conditions, our formulation does not take into account the relationships among temporal subgames, i.e., it does not track the system temporal evolution. We plan to further elaborate on this issue as future work.

A. Game Formulation and Resolution

Without loss of generality, we model the UAV scheduling operations according to the normal-form game defined as follows.

Definition 2: At each time slot $t_j \in T$, the normal-form game is defined as the triple $\langle A, Z, u^j \rangle$, where

1) $A = \{a_1, a_2, \ldots, a_{N_S}\}$ is the set of UAVs/players;
2) $Z = \{G_{\text{OK}}, G_{\text{NO}}, R_{\text{OK}}, R_{\text{NO}}\}$ is the action set available to each player. The meaning of each action has been described in Section III-A;
3) $u^j = \{u^{1,j}, u^{2,j}, \ldots, u^{N_S,j}\}$ is the profile of the utility functions at time slot t_j, where $u^{i,j}$ is the utility function or payoff function for player a_i, i.e., $u^{i,j} : Z \rightarrow \mathbb{R}$.

Let $\theta^{i,j} = \{p_G^{i,j}, (1 - p_G^{i,j}), p_R^{i,j}, (1 - p_R^{i,j})\}$ be the strategy for player a_i at time slot t_j defining the probability distribution over the set of possible actions Z. Here, $p_G^{i,j}$, $(1 - p_G^{i,j})$, $p_R^{i,j}$,

TABLE II
GENERAL GAME DESCRIPTION

S	Strategy	Z	Utility functions
s_{fly}	$p_G^{i,j}$	$\boldsymbol{G_{\text{OK}}}$	$u^{i,j}(G_{\text{OK}})$
s_{fly}	$(1-p_G^{i,j})$	$\boldsymbol{G_{\text{NO}}}$	$u^{i,j}(G_{\text{NO}})$
s_{rec}	$p_R^{i,j}$	$\boldsymbol{R_{\text{OK}}}$	$u^{i,j}(R_{\text{OK}})$
s_{rec}	$(1-p_R^{i,j})$	$\boldsymbol{R_{\text{NO}}}$	$u^{i,j}(R_{\text{NO}})$

and $(1-p_R^{i,j})$ denote the probabilities to execute actions G_{OK}, G_{NO}, R_{OK}, and R_{NO}, respectively.

We have a *mixed strategy* if more than one action in Z is associated to a nonzero probability; the corresponding actions are called *support* of the mixed strategy. Let $\Theta^{i,j}$ be the set of all possible mixed strategies for player a_i at time slot t_j. Finally, let $\Theta^j = \Theta^{1,j} \times \Theta^{2,j} \times \cdots \times \Theta^{N_S,j}$ be the set of all strategy profiles at time slot t_j.

We define $\Theta^{i,j} \doteq \Theta_G^{i,j}$ as the set of all possible mixed strategies for player a_i being in state s_{fly} (flying). In accordance with the state definition of Section III-A, the support of $\Theta_G^{i,j}$ includes G_{OK} and G_{NO} only. Moreover, we consider the mixed strategy $\theta_G^{i,j} \in \Theta_G^{i,j}$, defined as: $\theta_G^{i,j} = \{p_G^{i,j}, (1-p_G^{i,j}), 0, 0\}$. Similarly, we define $\Theta^{i,j} \doteq \Theta_R^{i,j}$ as the set of all possible mixed strategies for player a_i being in state s_{rec} (recharging). In this case, the support includes R_{OK} and R_{NO} only. Again, we consider the mixed strategy $\theta_R^{i,j} \in \Theta_R^{i,j}$, defined as: $\theta_R^{i,j} = \{0, 0, p_R^{i,j}, (1-p_R^{i,j})\}$. Table II depicts the game description where we also indicate (as first column) the current state of UAV $a_i \in A$ at time slot $t_{j-1} \in T$, in order to be able to execute the action indicated in the third column.

The goal of the analysis reported in the following is to compute the optimal values of $p_G^{i,j}$ and $p_R^{k,j}$, so that the system achieves a Nash-equilibrium. The final results of the analysis are constituted by Theorems 5 and 6, which provide closed formulations of $p_G^{i,j}$ and $p_R^{k,j}$. Since the game depends on the current state of the UAV, we consider the following two cases separately. 1) The so-called *Catch Game* that is played when the UAV a_i is in state s_{fly}. 2) The *Release Game* that is played by the UAV a_k in state s_{rec}. However, we highlight that such division is for ease of disposition only, since the two stages belong to the same game, although observed from two different perspectives (i.e., from the perspective of an UAV that is in a s_{fly} or in s_{rec} state); moreover, as a further proof of this concept, the solution of the two stages are mutually dependent, as better detailed in the following.

1) Catch Game: We played by a_i in state s_{fly}, having the mixed strategy $\theta_G^{i,j} \in \Theta_G^{i,j}$. At each time slot $t_j \in T$, we must determine the mixed strategy $\theta_G^{i,j} = \{p_G^{i,j}, (1-p_G^{i,j}), 0, 0\}$, and hence, the probability $p_G^{i,j}$ for UAV a_i to execute action G_{OK} (i.e., attempt to occupy the replenishment station); with probability $1-p_G^{i,j}$, UAV a_i executes the action G_{NO} (i.e., keeps flying). In game theory, a Nash equilibrium mixed strategy is achieved when the opponents randomize their actions in order to make the player a_i indifferent between the possible actions [35], i.e.,

$$u^{i,j}(G_{\text{OK}}) = u^{i,j}(G_{\text{NO}}). \tag{9}$$

We now define the utility functions $u^{i,j}(G_{\text{OK}})$ and $u^{i,j}(G_{\text{NO}})$. It is easy to notice that the execution of action G_{OK} can lead to the following two situations. First, if $\phi(t_j) = 1$, then the UAV a_i moves to state s_{rec}, occupies the station S_E, and starts recharging its battery ($s(a_i, t_j) = 1$). Second, if $\phi(t_j) = 0$, then the UAV remains in state s_{fly} ($s(a_i, t_j) = 0$). Let $U_F^{i,j}(G_{\text{OK}})$ and $U_B^{i,j}(G_{\text{OK}})$ be the payoffs received by UAV a_i in the two cases mentioned before. Moreover, let $p_B^{i,j}$ be the probability for UAV a_i to find the station S_E occupied by another UAV at time slot t_j. We can then express the utility function $u^{i,j}(G_{\text{OK}})$ as follows:

$$u^{i,j}(G_{\text{OK}}) = p_B^{i,j} \cdot U_B^{i,j}(G_{\text{OK}}) + (1-p_B^{i,j}) \cdot U_F^{i,j}(G_{\text{OK}}). \tag{10}$$

Similarly, we can define the utility function $u^{i,j}(G_{\text{NO}})$ as follows:

$$u^{i,j}(G_{\text{NO}}) = p_B^{i,j} \cdot U_B^{i,j}(G_{\text{NO}}) + (1-p_B^{i,j}) \cdot U_F^{i,j}(G_{\text{NO}}). \tag{11}$$

The functions $U_B^{i,j}(G_{\text{NO}})$ and $U_F^{i,j}(G_{\text{NO}})$ are the payoffs that UAV a_i receives if the station S_E remains busy ($\phi(t_j) = 0$) or free ($\phi(t_j) = 1$) during time slot t_j, respectively. How to define the payoff functions $U_B^{i,j}(G_{\text{OK}})$, $U_B^{i,j}(G_{\text{NO}})$, $U_F^{i,j}(G_{\text{OK}})$, and $U_F^{i,j}(G_{\text{NO}})$ is explained in Section V-B. Substituting (10) and (11) into (9), we can get a closed formulation of $p_B^{i,j}$ as a function of the payoff functions $U_B^{i,j}(\cdot)$ and $U_F^{i,j}(\cdot)$, i.e.,

$$p_B^{i,j} = \frac{U_F^{i,j}(G_{\text{NO}}) - U_F^{i,j}(G_{\text{OK}})}{U_B^{i,j}(G_{\text{OK}}) - U_F^{i,j}(G_{\text{OK}}) + U_F^{i,j}(G_{\text{NO}}) - U_B^{i,j}(G_{\text{NO}})}. \tag{12}$$

Let a_k be the UAV that is using the station S_E at time slot t_{j-1}. With probability $p_R^{k,j}$, UAV a_k can release the station by executing action R_{OK} at the beginning of time slot t_j; conversely, with probability $(1-p_R^{k,j})$, UAV a_k keeps recharging also during time slot t_j by executing the action R_{NO}. Now, the probability $p_B^{i,j}$ can be computed as the opposite of the idle case, that occurs when UAV a_k releases the station and then no one will try to catch it during slot t_j. More formally

$$p_B^{i,j} = 1 - \left(p_R^{k,j} \cdot \prod_{a_h \in A \setminus \{a_i, a_k\}} (1-p_G^{h,j}) \right). \tag{13}$$

We observe that the case where no UAV a_k is recharging at time slot t_{j-1} is a special instance of (13) with $p_R^{k,j} = 1$. Finally, we derive $p_G^{i,j}$ through the following Theorem.

Theorem 5: For the Nash-equilibrium, the probability $p_G^{i,j}$ for UAV a_i to choose action G_{OK} at the beginning of time slot t_j must be defined as follows:

$$p_G^{i,j} = 1 - \sqrt[(N_S-1)]{\frac{\prod_{a_h \in A \setminus \{a_i\}} (1-p_B^{h,j})}{(1-p_B^{i,j})^{N_S-2} \cdot p_R^{k,j}}} \tag{14}$$

where $p_R^{k,j}$ is the probability defined by (22) in case $\phi(t_{j-1}) = 0$, it is equal to 1 otherwise.

