Fly and Recharge: Achieving Persistent Coverage using Small Unmanned Aerial Vehicles (SUAVs)

Angelo Trotta*, Marco Di Felice*, Kaushik R. Chowdhury†, Luciano Bononi*
* DISI, University of Bologna, Italy
† Northeastern University, Boston, USA
Emails: {trotta,difelice,bononi}@cs.unibo.it, krc@neu.edu

***Abstract*—Several applications involving the utilization of Small Unmanned Aerial Vehicles (SUAVs) require stationary and long-term coverage of a target area. Unfortunately, this goal is hard to achieve due the need for coordination and the limited flight autonomy of the SUAVs. In this paper, we investigate how to guarantee persistent coverage of a target area through SUAVs by exploiting characteristics of fixed terrestrial infrastructure and inherent energy limitations. This paper makes three main contributions. First, the problem of SUAV activity scheduling is formulated for pre-existing fixed placements, and centrally solved to maximize the network lifetime given a target coverage ratio. Second, a distributed, bio-inspired algorithm is devised using local (1-hop) communication only, i.e., the scheme takes into account both positioning and charging issues allowing the SUAVs to self-organize into a maximum-coverage connected swarm, and coordinate the charging operations. Third, the performance of the distributed scheme is compared to the optimal solution, and the impact of the system parameters like the placement height and the discharging rate on the coverage metrics is discussed.**

I. Introduction

Device miniaturization and cost reduction are two of the main factors driving the utilization of Small Unmanned Aerial Vehicles (SUAVs) in novel application fields, from emergency communication to traffic and ambient monitoring services [1][2][3]. In most cases, SUAVs are deployed in swarms, and must provide stationary and persistent coverage of target areas. The clear advantages of aerial sensor networks over ground networks include better area oversight and enhanced device-to-device communication by exploiting uninterrupted Line of Sight (LOS) in the wireless links [4]. However, coping with the challenges of 3D mobility and the limited battery autonomy of SUAVs pose formidable research challenges in obtaining a feasible solution in tractable time.

Distributed algorithms like [5] and [6] enable groups of SUAVs to explore the environment and to self-organize into connected swarms. Energy efficiency is typically translated into enhanced communication and mobility protocols that mitigate the network traffic overhead [7] or avoid the unnecessary ascending/descending operations [8]. As an alternative, several recent studies like [9], [10] and [11] propose replenishment strategies by leveraging the presence of charging infrastructures on the ground. In this paper, we follow the second approach to devise a deployment strategy for the aerial networks of SUAVs, able to guarantee user-defined coverage ratio of the scenario while maximizing the network lifetime. This translates into a complex joint positioning and scheduling problem, where the current location, and the start/end of each charging event must be determined for each SUAV.

Our research methodology is composed of three steps. First, we formulate the scheduling problem in a rigorous manner, and we determine an optimal centralized solution assuming that all the SUAVs have global knowledge of the scenario and are placed in pre-assigned positions. Differently from previous studies (e.g. [9] and [11]), we take into account the energy overhead for ascending/descending maneuvers, and hence, we attempt to minimize also the number of swaps of SUAVs during the charging cycles. The exact performance of the algorithm in terms of number of steps and of charge swaps are computed, and its optimality in terms of network lifetime is demonstrated. Second, based on some key observations from the centralized framework, we propose a distributed algorithm that relies on local (1-hop) communication among the SUAVs. Our proposed algorithm borrows models from biological and natural systems [12], and guarantees network self-organization and autonomous coordination of charging operations. Third, we evaluate through simulations the performance of scheduling/mobility algorithms over realistic 3D scenarios. We show that our distributed scheme improves on other similar approaches while providing a network lifetime close to the optimal solution, but with reduced overhead.

The rest of the paper is organized as follows. Section II reviews the main studies on SUAVs regarding stationary coverage and lifetime maximization. Section III describes the system model and formulates the so called CCPANP optimization problem. A centralized solution is proposed in Section IV, and the distributed algorithm is presented in Section V. The performance results are in Section VI. Conclusions are drawn in Section VII.