Proof: The proof is reported in Appendix G. ■

Remark Since $0 \leq p_B^{i,j} \leq 1$, we can derive from (12) the following constraints on the settings of the payoffs $U_F^{i,j}(G_{\mathrm{NO}})$, $U_F^{i,j}(G_{\mathrm{OK}})$, $U_B^{i,j}(G_{\mathrm{OK}})$, and $U_B^{i,j}(G_{\mathrm{OK}})$:

$$\begin{cases} U_F^{i,j}(G_{\mathrm{NO}}) > U_F^{i,j}(G_{\mathrm{OK}}) \\ U_B^{i,j}(G_{\mathrm{OK}}) > U_B^{i,j}(G_{\mathrm{NO}}) \\ U_B^{i,j}(G_{\mathrm{OK}}) - U_F^{i,j}(G_{\mathrm{OK}}) + U_F^{i,j}(G_{\mathrm{NO}}) - U_B^{i,j}(G_{\mathrm{NO}}) > 0 \end{cases} \tag{15}$$

or

$$\begin{cases} U_F^{i,j}(G_{\mathrm{NO}}) < U_F^{i,j}(G_{\mathrm{OK}}) \\ U_B^{i,j}(G_{\mathrm{OK}}) < U_B^{i,j}(G_{\mathrm{NO}}) \\ U_B^{i,j}(G_{\mathrm{OK}}) - U_F^{i,j}(G_{\mathrm{OK}}) + U_F^{i,j}(G_{\mathrm{NO}}) - U_B^{i,j}(G_{\mathrm{NO}}) < 0. \end{cases} \tag{16}$$

2) Release Game: We now focus on the analysis of the game played by UAV a_k in state s_{rec}, having the mixed strategy $\theta_R^{k,j} \in \Theta_R^{k,j}$. At each time slot $t_j \in T$, we must determine the mixed strategy $\theta_R^{k,j} = \{0, 0, p_R^{k,j}, (1 - p_R^{k,j})\}$, and hence, the probability $p_R^{k,j}$ for the UAV a_k to execute action R_{OK} (i.e., release the replenishment station: $s(a_k, t_j) = 1$); with probability $1 - p_R^{k,j}$, UAV a_k executes the action R_{NO} (i.e., keeps recharging: $s(a_k, t_j) = 0$). As for the previous analysis, a Nash equilibrium mixed strategy is achieved when the opponents make UAV a_k indifferent on its possible choices, i.e.,

$$u^{k,j}(R_{\mathrm{OK}}) = u^{k,j}(R_{\mathrm{NO}}). \tag{17}$$

In order to compute the value of $u^{k,j}(R_{\mathrm{OK}})$, we distinguish the following two situations that might occur during slot t_j after that UAV a_k has executed action R_{OK}, i.e., first, no UAV attempts acquiring the station, and hence, $\phi(t_j) = 0$; or second, at least one of the other UAVs catches the station at time slot t_j, and hence, $\phi(t_j) = 1$. Let $U_{T_0}^{k,j}(R_{\mathrm{OK}})$ and $U_{T_0}^{k,j}(R_{\mathrm{NO}})$ be the payoffs that the UAV a_k receives by executing, respectively, the action R_{OK} or R_{NO}, and no other UAV is trying to grab the station S_E at time slot t_j. Similarly, let $U_{T_+}^{k,j}(R_{\mathrm{OK}})$ and $U_{T_+}^{k,j}(R_{\mathrm{NO}})$ be the payoffs received by UAV a_k after executing, respectively, the action R_{OK} or R_{NO}, and in case some other UAVs will try catching the station S_E at time slot t_j. How to define the payoff functions $U_{T_0}^{k,j}(R_{\mathrm{OK}})$, $U_{T_0}^{k,j}(R_{\mathrm{NO}})$, $U_{T_+}^{k,j}(R_{\mathrm{OK}})$, and $U_{T_+}^{k,j}(R_{\mathrm{NO}})$ is explained in Section V-B. We can derive the value of $u^{k,j}(R_{\mathrm{OK}})$ as follows:

$$u^{k,j}(R_{\mathrm{OK}}) = p_{T_0}^{k,j} \cdot U_{T_0}^{k,j}(R_{\mathrm{OK}}) + (1 - p_{T_0}^{k,j}) \cdot U_{T_+}^{k,j}(R_{\mathrm{OK}}). \tag{18}$$

Similarly, the term $u^{k,j}(R_{\mathrm{NO}})$ can be expressed as follows:

$$u^{k,j}(R_{\mathrm{NO}}) = p_{T_0}^{k,j} \cdot U_{T_0}^{k,j}(R_{\mathrm{NO}}) + (1 - p_{T_0}^{k,j}) \cdot U_{T_+}^{k,j}(R_{\mathrm{NO}}). \tag{19}$$

In both the equations mentioned above, the term $p_{T_0}^{k,j}$ is the probability that none of the other UAVs being in state s_{fly} performs the action G_{OK} at time slot t_j, and it is defined as follows:

$$p_{T_0}^{k,j} = \prod_{a_h \in A \setminus \{a_k\}} (1 - p_G^{h,j}). \tag{20}$$

Substituting (18) and (19) into (17), we get the formulation of $p_{T_0}^{k,j}$ as a function of the payoff functions $U_{T_0}^{k,j}$ and $U_{T_+}^{k,j}$, i.e.,

$$p_{T_0}^{k,j} = \frac{U_{T_+}^{k,j}(R_{\mathrm{NO}}) - U_{T_+}^{k,j}(R_{\mathrm{OK}})}{U_{T_0}^{k,j}(R_{\mathrm{OK}}) - U_{T_+}^{k,j}(R_{\mathrm{OK}}) + U_{T_+}^{k,j}(R_{\mathrm{NO}}) - U_{T_0}^{k,j}(R_{\mathrm{NO}})}. \tag{21}$$

We finally introduce the theorem mentioned above, which provides the closed-form equation of $p_R^{k,j}$.

Theorem 6: For the Nash-equilibrium, the probability $p_R^{k,j}$ for UAV a_k being in state s_{rec} to choose action R_{OK} at the beginning of time slot t_j must be defined as follow:

$$p_R^{k,j} = \sqrt[(N_S - 1)]{\frac{\prod_{a_h \in A \setminus \{a_k\}} (1 - p_B^{h,j})}{(p_{T_0}^{k,j})^{N_S - 2}}} \tag{22}$$

where $p_B^{h,j}$ is defined in (12).

Proof: The proof is reported in Appendix H. ■

Remark: Since $0 \leq p_{T_0}^{k,j} \leq 1$, we can derive the following constraints on $U_{T_0}^{k,j}(R_{\mathrm{OK}})$, $U_{T_0}^{k,j}(R_{\mathrm{NO}})$, $U_{T_+}^{k,j}(R_{\mathrm{OK}})$, and $U_{T_+}^{k,j}(R_{\mathrm{NO}})$:

$$\begin{cases} U_{T_+}^{k,j}(R_{\mathrm{NO}}) > U_{T_+}^{k,j}(R_{\mathrm{OK}}) \\ U_{T_0}^{k,j}(R_{\mathrm{OK}}) > U_{T_0}^{k,j}(R_{\mathrm{NO}}) \\ U_{T_0}^{k,j}(R_{\mathrm{OK}}) - U_{T_+}^{k,j}(R_{\mathrm{OK}}) + U_{T_+}^{k,j}(R_{\mathrm{NO}}) - U_{T_0}^{k,j}(R_{\mathrm{NO}}) > 0 \end{cases} \tag{23}$$

or

$$\begin{cases} U_{T_+}^{k,j}(R_{\mathrm{NO}}) < U_{T_+}^{k,j}(R_{\mathrm{OK}}) \\ U_{T_0}^{k,j}(R_{\mathrm{OK}}) < U_{T_0}^{k,j}(R_{\mathrm{NO}}) \\ U_{T_0}^{k,j}(R_{\mathrm{OK}}) - U_{T_+}^{k,j}(R_{\mathrm{OK}}) + U_{T_+}^{k,j}(R_{\mathrm{NO}}) - U_{T_0}^{k,j}(R_{\mathrm{NO}}) < 0. \end{cases} \tag{24}$$

B. Scheduling Algorithms

In this section, we define the values of the payoffs used by the Catch and Release games previously introduced. More specifically, we consider three different formulations of the payoffs, corresponding to three different scheduling algorithms.

1) *Global knowledge:* We assume that each UAV a_i knows the residual energy $E(a_h, t_j)$ of all the UAVs in the network $a_h \in A$ and the current state $\phi(t_j)$ of the station S_E, for each time slot $t_j < t_{\mathrm{final}}$. The information exchange is enabled via the periodic broadcast of a STRATEGY message, every T_{STRATEGY} seconds and from each UAV. The STRATEGY message includes the value of $E(a_i, t_j)$ and, for the UAV k being in the s_{rec} state, the time instant t_{START}^k when it started recharging. Through these values, each UAV can compute the payoff values and then the probabilities for mixed strategies [i.e., (14) and (22)]. Moreover, each message is

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
PAYOFFS FOR THE CATCH GAME

$U_F^{i,j}(G_{\text{OK}})$		$\beta_t - OP(h) - \tau$
$U_B^{i,j}(G_{\text{OK}})$	Global	$(-\alpha_t - OP(h) - \tau) \cdot (1 + \xi_G^{i,j})$
	Local	$(-\alpha_t - OP(h) - \tau) \cdot (1 + \xi_L^{i,j})$
	Personal	$(-\alpha_t - OP(h) - \tau) \cdot (1 + \xi_P^{i,j})$
$U_F^{i,j}(G_{\text{NO}})$	Global	$(-\alpha_t + \tau) \cdot (1 + \xi_G^{i,j})$
	Local	$(-\alpha_t + \tau) \cdot (1 + \xi_L^{i,j})$
	Personal	$(-\alpha_t + \tau) \cdot (1 + \xi_P^{i,j})$
$U_B^{i,j}(G_{\text{NO}})$		$-\alpha_t + \tau$

TABLE IV
PAYOFFS FOR THE RELEASE GAME

$U_{T_+}^{k,j}(R_{\text{OK}})$		1
$U_{T_0}^{k,j}(R_{\text{OK}})$	Global	$-N_S \cdot \chi_G^{k,j}$
	Local	$-N_S \cdot \chi_L^{k,j}$
	Personal	$-N_S \cdot \chi_P^{k,j}$
$U_{T_+}^{k,j}(R_{\text{NO}})$	Global	$-\chi_G^{k,j}$
	Local	$-\chi_L^{k,j}$
	Personal	$-\chi_P^{k,j}$
$U_{T_0}^{k,j}(R_{\text{NO}})$		0

rebroadcasted hop by hop, in order to reach all the UAVs of the aerial network. Reliable communication is assumed.

2) *Local knowledge:* We assume that each UAV a_i knows the residual energy $E(a_h, t_j)$ of all the UAVs $a_h \in A$ at one-hop distance, for each time slot $t_j < t_{\text{final}}$; however, it does not have information about the utilization of the station S_E. Let $\text{Neigh}_i \subseteq A$ be the set of one-hop neighbors of UAV a_i. As before, the information exchange is enabled via the periodic broadcast of STRATEGY messages every T_{STRATEGY} seconds, but without involving multihop retransmissions. The lack of global scenario knowledge implies that UAV i might not be able to compute $p_G^{i,j}$ and $p_R^{i,j}$ values such as (14) and (22), unless introducing some approximations, which are explained later in this section.

3) *Personal knowledge:* We assume that each UAV a_i knows its residual energy $E(a_i, t_j)$ only with no STRATEGY message exchanges. Again, the approximations needed for the computation of the mixed strategies are explained later in this section.

As before, we separately consider the *Catch Game* and the *Release Game*.

Table III shows, for all the three algorithms defined above, the payoffs that each UAV a_i being in state s_{fly} at time slot t_j will receive depending on the executed action (G_{OK} or G_{NO}) and on the S_E state ($\phi(t_j) = 0$ or $\phi(t_j) = 1$). Here, α_t, β_t, and $OP(h) = (\gamma_h + \delta_h)$ are the system parameters introduced in Section III-A, τ is a constant parameter modeling the coverage penalty/profit, while $\xi_G^{i,j}$, $\xi_L^{i,j}$, and $\xi_P^{i,j}$ represent the energy factor of a_i with respect to the actual knowledge, for the *global*, *local*, and *personal* algorithms, respectively. The $\xi^{i,j}$ factors are computed as follows:

$$\xi_G^{i,j} = \frac{\underset{1 \le h \le N_S}{\text{argmax}}(E(a_h, t_j)) - E(a_i, t_j)}{\underset{1 \le h \le N_S}{\text{argmax}}(E(a_h, t_j)) - \underset{1 \le h \le N_S}{\text{argmin}}(E(a_h, t_j))} \tag{25}$$

$$\xi_L^{i,j} = \frac{\underset{h \in \text{Neigh}_i \cup \{a_i\}}{\text{argmax}} (E(a_h, t_j)) - E(a_i, t_j)}{\underset{h \in \text{Neigh}_i \cup \{a_i\}}{\text{argmax}} (E(a_h, t_j)) - \underset{h \in \text{Neigh}_i \cup \{a_i\}}{\text{argmin}} (E(a_h, t_j))} \tag{26}$$

$$\xi_P^{i,j} = \text{MAX}\left(0, \frac{E_{\text{init}} - E_{a_i}}{E_{\text{init}}}\right). \tag{27}$$

We notice that the values described in Table III satisfy the constraints defined in (16).

The rationale behind the values of Table III is the following. When executing the action G_{OK}, and the station S_E is free (i.e., the case of $U_F^{i,j}(G_{\text{OK}})$), the payoff is the energy recharged (β_t) minus the energy lost for landing and flying back again ($OP(h)$). If, instead, the station S_E is found busy (i.e., the case of $U_B^{i,j}(G_{\text{OK}})$), the payoff is always a penalty, and includes also the energy lost for remaining in a flying state (α_t); moreover, the penalty increases proportionally with $\xi_\cdot^{i,j}$, i.e., based on the amount of residual energy of UAV a_i (for the personal algorithm) or to the energy level of a_i compared to other known players (for the global and local algorithms). Similarly, the action G_{NO} while the station S_E is free (i.e., the case of $U_F^{i,j}(G_{\text{NO}})$) always leads to a penalty, which is proportional to α_t and to the residual energy of the UAV, as discussed before. Finally, the payoff of executing action G_{NO} with the station S_E busy is always equal to the energy lost being in the s_{fly} state, i.e., α_t. It is easy to notice that all the above-mentioned equations contain the τ parameter, which takes into account the impact of the action been performed by UAV a_i on the scenario coverage. When executing the action G_{OK}, we always add a coverage penalty that is equal to $-\tau$, since the UAV a_i is attempting to move on the ground, hence potentially creating a coverage hole. Vice versa, when executing the action G_{NO}, we add a coverage profit that is equal to $+\tau$.

Table IV shows, for all the three algorithms defined above, the payoffs that the UAV a_k being in state s_{rec} at time slot t_j will receive depending on the executed action (R_{OK} or R_{NO}) and on the behavior of the others UAVs. The rationale behind the values is the following. When executing the action R_{OK}, and at least one other UAV attempts to occupy the station S_E (i.e., the case of $U_{T_+}^{k,j}(R_{\text{OK}})$), we give a unit payoff since the station S_E will not remain idle for the time slot t_j. Instead, when executing action R_{OK}, and no other UAV is willing to occupy the station S_E (i.e., the case of $U_{T_+}^{k,j}(R_{\text{NO}})$), we apply a penalty proportional to the number of UAVs N_S and to the estimated optimal recharge time. To this aim, let $\chi_G^{k,j}$, $\chi_L^{k,j}$, and $\chi_P^{k,j}$ be the ratio between the time spent by player a_k into the recharging station S_E at time slot t_j, and the optimal recharge time for the global, local, and personal schedulers, respectively.

We compute such values as follows:

$$\chi_G^{k,j} = \text{MIN}\left\{1, \frac{t_{S_E}^{k,j}}{t_{\text{EST}}^j(A)}\right\} \tag{28}$$

$$\chi_L^{k,j} = \text{MIN}\left\{1, \frac{t_{S_E}^{k,j}}{t_{\text{EST}}^j(\text{Neigh}_k \cup \{a_k\})}\right\} \tag{29}$$

$$\chi_P^{k,j} = \text{MIN}\left\{1, \frac{t_{S_E}^{k,j}}{t_{\text{EST}}^j(\{a_k\})}\right\}. \tag{30}$$

In the equations mentioned above, $t_{S_E}^{i,j}$ is the time spent by UAV a_k at the station S_E till time slot t_j and can be expressed as: $t_{S_E}^{i,j} = t_j - t_{\text{START}}^k$, while $t_{\text{EST}}^j(\Phi)$ is a the estimated optimal recharge time, computed similarly to Algorithm 1 (see line 1), i.e.,

$$t_{\text{EST}}^j(\Phi) = \frac{\max_E^j(\Phi) - OP(h)}{(N_S - 1) \cdot \alpha_t} \tag{31}$$

where $\max_E^j(\Phi)$ is the maximum energy level among the UAVs $a_h \in \Phi$, at time slot t_j.

In the same way, we penalize UAV a_k when executing action R_{NO} and at least one of the other $N_S - 1$ UAVs is attempting to recharge its battery at the same slot t_j (i.e., the case of $U_{T_+}^{k,j}(R_{\text{NO}})$); again, the penalty is proportional to the $\chi^{k,j}$ ratio defined before. Finally, if none of the other UAVs in state s_{fly} attempt to recharge at slot t_j, we give a neutral payoff that is equal to 0. We see that the values described in Table IV satisfy the constraints defined in (23).

Moreover, we remark that the Tables III and IV describe the different strategies that each UAV $a_i \in A$ will adopt at time slot $t_j \in T$ [see (14)–(22) and (12)–(21)].

1) Approximations for the Local and Personal Schedulers: Both (14) and (22) require global exchange of energy values among the UAVs, as well as the knowledge of t_{START}^k for the UAV in state s_{fly}. To address these issues, we relax the formulation of $p_G^{i,j}$ and $p_R^{i,j}$ for the Local and Personal schedulers.

1) *Local knowledge:* In this case, the UAV a_i gathers only the energy information from the UAV $a_h \in \text{Neigh}_i$, i.e., in its 1-hop neighborhood. Hence, we approximate the value of $p_G^{i,j}$ and $p_R^{i,j}$ as follows:

$$p_G^{i,j} = 1 - \sqrt[(N_S-1)]{\frac{\left(\prod_{a_h \in \text{Neigh}_i}(1 - p_B^{h,j})\right)}{(1 - p_B^{i,j})^{N_S-2} \cdot \overline{p}_R^{i,j}}} \cdot \sqrt[(N_S-1)]{(1 - \overline{p}_B^{i,j})^{(N_S-1-|\text{Neigh}_i|)}} \tag{32}$$

$$p_R^{i,j} = \sqrt[(N_S-1)]{\frac{\left(\prod_{a_h \in \text{Neigh}_i}(1 - p_B^{h,j})\right)}{(p_{T_0}^{i,j})^{N_S-2}}} \cdot \sqrt[(N_S-1)]{(1 - \overline{p}_B^{i,j})^{(N_S-1-|\text{Neigh}_i|)}} \tag{33}$$

where $\overline{p}_B^{i,j}$ is the average of the $p_B^{h,j}$ with $a_h \in \text{Neigh}_i$ and $\overline{p}_R^{i,j}$ is the estimation of $p_R^{k,j}$ for a potential UAV a_k being in state s_{rec}. The value of $\overline{p}_R^{i,j}$ is calculated in (33) by approximating the value of $t_{S_E}^{k,j}$ in (29), i.e., the charging time duration for UAV k that is currently using the station S_E, with the duration of the last charging operation performed by the current UAV a_i.

2) *Personal knowledge:* In this case each UAV a_i knows only its own residual energy $E(a_i, t_j)$, for $j \geq 0$. For this reason, we greatly simplify the formulation of $p_G^{i,j}$ and $p_R^{i,j}$ by assuming constant values of $p_B^{h,j} = p_B^{i,j}$, for $1 \leq h \leq N_S$. Hence, (14) and (22) become

$$p_G^{i,j} = 1 - \sqrt[(N_S-1)]{\frac{(1 - p_B^{i,j})}{\overline{p}_R^{i,j}}} \tag{34}$$

$$p_R^{i,j} = \frac{1 - p_B^{i,j}}{\sqrt[(N_S-1)]{(p_{T_0}^{i,j})^{N_S-2}}} \tag{35}$$

where $\overline{p}_R^{i,j}$ is defined as before for the local knowledge case.