II. Related Works

Most of the existing studies on SUAVs investigate path planning issues for complete coverage of an area of interest [16]. Conversely, this paper focuses on *stationary* coverage, which is a key requirement for applications like SUAV-based mesh networking [2] or urban sensing [3]. The study in [13] derives some fundamental results about 3D coverage, clarifying the relationship between altitude, beamwidth of the antenna and coverage probability. The trade-off between coverage and connectivity is investigated in [14], where the authors show that guaranteeing both these goals can be challenging in highly

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

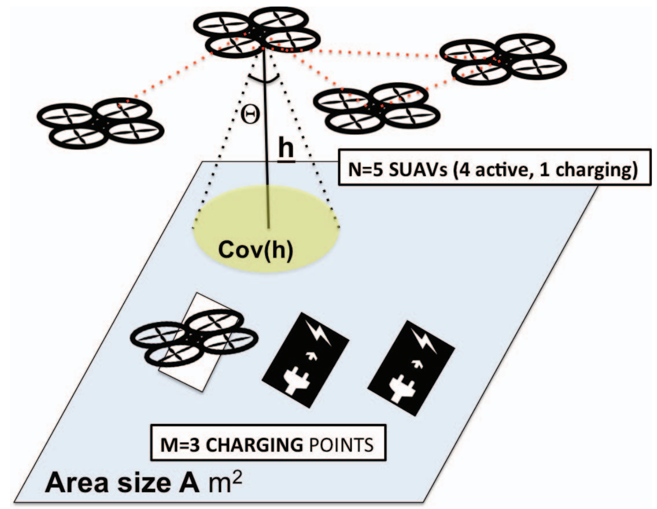


Fig. 1. The aerial mesh network with charging stations on the ground.

sparse networks. Examples of distributed mobility algorithms achieving maximum coverage while maintaining connectivity among the SUAVs are in [5] and [6]. [15] proposes a mobility scheme based on the virtual spring model, while [5] ensures that all the aerial links experience the same Quality of Service (QoS) regardless of the propagation conditions. In [6] a channel-aware swarm mobility scheme is proposed, based on the cluster-breathing technique and on the utilization of the RSSI metric to reflect the link quality. Beside SUAV positioning, lifetime maximization is also a crucial aspect of coverage problems, and has been addressed through two different approaches. Energy-aware protocols minimize unnecessary data transmission or maneuvers of the mobile devices [1]. In [7], the authors examine the problem of optimal beaconing periods, and provide a distributed learning scheme which ensures convergence to the Nash equilibrium point. In [8], an energy model for SUAV operations is discussed, and an energy-aware path planning algorithm meeting constraints on minimal coverage is described and evaluated. Energy replenishment through terrestrial infrastructure is considered, among others, in [9], [10] and [11]. In [9], the SUAV scheduling is modeled through a MILP scheduler, and a guidance system is proposed in order to make the SUAVs land on the charging stations. Similarly, the MILP formulation in [11] allows orchestrating charge and task allocation over group of SUAVs. The work in [9] is the most similar to our paper. However, differently from [9], we derive also the duration of charging operations, and we take into account the overhead for ascent/descent maneuvers. We base our work on specific assumptions on the relationship between the number of the SUAVs and the recharging stations, in order to make the problem solvable in polynomial time.

III. System Model

We consider a square deployment area of size of A m^2, and N SUAVs. Each SUAV is able to sense the environment, communicate wirelessly with other peers, and re-charge its battery at the stations which are located on the ground (at the center of the region). The SUAVs form a connected network h meters from the ground; the impact of h on system performance is investigated in Section VI. Depending on h, each SUAV is able to sense an area $Cov(h)$ equal to:

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

where θ is the angle of the sensing cone depicted in Figure 1. Let M be the number of available charging spots. We consider the non-trivial case with $M < N$. Without loss of generality, we assume a time discretization into slots of equal duration, denoted as T_{slot}. At each slot j, a SUAV i can be *active*, i.e. flying and covering a specific zone, or *charging*, i.e. placed on the ground at one of the M charging spots. We denote with $s(i,j) = 1$ the active state, and with $s(i,j) = 0$ the charging state, for SUAV i and slot j. Let E_C be the battery capacity of each UAV (homogeneous hardware is assumed), and $E(i,j)$ be the residual energy of SUAV i at time slot j. The $E(i,j)$ function is updated according to the following Equation:

$$\begin{aligned} E(i,j) \;\; = \;\; & E(i,j-1) - \alpha \cdot T_{slot} \cdot s(i,j) \\ & + \beta \cdot T_{slot} \cdot (1 - s(i,j)) && (2) \\ & - OP(h) \cdot s(i,j-1) \cdot (1 - s(i,j)) && (3) \\ & - OP(h) \cdot s(i,j) \cdot (1 - s(i,j-1)) && (4) \end{aligned}$$

Here, $E(i,j-1)$ is the energy level in the previous time slot. The second and third terms represent the energy charge/discharge while in active/charging states, assuming a linear energy decrease/increase over the time slot with coefficients equal to α and β. The fourth and fifth terms concern the energy overhead for the descent/ascent operations, i.e. the energy required by an active SUAV to fly to the ground or to fly from the ground to height h. The overheads for descent/ascent operations are assumed equal, and proportional to the network height like in the energy model proposed in [8], i.e. $OP(h) = \gamma \cdot h$. Clearly, the fine tuning of α, β and γ parameters must be based on hardware specifications, and is out of the scope of this paper.