C. Complexity Analysis

We investigate the complexity of proposed solutions by considering both the computational complexity and the information dissemination process overhead. It is easy to notice that the computational complexity is dominated by the calculus at each time slot t_j, for each UAV a_i, of the probability $p_G^{i,j}$ (if a_i is in state s_{fly}), and of the probability $p_R^{i,j}$ (if a_i is in state s_{rec}). Again, we treat separately the three information dissemination schemes, i.e., the global, local, and personal cases.

1) *Global:* We can notice that both $p_R^{i,j}$ and $p_G^{i,j}$ are performed in $O(N)$, since they are both characterized by the products of a sequence of N terms, i.e., the $p_B^{*,j}$. Even if these terms depend on the computation of the min/max variables among the N UAVs [see (12) and (25)], we can assume that these values are precomputed before evaluating $p_R^{i,j}$ and $p_G^{i,j}$.
2) *Local:* This case is similar to the global one, but the calculus is limited to the 1-hop neighborhood. Here, we can consider the cardinality of $|\text{Neigh}_i|, \forall a_i \in A$, as constant (see the following Section V-D). Hence, we can state that the computational complexity is $O(1)$.
3) *Personal:* In this case it is easy to see that both $p_R^{i,j}$ and $p_G^{i,j}$ are $O(1)$.

To analyze the information dissemination procedure, we need to examine the number of STRATEGY messages that are sent inside the UAV network. Again, we treat separately the following three information dissemination schemes.

1) *Global:* In order to implement networkwide energy information dissemination, each UAV $a_i \in A$ must retransmit the STRATEGY message to any other $a_j \in A$. Hence, the number of transmitted message is N^2.
2) *Local:* In this case there is no retransmission of messages; hence, the number of transmitted message is N.
3) *Personal:* Here, no message is transmitted at all, so the number of transmitted message is 0.

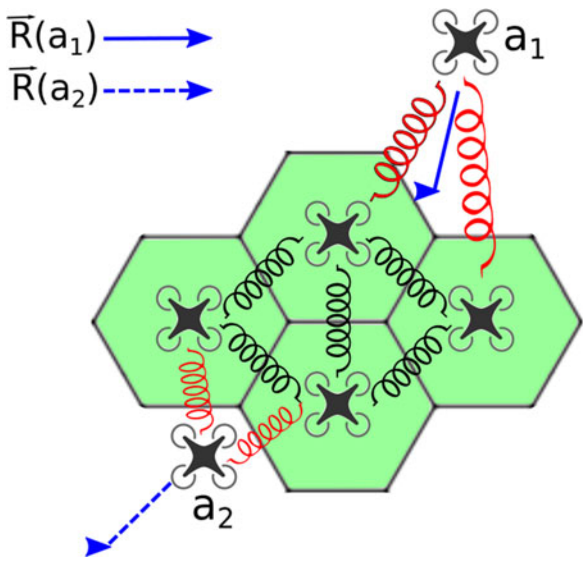

Fig. 3. Aerial mesh positioning algorithm: the virtual spring method places the UAVs according to an hexagonal pattern. The UAVs with a spring length greater than the spring equilibrium are subject to an attractive force (UAV a_1 in the figure), while the UAVs with spring length lower then the spring equilibrium are subject to a repulsive force (UAV a_2 in the figure).

D. UAV Positioning

We assume that each UAV is equipped with GPS and Wi-Fi modules, so that it can know its position and communicate with other peers using the ad-hoc mode. Every $T_{\mathrm{BEACON}} \leq t_{\mathrm{slot}}$ intervals, each UAV i broadcasts a BEACON message containing its identifier and its position ($\vec{x_i}$). The UAV positioning algorithm extends the virtual spring model described in [9] and [11]. A virtual spring force $\vec{F}(a_i, a_h)$ acts between each couple of UAVs (a_i, a_h) that are located at 1-hop distance, i.e., that are able to exchange the BEACON messages. The intensity of $\vec{F}(a_i, a_h)$ is computed by UAVs a_i and a_h according to the Hooke's law

$$|\vec{F}(a_i, a_h)| = -(|\vec{x_i} - \vec{x_h}| - d_{\mathrm{EQ}}) \cdot k_{\mathrm{ST}}. \tag{36}$$

Here, the first term is the spring *displacement*, given by the difference between the current distance from UAV a_i to UAV a_h and the length in equilibrium of the spring, indicated by d_{EQ}. In our case, d_{EQ} is equal to $R \cdot \sqrt{3}$ [here, R is the radius of $\mathrm{Cov}(h)$ in (1)], which is the distance among the UAVs when they are positioned according to hexagonal patterns for the optimal scenario coverage [34]. The force is attractive when the distance between the UAVs is greater than d_{EQ}, repulsive otherwise. The term k_{ST} is the stiffness of the spring and is assumed to be a constant value. Every T_{BEACON} seconds, each UAV a_i gathers the BEACON messages from its 1-hop neighbors (Neigh_i). Then, it determines $\vec{F}(a_i, a_h)$ for each neighbor $h \in \mathrm{Neigh}_i$, and it computes the resultant force $\vec{R}(a_i) = \sum_{h \in \mathrm{Neigh}_i} \vec{F}(a_i, a_h)$. If the module of $\vec{R}(a_i)$ is greater than a threshold value that is analogous to inertia of a mechanical system, then UAVa_i moves toward the direction of the resultant force in a fixed step (see Fig. 3). In this way, the proposed method balances the "push" and "pull" forces' and avoids oscillations in the ensuing movements.

VI. Performance Evaluation

In this section, we evaluate the proposed CCPANP problem solutions via a simulation study in a 3-D network scenario in OMNeT++ [36]. We design and implement a comprehensive set of simulation models of UAV mobility, battery usage, and wireless communication protocols. We compared eight different algorithms, corresponding to the following four main approaches.

1) A *centralized optimal solution* based on Algorithm 1 (denoted as AlgoIV in the following), assuming a global coordination and complete scenario knowledge. This algorithm provides an upper bound to the system performance.
2) A *no-recharge solution,* where the recharge station S_E is not present on the ground, and hence the UAVs must stay in the s_{fly} state all the time. We indicate with Norec such solution, which provides a lower bound to the system performance;
3) The *three game-theory-based distributed algorithms* described in Section V-B. More in details, we consider the following three variants based on the amount of information exchanged by the UAVs.
    a) A global cooperation algorithm (Game_G) implementing the global knowledge game of Section V-B.
    b) A local cooperation algorithm (Game_L) implementing the local knowledge game of Section V-B.
    c) A personal cooperation algorithm (Game_L) implementing the personal knowledge game of Section V-B.
    All the variants use the virtual spring algorithm of Section V-D for the distributed UAV positioning.
4) Three *distributed probabilistic schemes* that let each UAV i recharge at slot j with probability $\mathrm{PR}_{i,j}$. Again, we consider the following three variants of the $\mathrm{PR}_{i,j}$ function based on the amount of information exchanged by the UAVs.
    a) A global cooperation probabilistic algorithm (Prob_G): the $\mathrm{PR}_{i,j}$ value is computed by UAV i at slot j by comparing its actual energy level $E(a_i, t_j)$ with those of the most charged and discharged UAVs in the network

$$\mathrm{PR}_{i,j} = \frac{\underset{1 \leq h \leq N_S}{\mathrm{argmax}}\, E(a_h, t_j) - E(a_i, t_j)}{\underset{1 \leq h \leq N_S}{\mathrm{argmax}}\, E(a_h, t_j) - \underset{1 \leq h \leq N_S}{\mathrm{argmin}}\, E(a_h, t_j)}. \tag{37}$$

    b) A local knowledge probabilistic algorithm (Prob_L): The $\mathrm{PR}_{i,j}$ value is computed by UAV i at slot j by comparing its actual energy level $E(a_i, t_j)$ with those of the most charged and discharged UAVs in its 1-hop neighborhood and is defined as follows:

$$\mathrm{PR}_{i,j} = \frac{\operatorname*{argmax}_{h \in \mathrm{Neigh}_i^+} E(a_h, t_j) - E(a_i, t_j)}{\operatorname*{argmax}_{h \in \mathrm{Neigh}_i^+} E(a_h, t_j) - \operatorname*{argmin}_{h \in Neigh_i^+} E(a_h, t_j)} \tag{38}$$

where $\mathrm{Neigh}_i^+ = \mathrm{Neigh}_i \cup \{a_i\}$.

c) A personal knowledge probabilistic algorithm (Prob_P): The $\mathrm{PR}_{i,j}$ value is computed by UAV i at slot j by comparing its actual energy level $E(a_i, t_j)$ with the initial battery capacity E_{init}, i.e., $\mathrm{PR}_{i,j} = \frac{E(a_i,t_j)}{E_{\mathrm{init}}}$.

For all the three probabilistic schemes described above, we consider a fixed charging time duration that is equal to $\mathrm{Prob}_{\mathrm{rec}}$ time slots. The UAV positioning is handled by the virtual spring algorithm.

The performance evaluation focuses on the following three quality indexes.

1) *Lifetime* (L_{final}*:)* This index is a measure of the system lifetime, computed as the t_{final} value introduced in Section III-A, i.e., the time slot in which the first UAV runs out of battery. We consider also an alternative metric for lifetime, which also accommodates the coverage and service persistence constraints of the CCPANP problem.
2) *CCPANP lifetime* (L_{CCPANP}*:)* This index is an alternate measure of the system lifetime, measured as smallest time slot $t_j \in T$ such that at least one of the constrains of the *CCPANP* problem in the Definition 1 is not satisfied, i.e., first, one UAV runs out of battery or second, the current mesh coverage at slot t_j, i.e., $C(t_j)$, becomes lower than the κ threshold for a given number of consecutive seconds. We introduce the parameter Δt_κ that defines the maximum time interval in which the κ constraint can be violated, thus allowing small interruptions of the coverage service.
3) *Failed attempt ratio* (F_{ratio}*:)* This index defines the ratio between the failed recharge attempts and the total recharge attempts performed by the UAVs.

Unless stated otherwise, we used the following setting of the system parameters: $N_S = 8$, $T_{\mathrm{BEACON}} = 1\,\mathrm{s}$, $T_{\mathrm{STRATEGY}} = 1\,\mathrm{s}$, $\theta = \frac{2}{3} \cdot \pi$, $\kappa = 0.75$, $\Delta t_\kappa = 10$ s, $h = 20$ m, $k_{\mathrm{ST}} = 1$, $\mathrm{Prob}_{\mathrm{rec}} = 1$, $\tau = 150$, $E_{\mathrm{init}} = 130$ kJ, $\alpha = 100$ W, $\beta = 25$ W, $\gamma = 5$ J, $\delta = 5$ J (we modeled an UAV equipped with a generic 3-cell ($3S$) LiPo 11.1 V battery with 3250mAh with an approximated flight autonomy of 20 min and a full recharge time of 80 min).

We split the performance analysis in three parts. Section VI-A investigates the relationship between system performance and scenario deployment characteristics, such as the number of UAVs and the altitude from the ground. Section VI-B shows the impact of system parameters related to the recharge/discharge operations. Finally, Section VI-C investigates how cooperation parameters, such as the frequency of the STRATEGY messages exchanged among the UAVs, affect the system performance.

A. Scenario Analysis

In this section, we analyze the performance of the algorithms by varying characteristics such as the number of available UAVs (N_S) and the flight altitude (h). Fig. 4(a) shows the L_{final} metric on the y-axis, when varying the N_S value on the x-axis. We adopted the following order in this figure and in the following histograms: the first bar is the Algo1 method; the second bar represents the Norec scheme; the next three bars depict the *distributed game-theory approaches* considering the three variants based on the amount of cooperation among the UAVs, i.e., the Game_G, Game_L, and Game_P schemes; finally, the last three bars correspond to the *probabilistic approaches* considering again the three variants according to the amount of cooperation, i.e.,: Prob_G, Prob_L, and Prob_P. We notice that: first, the *Norec* scheme performs worst, as expected; second, while the global cooperation schemes (Game_G and Prob_G) always outperform the personal solutions (i.e., Game_P and Prob_P), they do not provide significant gains over local cooperation schemes (Game_L and Prob_L); third, the game-theory-based schemes perform worse than the probabilistic ones with few UAVs (i.e., $N_S \leq 4$), while the trend reverses for $N_S \geq 6$. The results in Fig. 4(a) only take into account the energy issue, but do not consider the coverage constraint κ. Fig. 4(b) depicts the L_{CCPANP} metric. From $N_S > 4$, all the game-theory-based schemes perform better then the probabilistic ones. Also, it is interesting to notice that the Game_L scheme provides almost the same performance than Game_G, and pretty close to Algo1, i.e., to the optimal upper bound. In other words, the distributed mobility and charging scheduler solution provides a good approximation of the optimal one, but without requiring a global controller, and introducing a much lower network overhead than Game_G. On the opposite, the probabilistic schemes do not cope well with the coverage requirements (i.e., κ); for $N_S \geq 6$, they perform even worse than the Norec method, basically nullifying the gain of the recharging operations. This behavior can be explained by considering the F_{ratio} metric in Fig. 4(c). Probabilistic schemes result in greater number of recharge attempts than the game-theory-based schemes, and most of them fail because the charging station S_E is found busy. The failures also increase with the number of UAVs. When the UAVs move from the air to the ground, a coverage hole may occur, driving the full coverage metric below the κ threshold. This results in the poor performance of the probabilistic schemes in terms of L_{CCPANP} metric. The game-theoretical algorithms leverage the computation of the mixed strategies, and hence, optimize the number of recharge attempts to improve the F_{ratio} quality index. This trend is confirmed in Fig. 5, showing the scenario coverage ratio ρ_j [see (5)] over simulation time, for $N_S = 8$. The spikes in the graph correspond to coverage holes, caused by single or multiple recharge attempts, and by the consequent repositioning of the UAVs according to the virtual spring mobility model. The line interruptions correspond to the L_{CCPANP} lifetime values in Fig. 4(b), i.e., the time slot t_j after which the energy or the coverage (κ) constraints are no more satisfied. We notice that on average, the game-theory-based approaches remain above the value of $\rho > 0.9$. Conversely, the probabilistic methods present

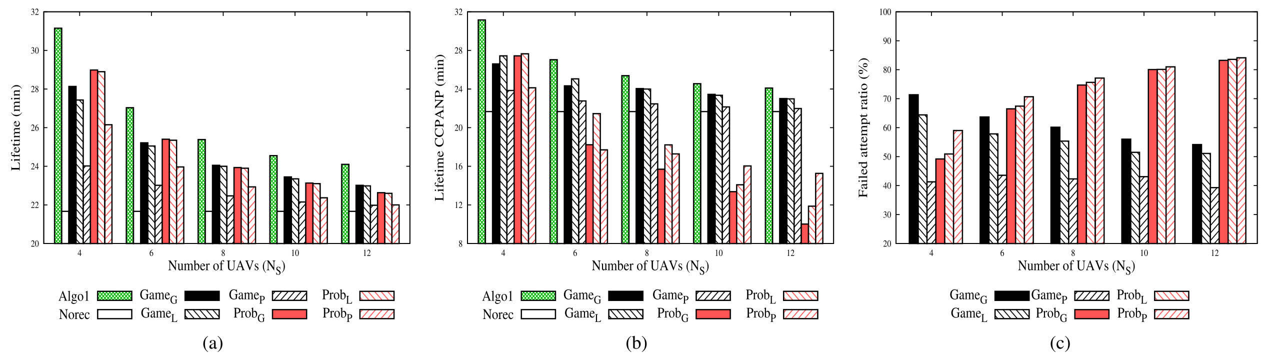

Fig. 4. Performance indexes as a function of N_S. (a) Lifetime. (b) CCPANP lifetime. (c) Failed attempt ratio.

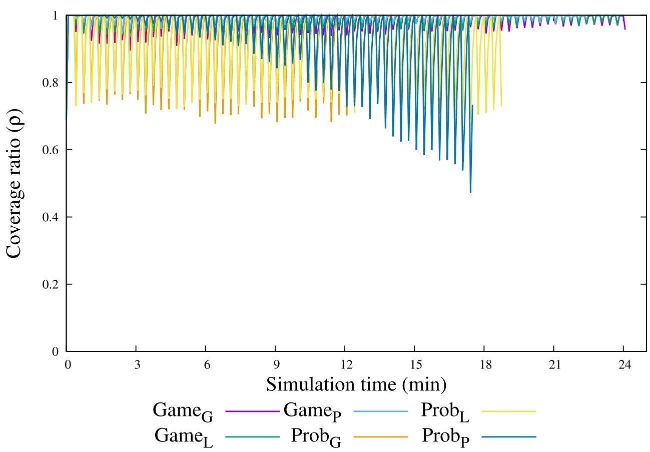

Fig. 5. Value of ρ_j over the simulation time.

large decrease in coverage due the number of recharge attempts at each slot. Also, the spikes increase in the frequency when the average UAV battery power level decreases. Such spikes tend to violate the κ constraint for more than consecutive Δt_κ seconds. Hence, the Game$_L$ scheme provides a performance gain of around +30% L_{CCPANP} lifetime compared to the Prob$_L$ scheme.

The following analysis focuses on the impact of the flight altitude (i.e., h) on the system performance. Fig. 6(a) depicts the L_{final} index when varying the flight altitude from $h = 5$ m to $h = 40$ m. We recall that the variable h impacts on the energy cost of ascending/descending operations (the γ_h and δ_h parameters of Section III) and on the UAV coverage radius ($R = h \cdot \tan(\frac{\theta}{2})$). The trend of the L_{final} index is similar to the previous analysis shown in Fig. 4(a), i.e., the probabilistic approaches initially outperform the game theoretical methods, but provide much lower lifetime when $h \geq 20$ m, i.e., when the energy costs of ascending/descending operations become significative. Fig. 6(b) shows the L_{CCPANP} metric over h. In all the distributed schemes, the UAVs need to fly over a longer distance after each recharge attempt, as the separation distance between the UAVs is a function of the sensing radius R, and thus of h. Hence, the probability of not meeting the minimum coverage ratio κ increases with the flight altitude. However, the game-theory-based approaches still outperform the probabilistic ones, and the gain becomes more relevant when increasing the altitude. This is confirmed by Fig. 6(c) showing the F_{ratio} metric over h, and demonstrating that the probability schemes (Prob$_G$, Prob$_L$, and Prob$_P$) perform an excessive number of recharge attempts. Vice versa, the Game$_G$, Game$_L$, and Game$_P$ schemes are able to cope with the increasing altitude owing to better scheduling of the recharge operations.

B. Parameters Analysis

In this section, we explore the impact of the UAV parameters that directly characterize the discharge/recharge operations, i.e., α, β, γ, and δ on the system performance. In the following, for ease of disposition, we always set the parameters γ and δ to the same values, i.e., $\gamma = \delta$. In Fig. 7(a)–(c), we depict the L_{final}, L_{CCPANP}, and F_{ratio} indexes with $h = 20$ m and $N_S = 8$. The L_{CCPANP} metric of the distributed game-theory-based algorithms are barely affected by the above-mentioned parameters; system performances decrease slowly even with an high value of $\gamma = \delta = 10$ J. Vice versa, for any configuration of γ and δ, the probabilistic methods always perform worse than the basic method Norec algorithm, due to the impact of charging attempts, as also confirmed by Fig. 7(c). In Fig. 8, we show the L_{CCPANP} index when varying the κ threshold (on the x-axis) and the γ-δ parameter (on the y-axis). We only compare the Game$_L$ and Prob$_L$ algorithms, i.e., the game-theory-based and probabilistic schemes both exploiting local cooperation among the UAVs. The Game$_L$ algorithm keeps the L_{CCPANP} index very close to the optimal method, i.e., Algo1, dropping its performance only when requesting a continuous total coverage (i.e., κ equal to 1). The Prob$_L$ method, instead, starts reducing its performance from $\kappa = 0.4$, and achieves much lower L_{CCPANP} values than the Game$_L$ scheme for $\kappa \geq 0.8$. In Fig. 9, we show the L_{CCPANP} index when varying the α and β parameters, while keeping constant the values of $\gamma = \delta = 5J$. Again, we compared the Game$_L$ and Prob$_L$ algorithms. In both cases, the optimal value is achieved when $\beta \gg \alpha$, as expected. However, the Game$_L$ scheme exploits much more efficiently the presence of the recharging station than the Prob$_L$ algorithm, for both different recharge powers (β) and flight discharge characteristics (α).