In this work, we are interested in deploying an energy-efficient SUAV network, with constraints in terms of area covered and temporal persistence. Let κ be a threshold on the fraction of the area covered by the aerial network. The Constrained Coverage and Persistence Aerial Network Deployment (CCPANP) problem can be informally defined as: how to determine an optimal charging scheduling function $s(i,j)$, so that: (*i*) the fraction of area covered by the SUAV network at slot j i.e. $\frac{C(j)}{A}$, with $C(j) = \bigcup_{i=1}^{N} Cov(h) \cdot (1 - s(i,j))$, is always greater than κ, for each j between 0 and T, and (*ii*) T is maximized, i.e. the network lifetime is prolonged as much as possible. Formally, the CCPANP problem is defined as:

Definition 1 (The CCPANP Problem). *Given κ, N and M, we want the optimal $s(i,j)$ function $\forall i \in \{0...n\}$, $\forall j \in \{0, ..., T\}$ such that T (the network lifetime) is maximized, and the following constraints are met, $\forall i \in \{0...n\}$, $\forall j \in \{0, ..., T\}$:*

1) $\frac{C(j)}{A} \geq \kappa$
2) $E(i,j) > 0$ *and* $E(i,j) < E_C$
3) $N - \sum_{i=0}^{N} s(i,j) \leq M$

Here, the second condition asserts that no SUAV can run out of battery, while the third condition asserts that at each slot, a maximum of M devices are recharging on the ground.

IV. Centralized Approach

In this section, we provide the optimal solution to the CCPANP problem defined above. First, we observe from [17] that the optimal coverage is achieved when placing the SUAVs according to regular hexagon patterns, with side length equal to $R \cdot \sqrt{3}$. Let N_{min} be the minimum number of SUAVs that is required to guarantee that $\frac{Cov(j)}{A} \geq \kappa$. Using results in [17], we compute N_{min} as follows:

$$N_{min} = \left\lceil \frac{\kappa \cdot A}{\left(h \cdot tan(\frac{\theta}{2})\right)^2 \cdot \frac{3 \cdot \sqrt{3}}{2}} \right\rceil \tag{5}$$

We assume that $N = N_{min} + M$. It is easy to notice that in all the other cases the problem is ill-posed since: (*i*) if $N > N_{min} + M$, the number of SUAVs exceeds the system need, and in some circumstances the network lifetime can be infinitely prolonged by simply swapping the active nodes with the additional (idle) nodes not used for coverage; (*ii*) if $N < N_{min} + M$, the number of charging stations exceeds the system need, i.e. at least one station will remain unused at each slot. The CCPANP algorithm starts by assigning a subset of SUAVs to each station; the cardinality of each subset is equal to $N_s = \frac{N_{min}}{M}$. Using this relation, we reduce the scheduling problem to the case where N_s SUAVs must contend for one single charging station. The case where N_{min} is not multiple of M will be investigated as future work.

Algorithm 1 shows the pseudo-code for solving the CCPANP problem. At each time-slot, the `schedule` method is executed, and the SUAV with id equal to *currentCharge* is charged. Moreover, we check that all SUAVs have energy greater than zero, otherwise the algorithm ends (at line 4). The algorithm can work in two stages, respectively the ROUND_ROBIN_STAGE and the RECHARGE_MINIMUM_STAGE, which can be repeated over several iterations. In the ROUND_ROBIN_STAGE, we let each SUAV 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: (*i*) considering the SUAV with maximum residual energy (line 47); (*ii*) computing the maximum number of slots before such node will drain its energy ($numRoundsPerUAV$); (*iii*) assigning $numRoundsPerUAV$ to all the SUAVs (lines 51-53); (*iv*) in case of residuals, allocate the $extraRounds$ slots to the SUAVs with minimum energy (lines 54-56). We then order all the SUAVs based on their energy in ascending order (let S_{round} be this set at line 58), and in turn we extract the first element from S_{round} (line 21), recharging it of $roundSize[i]$ consecutive slots (line 16-17). Once all the