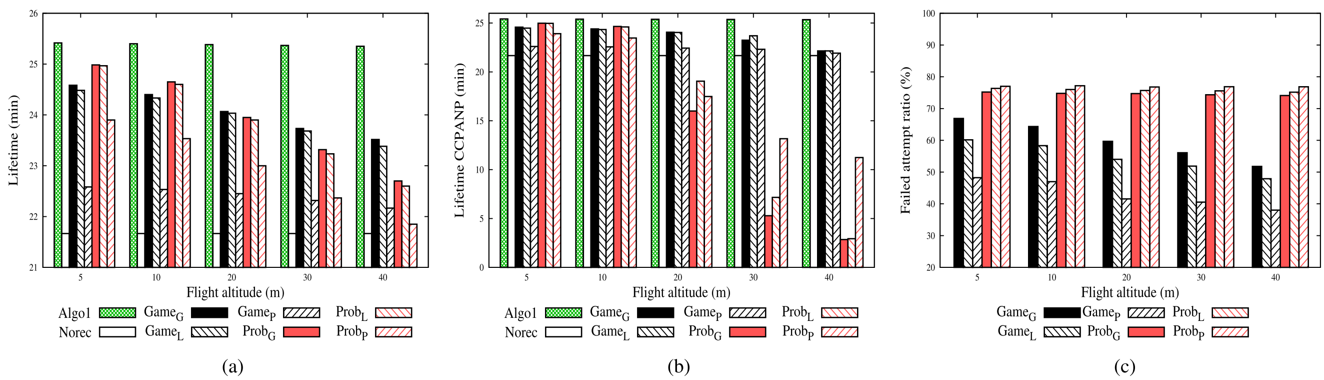

Fig. 6. Performance indexes as a function of the flight altitude h. (a) Lifetime. (b) CCPANP lifetime. (c) Failed attempt ratio.

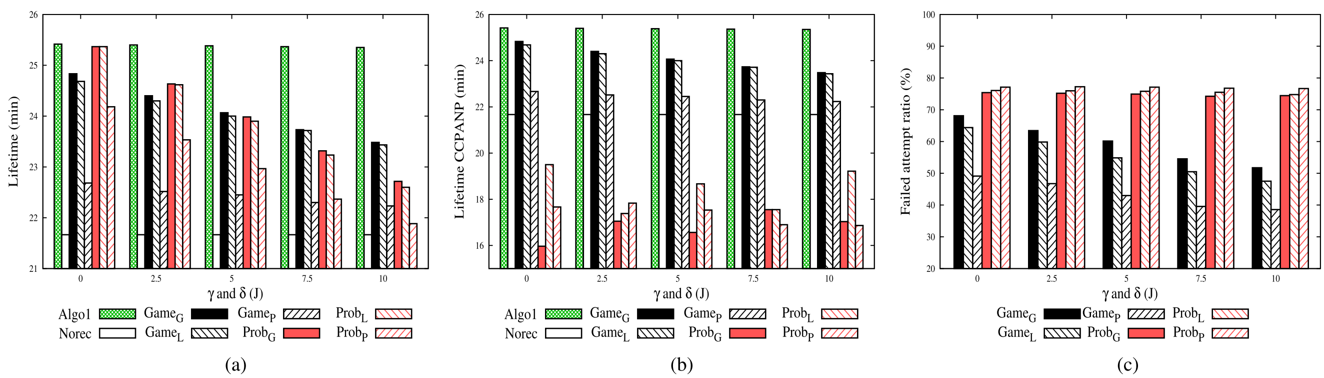

Fig. 7. Performance indexes as a function of γ and δ (in the experiments $\gamma = \delta$). (a) Lifetime. (b) CCPANP lifetime. (c) Failed attempt ratio

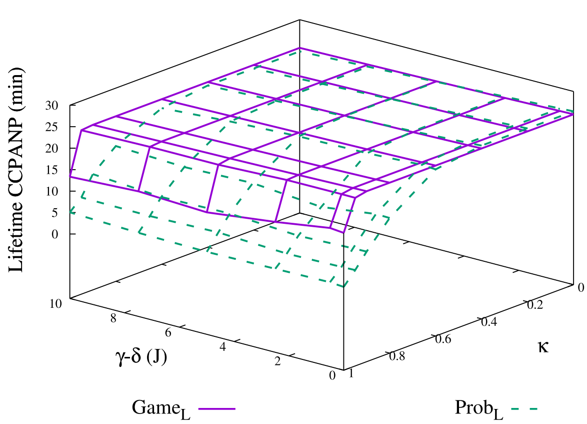

Fig. 8. CCPANP lifetime index varying γ-δ and κ for the methods Game_L and Prob_L.

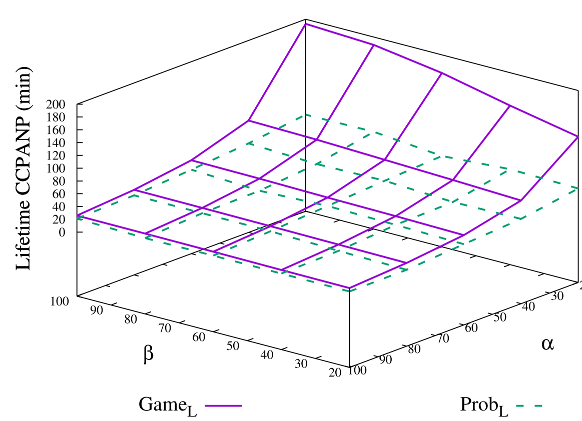

Fig. 9. CCPANP lifetime index varying α and β for the methods Game_L and Prob_L.

C. Cooperation Analysis

Finally, we analyze the impact of UAV cooperation rate on the system performance, by considering the interval T_{STRATEGY} in exchanging the STRATEGY broadcast messages. We recall from Section V-B that the STRATEGY messages contain the $E(a_i, t_j)$ values, needed to compute the mixed strategies in the game-theory-distributed schemes as well as the charging probabilities in the distributed probabilistic schemes. The no-cooperation schemes, i.e., Game_P and Prob_P, are clearly not affected by the analysis and, hence, perform in the same way for all values of T_{STRATEGY}. In Fig. 10(a)–(c), we depict the performance for the L_{final}, L_{CCPANP}, and F_{ratio} indexes, over increasing values of T_{STRATEGY}. We can notice that the game-theory-based approaches keep good performance and are only slightly affected by the freshness of information coming from the other UAVs. This is a quite relevant result, since it showsw that the network communication overhead can be reduced without impacting the L_{CCPANP} performance, although the amount

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

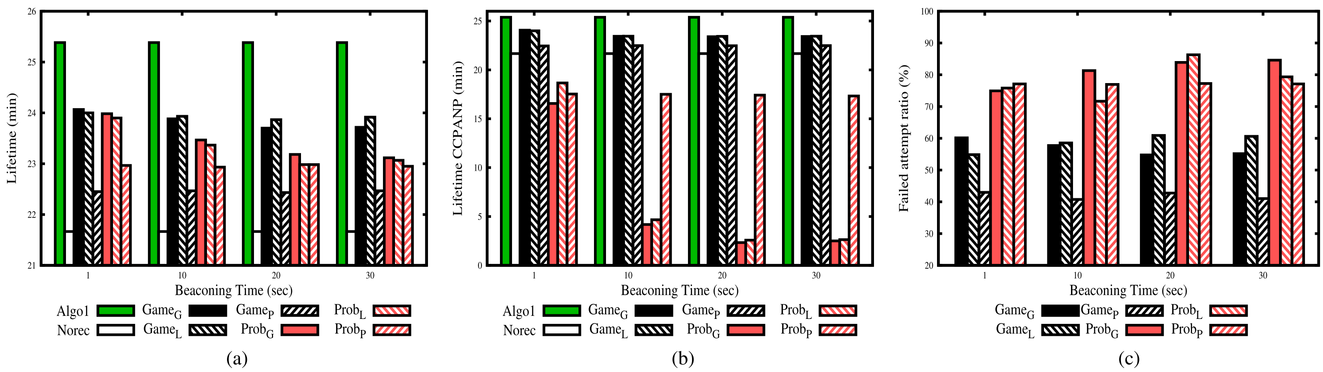

Fig. 10. Performance indexes as a function of the broadcast frequency of the STRATEGY message. (a) Lifetime. (b) CCPANP lifetime. (c) Failed attempt ratio.

of energy drained by the communication module can represent only a small fraction of the energy drained by the rotors [15], [16]. Vice versa, the performances of probabilistic approaches (Prob$_G$ and Prob$_L$) decrease quite significantly when increasing the T_{STRATEGY} interval. Hence, they need much higher communication overhead than the game-theory-based algorithms.

VII. Conclusion

In this paper, we have investigated the deployment of aerial mesh networks meeting requirements of scenario coverage and service continuity through scheduling and UAVs for ground-based recharging. We have developed the optimal solution in the presence of a centralized controller as well as distributed deployment strategies based on game theory techniques and swarm mobility algorithms; different variants of the charging scheduling policies have been proposed according to the extent of cooperation among the UAVs. Our simulation results show that the distributed game-theory-based solutions based on 1-hop neighbor messaging outperforms probabilistic approaches and performs close to the optimal solution, but without the overheads of central coordination and global cooperation. Our future work will extend the theoretical framework to multiple ground-charging stations, consider mobility of these stations, and investigate different altitude values for different UAVs.

Appendix A
Proof of Lemma 1

We observe that the Algorithm 1 keeps iterating the ROUND_ROBIN_STAGE till the following condition becomes true (see line 40):

$$E(*, k) < \alpha_t \cdot (N_s - 1) + OP(h) \tag{39}$$

where $E(*, k)$ is the energy of maxNode (i.e., of the UAV with maximum residual energy) after having completed k iterations in the ROUND_ROBIN_STAGE mode. Moreover, at each iteration k, the following recursive property holds:

$$E(*, k) = E(*, k-1) - x_k \cdot (\alpha_t \cdot (N_s - 1) - \beta_t) - OP(h) \tag{40}$$

where x_k is the number of charging slots assigned to each UAV (i.e., the numRoundsPerUAV at line 49, assuming extraSlots = 0). By construction, x_k is always equal to $\lfloor \frac{E(*,k-1)-OP(h)}{\alpha_t \cdot (N_s-1)} \rfloor$. Hence, (40) can be rewritten into

$$E(*, k) = \psi \cdot (E(*, k-1) - OP(h)) \tag{41}$$

where $\psi = \frac{\beta_t}{\alpha_t \cdot (N_s-1)}$. By substituting (41) into (39) and iterating over k, we get the following condition:

$$\psi^k \cdot \left(E_{\mathrm{init}} - OP(h) + \frac{OP(h)}{1-\psi} \right) - \frac{OP(h) \cdot \psi}{1-\psi} < \alpha_t \cdot (N_s - 1). \tag{42}$$

Since the ROUND_ROBIN_STAGE ends as soon as the condition mentioned above becomes true, we derive the value of k solving the equation. Let K be such a value. After some calculations (not reported here for space reasons), we obtain the expression of K reported in (8).