Algorithm 1: CCPANP centralized algorithm

1: **procedure** `schedule`(slotNumber j)
2: Invoke *decideStatus*(j)
3: **for** $i = 0$ to N_s **do**
4: **if** $E(i,j) \leq 0$ **then**
5: **return** //End of network lifetime
6: **end if**
7: **if** currentCharge == i **then**
8: $s(i,j) \leftarrow 0$ //Node i is *charging*
9: **else**
10: $s(i,j) \leftarrow 1$ //Node i is *active*
11: **end if**
12: **end for**
13:
14: **procedure** `decideStatus`(slotNumber j)
15: **if** status == **ROUND_ROBIN_STAGE then**
16: currentRoundCharge $\leftarrow$ currentRoundCharge + 1
17: **if** currentRoundCharge $>$ roundSize[currentNode] **then**
18: **if** $S_{round} = \emptyset$ **and** isRRStageFeasible(j) **then**
19: $S_{round} \leftarrow$ allocateRoundCharge(j)
20: **end if**
21: currentCharge $\leftarrow$ removeFirst(S_{round})
22: currentRoundCharge $\leftarrow$ 1
23: **if** currentCharge==NULL **then**
24: status $\leftarrow$ RECHARGE_MINIMUM_STAGE
25: **end if**
26: **else**
27: currentRoundCharge $\leftarrow$ currentRoundCharge +1
28: **end if**
29: **end if**
30: **if** status == **RECHARGE_MINIMUM_STAGE then**
31: minNode $\leftarrow$ getMinEnergyNode()
32: **if** $E(minNode, j) < E(currentCharge, j)$ **then**
33: currentCharge $\leftarrow$ minNode
34: **end if**
35: **end if**
36:
37: **procedure** `isRRPhasePossible`(slotNumber j)
38: maxNode $\leftarrow$ getMaxEnergyNode()
39: numRounds $\leftarrow \lfloor \frac{E(maxNode,j) - 2 \cdot OP(h)}{\alpha} \rfloor$
40: **if** numRounds $> N_s - 1$ **then**
41: **return True**
42: **else**
43: **return False**
44: **end if**
45:
46: **procedure** `allocateRoundCharge`(slotNumber j)
47: maxNode $\leftarrow$ getMaxEnergyNode()
48: numRounds $\leftarrow \lfloor \frac{E(maxNode,j) - 2 \cdot OP(h)}{\alpha} \rfloor$
49: numRoundsPerUAV $\leftarrow \lfloor \frac{numRounds}{N_s - 1} \rfloor$
50: extraRounds $\leftarrow numRounds \% (N_s - 1)$
51: **for** all SUAV i in $N_s \setminus \{currentPivot\}$ **do**
52: roundSize[i] $\leftarrow$ numRoundsPerUAV
53: **end for**
54: **for** $k = 0$ to extraRounds **do**
55: minNode $\leftarrow$ getMinEnergyNode()
56: roundSize[minNode] $\leftarrow$ roundSize[minNode] + 1
57: **end for**
58: $S_{round} \leftarrow \{0, ... N_s\}$
59: Order S_{round} based on roundSize in ascendent order
60: **return** S_{round}

SUAVs in S_{round} have been charged once, we check whether the ROUND_ROBIN_STAGE can be iterated again through the `isRRPhasePossible` method (line 37-44); 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, where at each slot the minimum energy SUAV is charged (line 30-34). 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 SUAVs assigned to each charging station. 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:*

$$K = log_\delta \left\lceil \frac{\alpha \cdot (N_s - 1) - \frac{2 \cdot OP(h)}{1-\delta}}{E_C + \frac{2 \cdot OP(h)}{1-\delta}} \right\rceil \tag{6}$$

where $\delta = \frac{\beta}{\alpha \cdot (N_s - 1)}$.

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

$$E(*, k) < \alpha \cdot (N_s - 1) \tag{7}$$

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

$$E(*, k) = E(*, k-1) - x_k \cdot (\alpha \cdot (N_s - 1) - \beta) - 2 \cdot OP(h) \tag{8}$$

where x_k is the number of charging slots assigned to each SUAV (i.e. the $numRoundsPerUAV$ at line 49, assuming $extraSlots = 0$). By construction, x_k is always equal to $\lfloor \frac{E(*,k-1)}{\alpha \cdot (N_s - 1)} \rfloor$. Hence, Equation 8 can be re-written into:

$$E(*, k) = E(*, k-1) \cdot \delta - 2 \cdot OP(h) \tag{9}$$

where $\delta = \frac{\beta}{\alpha \cdot (N_s - 1)}$. By substituting Equation 9 into Equation 7 and iterating over k, we get the following condition:

$$\delta^k \cdot \left(E_C + \frac{2 \cdot OP(h)}{1-\delta} \right) - \frac{2 \cdot OP(h)}{1-\delta} < \alpha \cdot (N_s - 1) \tag{10}$$

Since the ROUND_ROBIN_STAGE ends as soon as the condition above becomes true, we derive the value of k solving the equation. Let K be such value. After some calculations (not reported here for space reasons), we obtain the expression of K reported in Equation 6. $\square$

Theorem 1 (**Network Lifetime**). *The network lifetime T of Algorithm 1 is in range: $\{T_{RR}...T_{RR} + N_s - 1\}$.*

Proof. 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=$T_{RR} + T_{MIN}$. In order to compute T_{RR}, we consider the term $x_{RR} = \sum_{i=0}^{K-1} x_i$, which is the total number of charging slots assigned to each SUAV during the ROUND_ROBIN_STAGE. Through Equation 9 and the definition of x_i, we can derive x_{RR} as:

$$\begin{aligned} x_{RR} &= \left\lfloor \frac{1}{\alpha \cdot (N_s - 1)} \cdot \sum_{i=0}^{K-1} E(*, i) - 2 \cdot K \cdot OP(h) \right\rfloor \\ &= \left\lfloor \frac{E_C}{\alpha \cdot (N_s - 1)} \cdot \sum_{i=0}^{K-1} \delta^i - 2 \cdot K \cdot OP(h) \right\rfloor \\ &= \left\lfloor \frac{E_C \cdot (1 - \delta^K)}{\alpha \cdot (N_s - 1) \cdot (1 - \delta)} - 2 \cdot K \cdot OP(h) \right\rfloor \end{aligned} \tag{11}$$

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

$$T_{RR} = x_{RR} \cdot N_s \tag{12}$$

When entering the RECHARGE_MINIMUM_STAGE, the energy of $maxNode$ is lower than $\alpha \cdot (N_s - 1)$. Since each active node will discharge of α energy units at each slot, we have that at most $N_s - 1$ can be completed till one SUAV will drain its energy, hence: $0 \leq T_{MIN} < N_s + 1$. Combining this result with Equation 12, we have the statement of Theorem 1. $\square$

Corollary 1 (**Number of swaps**). *The maximum number of charge swaps, i.e. of the number of changes of the SUAV currently under charge, is in range $\{N_s \cdot K ... N_s \cdot K + N_s - 1\}$ for the Algorithm 1 (K is the value given by Equation 6).*

Proof. The proof is derived from Lemma 1. At each iteration of the RECHARGE_ROBIN_STAGE, all the N_s SUAVs enter the charging state exactly once. Since this stage is iterated K times (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 SUAV is selected. Since $T_{MIN} < N_s - 1$, we have that the total number of swaps is upper bound by $K \cdot N_s + N_s - 1$. $\square$

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 $\gamma = 0$, i.e. with no the penalty for swap operations. Then, we prove that Algorithm 1 minimizes the number of swap operations with $\gamma > 0$.

Theorem 2 (**Optimality1**). *If $\gamma = 0$, then Algorithm 1 guarantees the maximum lifetime, i.e. T is maximum.*

Proof. We observe that the network lifetime is maximized when all the SUAVs discharge at the same rate, i.e. when the difference between the energy of the most charged and least charged SUAVs is minimized. We indicate with Δ such difference. Let $MINSCHED$ a scheduler selecting the minimum energy node at each slot j (i.e. $s(i, j) = 1$ if i=*getMinEnergyNode()* $\forall j$). We notice that

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

$MINSCHED$ is optimal in terms of maximum lifetime, and that $\Delta \leq \alpha + \beta$. 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 satified 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 SUAVs is at most equal to one. This implies that, at the end of each iteration, we still have that $\Delta \leq \alpha + \beta$. □

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

Proof. 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 SUAV 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 equal to N_s, we deduce that $MINSWAP$ performs less iterations than Algorithm 1, which implies that for a given k, $roundSize[i, k]_{MINSWAP} \geq roundSize[i, k]_{Algorithm1}, \forall i$. However, this is not possible, since by construction, Algorithm 1 computes the maximum duration of $roundSize[i, k]$ so that the last node going to recharge at iteration k will not drain its battery before the end of the iteration. □

V. Distributed approach

Despite its optimality, Algorithm 1 is not conducive for easy implementation since it assumes strict coordination among the SUAVs, and global knowledge of the scenario. In the following, we describe an alternative, distributed approach inspired from some key findings from the literature [17] and from Algorithm 1 (i.e., optimal charge duration), but relying on local data dissemination mechanisms only. Our solution includes mechanisms for the distributed positioning of the SUAVs and for the coordination of the recharge operations. These two components are described separately as follows.

A. UAV Positioning

We assume that each SUAV is equipped with GPS and WI-FI modules, so that it knows its position, and communicates with other peers using the ad-hoc mode. Every T_{BEACON} seconds, each SUAV i broadcasts a `BEACON` message containing its identifier, its current energy level ($E(i)$), and its position ($\vec{x_i}$). The SUAV positioning algorithm extends the virtual spring model described in [5][15]. A virtual spring force $\vec{F}(i, j)$ acts between each couple of SUAVs (i, j) that are located at 1-hop distance, i.e. that are able to exchange the `BEACON` messages. The intensity of $\vec{F}(i, j)$ is computed by SUAV i and j according to the Hooke's law:

$$|\vec{F}(i, j)| = -(|\vec{x_i} - \vec{x_j}| - d_{EQ}) \cdot k_{ST} \tag{13}$$

Here, the first term is the spring *displacement*, given by the difference between the current distance from SUAV i to SUAV j 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 $Cov(h)$ in Equation 1), which is the distance among the SUAVs when they are positioned according to hexagonal patterns for the optimal scenario coverage [17]. The force is attractive when the distance between the SUAVs 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 SUAV i gathers the `BEACON` messages from its 1-hop neighbours; let NE indicate this set. Then, it determines $\vec{F}(i, j)$ for each neighbour $j \in NE$, and it computes the resultant force $\vec{R}(i) = \sum_{j=0}^{|NE|} \vec{F}(i, j)$. If the module of $\vec{R}(i)$ is greater than a threshold value (modeling the inertia), then $SUAV(i)$ moves on the direction of the resultant force of a fixed step. In this way the proposed method is able to face with the presence of noise in the communication channel avoiding oscillations in the vehicle movements.

B. UAV Charge Scheduling

In the distributed approach, each SUAV decides autonomously when it is time to recharge, and it broadcasts a `LEAVE` message to inform its 1-hop neighbours. In case all charging stations are busy, the SUAV i flies back to its previous location and remains in the *active* state. Otherwise, it enters the charging state for $chargeLength(i)$ number of slots. The value of $chargeLength(i)$ is computed according to Algorithm 1 (line 49), considering local information only:

$$chargeLength(i) = \left\lfloor \frac{E_{avg}^{|NE|} - 2 \cdot OP(h)}{\alpha \cdot (N-1)} \right\rfloor \tag{14}$$

Here, $E_{avg}^{|NE|}$ is the average energy value among the $|NE|$ neighbours, while α is the discharge rate of Section III. After the replenishment, the SUAV flights back to its previous location, and all the neighbours adjust their positions according to the distributed virtual-spring model.

Differently from Algorithm 1, the distributed scheduler is probabilistic and is modeled according to the division of labour model described in [12], and deriving from the Swarm Intelligence (SI) discipline. The original model explains how different tasks are dynamically allocated in colonies of insects presenting morphological differences among the individuals. Here, each task is associated a *stimulus* function $S(T)$ expressing the need of executing it at time T. The probability of performing the task (p_i) by each individual i is:

$$p_i = \frac{S(T)}{\theta_i + S(T)} \tag{15}$$

Here, θ_i represents the *response* function, i.e. the ability of individual i on performing the task. It is easy to notice that the task is likely executed by individual i when $S(T) >> \theta_i$. In our scheduling problem, the function θ_i reflects the need for keeping SUAV i in *active* state. Clearly, this need depends on the network coverage requirements, and more specifically

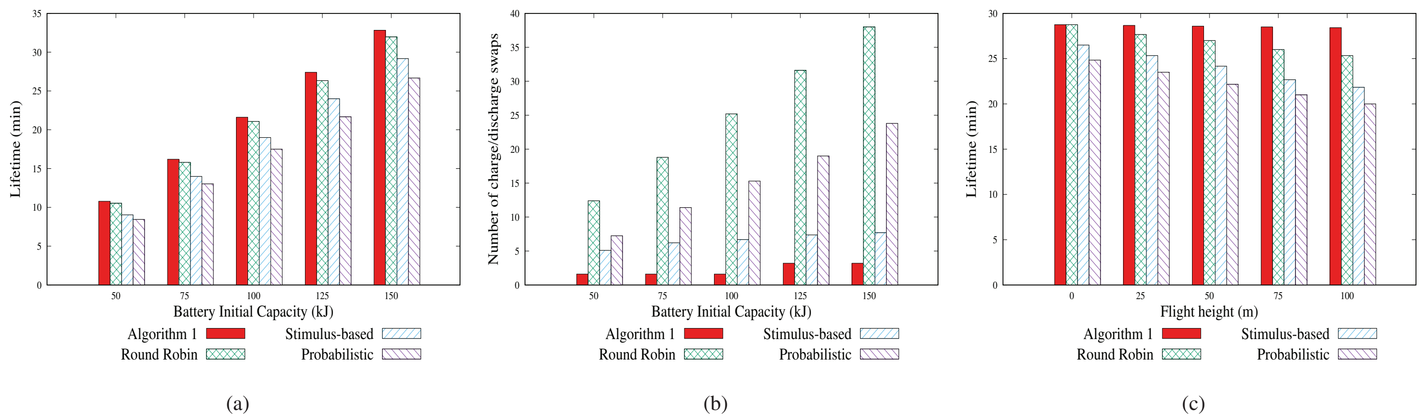


Fig. 2. The network lifetime as a function of E_C (capacity) and h (height) is shown in Figure 2(a) and 2(c). The number of charge swaps is in Figure 2(b).