Appendix B
Proof of Theorem 1

In order to compute T_{RR}, we consider the term $x_{\mathrm{RR}} = \sum_{i=0}^{K-1} x_i$, which is the total number of charging slots assigned to each UAV during the ROUND_ROBIN_STAGE. From (41), we can derive the energy at step k, as follows:

$$E(*, k) = \psi^k \cdot E_{\mathrm{init}} - OP(h) \cdot \frac{\psi - \psi^{k+1}}{1-\psi}. \tag{43}$$

Through (43) and the definition of x_k, we can derive x_{RR} as

$$\begin{aligned} x_{\mathrm{RR}} &= \left\lfloor \sum_{k=0}^{K-1} \frac{(E(*, k) - OP(h))}{\alpha_t \cdot (N_s - 1)} \right\rfloor \\ &= \left\lfloor \sum_{k=0}^{K-1} \frac{\left(\psi^k \cdot E_{\mathrm{init}} - OP(h) \cdot \frac{\psi - \psi^{k+1}}{1-\psi} - OP(h) \right)}{\alpha_t \cdot (N_s - 1)} \right\rfloor \\ &= \left\lfloor \frac{\left(E_{\mathrm{init}} + \frac{OP(h) \cdot \psi}{1-\psi} \right) \cdot (1 - \psi^K) - K \cdot OP(h)}{\alpha_t \cdot (N_s - 1) \cdot (1 - \psi)} \right\rfloor. \end{aligned}$$

Assuming that all the N_s UAVs will recharge of the same energy amount (i.e., extraRounds will always be equal to 0), we can derive T_{RR} as follows:

$$T_{\mathrm{RR}} = x_{\mathrm{RR}} \cdot N_s. \tag{44}$$

When entering the RECHARGE_MINIMUM_STAGE, the energy of maxNode is lower than $\alpha_t \cdot (N_s - 1)$. Since each active node will discharge of α_t energy units at each slot, we have that at most $N_s - 1$ can be completed till one UAV will drain its energy, hence: $0 \leq T_{\mathrm{MIN}} < N_s - 1$. Combining this result with (44), we have the statement of Theorem 1.

Appendix C
Proof of Corollary 1

The proof is derived from Lemma 1. At each iteration of the RECHARGE_ROBIN_STAGE, all the N_s UAVs enter the charging state exactly once. Since this stage is iterated K times (see Lemma 1), the number of swaps is exactly equal to $K \cdot N_s$. Vice versa, in the RECHARGE_MINIMUM_STAGE, a swap can occur at each slot, since the minimum UAV is selected. Since $T_{\mathrm{MIN}} < N_s - 1$, we have that the total number of swaps is upper bound by $K \cdot N_s + N_s - 1$.

Appendix D
Proof of Theorem 2

We observe that the network lifetime is maximized when all the UAVs discharge at the same rate, i.e., when the difference between the energy of the most charged and least charged UAVs is minimized. We indicate with Δ such difference. Let MINSCHED a scheduler selecting the minimum energy node at each slot j (i.e., $s(a_i, t_j) = 1$ if $i =$ getMinEnergyNode() $\forall j$). We notice that MINSCHED is optimal in terms of maximum lifetime, and that $\Delta \leq \alpha_t + \beta_t$. We now prove that such a condition on Δ is also guaranteed by Algorithm 1. In the RECHARGE_MINIMUM_STAGE, our Algorithm follows the MINSCHED policy; hence, the condition is always satisfied at each slot. In the RECHARGE_ROBIN_STAGE, the condition might not hold at each slot. However, we notice that in the `allocateRoundCharge` method, the difference of roundSize (i.e., of charging slots) between two UAVs is at most equal to one. This implies that, at the end of each iteration, we still have that $\Delta \leq \alpha_t + \beta_t$.

Appendix E
Proof of Theorem 3

By Theorem 1, we show that the lifetime is maximized when charging operations are scheduled according to a round robin policy. Let roundSize$[i, j]$ be the duration of the charge—in terms of number slots—for UAV i at iteration j. In Algorithm 1, roundSize$[i, j]$ is computed according to the allocateRoundCharge method. By absurd, let MINSWAP be another scheduler providing a number of swaps lower than $N_s \cdot K$, but guaranteeing the same lifetime than Algorithm 1. Since at each iteration, the number of swaps is constant, and equals N_s, we deduce that MINSWAP performs less iterations than Algorithm 1, which implies that for a given k, roundSize$[i, k]_{\mathrm{MINSWAP}} \geq$ roundSize$[i, k]_{\mathrm{Algorithm1}}, \forall i$. However, this is not possible, since by construction, Algorithm 1 computes the maximum duration of roundSize$[i, k]$ so that the last UAV going to recharge at iteration k will not drain its battery before the end of the iteration.

Appendix F
Proof of Theorem 4

We assume that N_S is always greater then $N_{\min} + 1$, where $N_{\min}$ is defined in (7). At each time slot $t_j < t_{\mathrm{final}}$, we have always $N_s - 1$ UAVs in the state s_{fly} and 1 UAV in state s_{rec} (lines 1–1 of Algorithm 1). Since the centralized approach places the UAVs in an hexagonal pattern, each UAV a_i being in state s_{fly} at time slot t_j uniquely covers at least a surface of

$$\mathrm{Surf}(t_j, a_i, h, \theta) \geq \left(h \cdot \tan\left(\frac{\theta}{2}\right)\right)^2 \cdot \frac{3 \cdot \sqrt{3}}{2}. \tag{45}$$

With this result, we can rewrite (6) as follow:

$$C(t_j) \geq \mathrm{Surf}(t_j, a_i, h, \theta) \cdot (N_S - 1) \tag{46}$$

and hence, we have that $\rho_j \geq \kappa, \forall t_j < t_{\mathrm{final}}$.

Appendix G
Proof of Theorem 5

Let us consider the values of $p_B^{i,j}$ defined by (12) and $p_R^{k,j}$ as known. From (13), we can derive a system of $N_S - 1$ equations with $N_S - 1$ unknown variables, i.e., the $p_G^{i,j}$ variables, with $1 \leq i \leq N_S, i \neq k$. We have $\forall t_j \in T$:

$$\begin{cases} p_B^{1,j} = 1 - \left(p_R^{k,j} \cdot \prod_{a_h \in A \setminus \{a_1\}} (1 - p_G^{h,j})\right) \\ p_B^{2,j} = 1 - \left(p_R^{k,j} \cdot \prod_{a_h \in A \setminus \{a_2\}} (1 - p_G^{h,j})\right) \\ \cdots \\ p_B^{N_S,j} = 1 - \left(p_R^{k,j} \cdot \prod_{a_h \in A \setminus \{a_{N_S}\}} (1 - p_G^{h,j})\right). \end{cases} \tag{47}$$

If $\phi(t_{j-1}) = 1$, then $p_R^{k,j} = 1$; hence, we have again a system of N_S equations with N_S unknown variables. The solution of such system is the one presented in Theorem 5 [see (14)].

We can prove it by construction, i.e., by substituting (14) into any equation of the system of equations (47), verifying

$$p_B^{i,j} = 1 - \left(p_R^{k,j} \cdot \prod_{a_h \in A \setminus \{a_i\}} \sqrt[(N_S - 1)]{\frac{\prod_{a_q \in A \setminus \{a_h\}} (1 - p_B^{q,j})}{(1 - p_B^{h,j})^{N_S - 2} \cdot p_R^{k,j}}}\right). \tag{48}$$

We need to remark that (14) can become inconsistent for UAV a_i, i.e., $p_G^{i,j} < 0$, when the following condition holds:

$$\prod_{a_h \in A \setminus \{a_i\}} (1 - p_B^{h,j}) > (1 - p_B^{i,j})^{N_S - 2} \cdot p_R^{k,j}. \tag{49}$$

In such case, we transform the mixed strategy in a pure strategy, i.e., the support of the mixed strategy consists in a single action, and (14) is rewritten as follows:

$$p_G^{i,j} = \mathrm{MAX}\left(0, 1 - \sqrt[(N_S - 1)]{\frac{\prod_{a_h \in A \setminus \{a_i\}} (1 - p_B^{h,j})}{(1 - p_B^{i,j})^{N_S - 2} \cdot p_R^{k,j}}}\right). \tag{50}$$

APPENDIX H
PROOF OF THEOREM 6

Similar to the previous case, by substituting (14) in (20), we obtain

$$p_{T_0}^{k,j} = \prod_{a_h \in A \setminus \{a_k\}} \left(\sqrt[(N_S-1)]{\frac{\prod_{a_q \in A \setminus \{a_h\}} (1 - p_B^{q,j})}{(1 - p_B^{h,j})^{N_S-2} \cdot p_R^{k,j}}} \right). \quad (51)$$

After some arithmetic calculus, it is easy to notice that (22) is a rearrangement of (51).

As for the previous proof, we have to remark that the equation mentioned above can become inconsistent for the UAV a_k, i.e., $p_R^{k,j} > 1$, if the following condition holds:

$$\prod_{a_h \in A \setminus \{a_k\}} (1 - p_B^{h,j}) > (p_{T_0}^{k,j})^{N_S-2}. \quad (52)$$

In this case, we transform the mixed strategy in a pure strategy, i.e., the support of the mixed strategy is formed by only one action. More formally (22) is rewritten as follows:

$$p_R^{k,j} = \mathrm{MIN} \left(1, \sqrt[(N_S-1)]{\frac{\prod_{a_h \in A \setminus \{a_k\}} (1 - p_B^{h,j})}{(p_{T_0}^{k,j})^{N_S-2}}} \right). \quad (53)$$

REFERENCES

[1] Valour consultancy, The future of commercial and industrial UAVs. Brochure, 2016. [Online]. Available: http://www.valourconsultancy.com/wp-content/uploads/2015/12/The-Future-of-UAVs-Drones-Report-Brochure.pdf

[2] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison, and D. H. Shim, "Persistent UAV service: An improved scheduling formulation and prototypes of system components," *J. Intell. Robot. Syst.*, vol. 74, no. 1, pp. 221–232, 2014.