on the amount of area that is covered exclusively by SUAV i. Unfortunately, computing such area is a complex geometric problem for more than three nodes. For this reason, we consider an alternative formulation that is still based on the same rationale, but uses the degree of each SUAV. We observe that the positioning algorithm described in Section V-A attempts to deploy the SUAVs in hexagonal cells, so that the SUAVs located at internal positions of the aerial network will have six neighbors on average. Vice versa, the SUAVs located on borders will have less than six neighbors and will likely cover a larger area, hence they should be mapped to larger values of θ_i. The proposed formulation of θ_i is:

$$\theta_i = max\left\{\frac{|NE|}{6}, 1\right\} \tag{16}$$

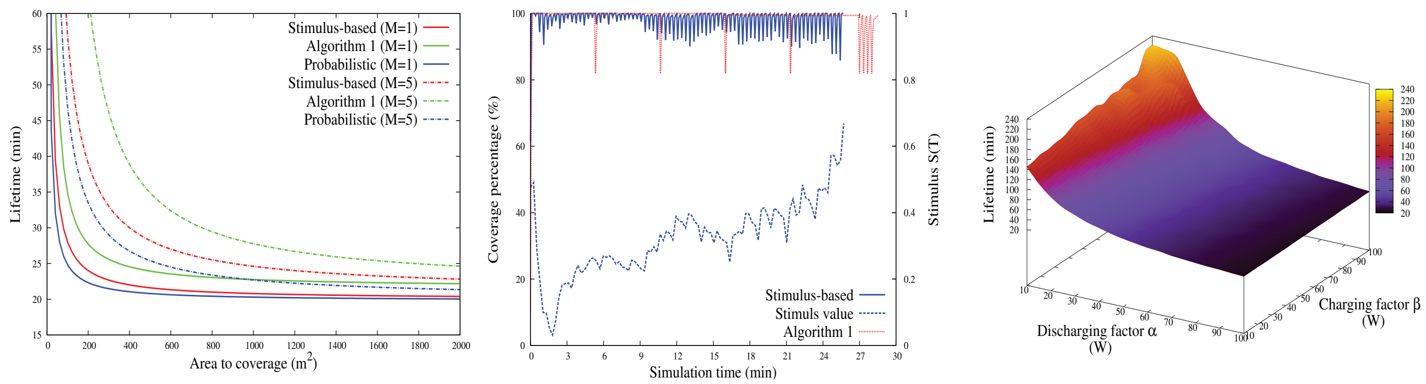
We now formulate the stimulus function of Equation 15, which refects the need for charge of SUAV i. In the proposed algorithm, $S(T)$ is computed as follows:

$$S(T) = \left(\frac{T - T_{last}}{chargeLength(i) \cdot |NE|}\right)^{\frac{E(i)-E^{|NE|}_{min}}{E^{|NE|}_{max}-E^{|NE|}_{min}}} \tag{17}$$

Here, T is the current time slot, while T_{last} is the time-slot of the last charge event of SUAV i. The exponent in Equation 17 compares the energy level of SUAV i with those of the maximum charged/minimum charged peers among the $|NE|$ neighbours. Combining the action of the base and exponent, the stimulus $S(T)$ increases when: (*i*) the SUAV i has not charged for a long time; (*ii*) the residual energy of SUAV i is significantly lower than its neighbours.

VI. Performance Evaluation

In this Section, we evaluate different solutions for the CCPANP problem through a simulation study. We implement a 3D network scenario in OMNeT++, deploying simulation models of SUAV mobility, battery and wireless communication protocols. We compare four different solutions: (*i*) a centralized optimal solution based on Algorithm 1, (*ii*) a centralized round-robin scheduler; (*iii*) a pure distributed probabilistic scheme, in which each SUAV i decides whether to recharge at step j with probability equal to $\frac{E(i,j)}{E_C}$; (*iv*) a distributed probabilistic scheme using the stimulus-response model proposed in Section V. For the round-robin and probabilistic schemes, we consider a fixed charging time duration equal to one time slot. Unless stated otherwise, we used the following system parameter's setting: N=15, M=3, α=100 W, β=25 W, γ=5 J/m, T_{slot}=20 s, E_c=130 kJ, θ=$\frac{\pi}{3}$, κ=100, h=30 m, k_{ST}=1 (we modeled a SUAV equipped with a generic 3-cell ($3S$) LiPo $11.1V$ battery with $3250mAh$ with an approximated flight autonomy of 20 minutes). Figure 2(a) shows the network lifetime as a function of the E_C parameter, i.e. of the SUAV battery capacity. The lifetime metric is computed as the time interval from the simulation start till the first SUAV runs out of battery. Clearly, the lifetime increases with E_C, and Algorithm 1 provides the optimal performance. The stimulus-based algorithm outperforms the probabilistic over all the configurations, and approaches the centralized round robin, which however relies on global coordination among the SUAVs. Figure 2(b) extends the previous analysis, by showing the cumulative number of charge swaps. Algorithm 1 minimizes the number of swaps (Theorem 2), and this explains the lifetime gap with the round-robin scheduler, which instead performs a swap at each time step. Similarly, the stimulus-based scheme outperforms the pure probabilistic one, since each SUAV takes into account the average energy level in its 1-hop neighborhood before issuing charging operations. However, it is worth highlighting that in both the distributed schemes, SUAVs might fail in finding an available charging station, introducing some extra ascent/descent maneuvers. Figure 2(c) confirms the previous trends by showing the network lifetime as a function of the network height (h). The network lifetime decreases for larger values of h as a consequence of the additional overhead of ascent/descent operations (Equation 11). Figure 3(a) depicts the lifetime metric when varying the scenario area size (A) in m^2. We scale the number of available SUAVs (i.e. N) proportionally with A (where N is derived for Equation 5), and we consider two configurations of the number of charging stations (i.e. M=1 and M=5). We observe that regardless of the algorithm in use, the network lifetime