[3] H. Shakhatreh, A. Khreishah, J. Chakareski, H. B. Salameh, and I. Khalil, "On the continuous coverage problem for a swarm of UAVs," in *Proc. 37th IEEE Sarnoff Symp.*, Princeton, NJ, USA, 2016, pp. 130–135.

[4] A. Trotta, M. Di Felice, K. Chowdhury, and L. Bononi, "Fly and recharge: Achieving persistent coverage using small unmanned aerial vehicles (SUAVs)," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, 2017, pp. 1–7.

[5] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the Sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan.–Mar. 2017.

[6] A. Osseiran *et al.*, "Scenarios for 5G mobile and wireless communications; the vision of the METIS project," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26–35, May 2014.

[7] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.

[9] M. Di Felice, A. Trotta, L. Bedogni, K. R. Chowdhury, and L. Bononi, "Self-organizing aerial mesh networks for emergency communication," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Washington, DC, USA, 2014, pp. 1631–1636.

[10] J. Yu, S. J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 327–341, Feb. 2015.

[11] K. Derr and M. Manic, "Extended virtual spring mesh (EVSM): The distributed self-organizing mobile ad hoc network for area exploration," *IEEE Trans. Ind. Electron.*, vol. 58, no. 12, pp. 5424–5437, Dec. 2011.

[12] M. Asadpour, B. Van den Bergh, D. Giustiniano, K. A. Hummel, S. Pollin, and B. Plattner, "Micro aerial vehicle networks: An experimental analysis of challenges and opportunities," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 141–149, Jul. 2014.

[13] R. Shinkuma and Y. Goto, "Wireless multihop networks formed by unmanned aerial vehicles with separable access points and replaceable batteries," in *Proc. IEEE 7th Annu. Ubiquitous Comput. Mobile Commun. Conf.*, New York, NY, USA, 2016, pp. 1–6.

[14] J. Leonard, A. Savvaris, and A. Tsourdos, "Energy management in swarm of unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircraft Syst.*, 2013, pp. 124–133.

[15] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tut.*, vol. 18, no. 2, pp. 1123–1152, Secondquarter 2016.

[16] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," in *Proc. IEEE HotPower*, Vancouver, BC, Canada, 2010, pp. 1–5.

[17] J. Kim, B. D. Song, and J. R. Morrison, "On the scheduling of systems of UAVs and fuel service stations for long-term mission fulfillment," *J. Intell. Robot Syst.*, vol. 70, no. 1, pp. 347–359, 2013.

[18] J. Scherer and B. Rinner, "Persistent multiUAV surveillance with energy and communication constraints," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Fort Worth, TX, USA, 2016, pp. 1225–1230.

[19] "The Role of Deployable Aerial Communications Architecture in Emergency Communications and Recommended Next Steps," White Paper, Federal Communications Commission (FCC), Washington, DC, USA, 2011.

[20] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub, and M. Anwander, "UAVNet: A mobile wireless mesh network using unmanned aerial vehicles," in *Proc. IEEE Globecom Workshops*, Anaheim, CA, USA, 2012, pp. 1603–1608.

[21] M. Alzenad, A. El-Key, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.

[22] E. Yanmaz, "Connectivity versus area coverage in unmanned aerial vehicle networks," in *Proc. IEEE Int. Conf. Commun.*, Ottawa, ON, Canada, 2012, pp. 719–723.

[23] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of UAVs," in *Proc. IEEE Int. Conf. Auton. Robot Syst. Competitions*, Vila Real, Portugal, 2015, pp. 111–117.

[24] K. Daniel, S. Rohde, N. Goddemeier, and C. Wietfield, "Cognitive agent mobility for aerial sensor networks," *IEEE Sensors J.*, vol. 11, no. 11, pp. 2671–2682, Nov. 2013.

[25] N. Nigam, S. Bicniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, Sep. 2012.

[26] C.-I. Vasile and C. Belta, "An automata-theoretic approach to the vehicle routing problem," in *Proc. Robot. Sci. Syst.*, 2014, pp. 1–9.

[27] K. A. Swieringa *et al.*, "Autonomous battery swapping systems for small-scale helicopters," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, 2010, pp. 3335–3340.

[28] K. Yu, A. K. Budhiraja, and P. Tokekar, "Algorithms for routing of unmanned aerial vehicles with mobile recharging stations and for package delivery," *Computing Research Repository (CoRR)*, arXiv:1704.00079v3, Sep. 2017.

[29] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 128–142, Feb. 2015.

[30] N. K. Ure, G. Chowdhary, T. Toksoz, J. P. How, M. A. Vavrina, and J. Vian, "An automated battery management system to enable persistent missions with multiple aerial vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 1, pp. 275–286, Feb. 2015.

[31] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[32] H. Zhang, Z. Zhang, and H. Dai, "Gossip-based information spreading in mobile networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5918–5928, Nov. 2013.

[33] S. Vasudevan, J. Kurose, and D. Towsley, "Design and analysis of a leader election algorithm for mobile ad hoc networks," in *Proc. 12th IEEE Int. Conf. Netw. Protocols*, 2004, pp. 350–360.

[34] B. Wang, H. B. Lim, and D. Ma, "A survey of movement strategies for improving network coverage in wireless sensor networks," *Comput. Commun.*, vol. 32, no. 1, pp. 1427–1436, 2009.

[35] M. J. Osborne, *An Introduction to Game Theory*, vol. 3, no. 3. New York, NY, USA: Oxford Univ. Press, 2004.

[36] The OMNeT++ tool. Project website. [Online]. Available: http://omnetpp.org. [Accessed on: Feb. 2018].

Angelo Trotta received the Master's degree (*summa cum laude*) in computer science in 2011 and the Ph.D. degree in computer science and engineering in May 2017, both from University of Bologna, Bologna, Italy.

From October 2012 to October 2013, he was a Research Student for the Italian PRIN 2009 project STEM-Net working on the study, modeling, and simulation of cognitive radio technologies and self-organizing wireless networks. In 2015–2016, he was a Visiting Researcher with the Heudiasyc Laboratory, Sorbonne Universits, UTC, Compigne, France. In 2018, he was a Visiting Researcher with the Genesys-Laboratory, Northeastern University, Boston, MA, USA. Currently, he is a Postdoc Research Fellow with the Department of Computer Science and Engineering, University of Bologna, working on nature-inspired algorithms for self-organizing multirobots wireless systems.

Marco Di Felice received the Laurea (*summa cum laude*) and Ph.D. degrees in computer science from University of Bologna, Bologna, Italy, in 2004 and 2008, respectively.

In 2007, he was a Visiting Researcher with Georgia Institute of Technology, Atlanta, GA, USA. In 2009, he was a Visiting Researcher with Northeastern University, Boston, MA, USA. Currently, he is an Associate Professor of computer science with University of Bologna. He authored more than 80 papers on wireless and mobile systems. His research interests include self-organizing wireless networks, unmanned aerial systems, IoT, and mobile applications.

Dr. Di Felice is currently on the editorial board of Elsevier's *Ad Hoc Networks* journal. He was a recipient of the Best Paper Award at the Association for Computing Machinery International Symposium on Mobility Management and Wireless Access in 2012 and the IEEE Annual Mediterranean Ad Hoc Networking Workshop in 2013.

Federico Montori received the Master's degree (*summa cum laude*) in computer science from University of Bologna, Bologna, Italy, in 2015. He is currently working toward the Ph.D. degree at the Department of Computer Science and Engineering, University of Bologna, working on algorithms for Mobile Crowdsensing and architectures for the collaborative IoT within the scope of Smart Cities and environmental monitoring.

He was with the European projects Internet of Energy (from 2012 to 2013) and Arrowhead (from 2014 to 2016) within the scope of IoT platform interoperability in the field of smart energy. In 2015, he held a research grant from the Advanced Research Center on Electronic Systems and was a Visiting Researcher with the Luleå University of Technology, Luleå, Sweden.

Kaushik R. Chowdhury (M'09–SM'15) received the Ph.D. degree from Georgia Institute of Technology, Atlanta, GA, USA, in 2009.

He is currently an Associate Professor and a Faculty Fellow with the Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA. His current research interests include dynamic spectrum access networks and systems, wearables and implant communication, and energy harvesting sensors.

Dr. Chowdhury was a recipient of the Presidential Early Career Award for Scientists and Engineers (PECASE) in January 2017, the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award in 2017, the Office of Naval Research Director of Research Early Career Award in 2016, and the National Science Foundation (NSF) CAREER Award in 2015. He received multiple best paper awards, including three in the IEEE ICC conference, in 2009, 2012, and 2013, and ICNC conference in 2013. He was on the editorial board of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. His research has been supported by the NSF, Office of Naval Research, DARPA, MathWorks, among others.

Luciano Bononi received the M.Sc. degree (*summa cum laude*) in 1997 and the Ph.D. degree from the University of Bologna, Italy, in 2001.

He is an Associate Professor of wireless and mobile systems, mobile applications, and Internet of Things with the Department of Computer Science and Engineering, University of Bologna, Bologna, Italy. He has co-authored more than 110 peer-reviewed conference and journal publications and 8 book chapters, receiving three Best Paper Awards. He is the founder and the Director of the Laboratory of Wireless Systems and Mobile Applications, CSE. His research interests include wireless systems and networks, protocol architectures, Internet of things, Internet of energy, modeling, simulation, performance evaluation, mobile services, and mobile applications.

Dr. Bononi has been involved in more than ten international research projects, and he is an Associate Editor of five international journals and a Guest Editor of more than ten special issues. He was a Chair in more than 15 IEEE/ACM conferences and TPC member in more than 140 IEEE/ACM conferences.