(a) (b) (c)

Fig. 3. The network lifetime as a function of A (area size) and of α-β is shown in Figure 3(a) and Figure 3(c). The coverage over time is in Figure 3(b).

decreases and then stabilizes on a threshold value, which depends on M only, and is not affected by the number of available SUAVs. Again, our stimulus-based algorithm improves upon the purely distributed one over all the configurations. Figure 3(b) shows the fraction of area covered by the SUAVs over the simulation time, comparing Algorithm 1 against the distributed stimulus-response scheme. For this, we also show on the x-$y2$ axis, the average value of the stimulus function over time (Equation 17). It is easy to notice that both algorithms meet -on average- the coverage requirement (κ=1), and that Algorithm 1 extends the network lifetime of around five minutes. The spikes in Figure 3(b) correspond to charge swap attempts. In Algorithm 1, charge swaps occur at fixed time intervals and involves always $\frac{N}{M}$ nodes. Moreover, they increase in frequency near the simulation end when the algorithm enters the RECHARGE_MINIMUM_STAGE. In the stimulus-distributed scheme, charge swaps are attempted asynchronously, based on the energy value of each SUAV; as a result, spikes becomes more frequent as the number of stimuli increases. Finally, Figure 3(c) shows the impact of the discharging/charging factors on the network lifetime experienced by the stimulus-based algorithm. As expected, the lifetime is maximized at the maximal ratio of $\frac{\beta}{\alpha}$.

VII. Conclusions

This paper addressed the deployment of aerial mesh networks for long-term, stationary area coverage through an intelligent energy replenishment strategy using ground charging stations. Two scheduling policies were proposed and evaluated: an optimal centralized policy, assuming perfect positioning and coordination between SUAVs, and a distributed policy based on self-organization principles and decentralized charging control. Future works will focus on the refinement of the energy models considering non-linear charging/discharging operations, the extension to 3D network scenarios with SUAVs located at different altitudes, and the realization of a test-bed.

Acknowledgment

This work was supported in part by the U.S. Office of Naval Research under grant number N000014-17-1-20416.

References

[1] Y. Zeng, R. Zhang and T. Joon Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communication Magazine*, 54(5), pp. 36-42, 2016.
[2] Federal Communication Commission (FCC). Deployable Aerial Communications Architecture in Emergency Communications. 2011.
[3] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub and M. Anwander. UAVNet: a mobile wireless mesh network using unmanned aerial vehicles. *Proc. of IEEE Globecom, Anaheim, USA, 2012.*
[4] E. Yanmaz, R. Kuschnig, and C. Bettstetter. Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility. *Proc. of IEEE INFOCOM, Turin, Italy, 2013.*
[5] M. Di Felice, A. Trotta, L. Bedogni, K.R. Chowdhury and L. Bononi. Self-Organizing Aerial Mesh Networks for Emergency Communication. *Proc. of IEEE PIMRC, Washington, USA.*
[6] K. Daniel, S. Rohde, N. Goddemeier and C. Wietfield. Cognitive agent mobility for aerial sensor networks. *IEEE Sensors Journal: 11(11), pp. 2671-2682, 2013.*
[7] S. Koulali, E. Sabir, T. Taleb and M. Azizi. A green strategic activity scheduling for UAV networks: a sub-modular game perspective. *IEEE Communication Magazine*, 54(5), pp. 58-64, 2016.
[8] C. Di Franco and G. Buttazzo. Energy-aware coverage path planning of UAVs. *Proc. of IEEE ICARSC*, Vila Real, Portugal, 2015.
[9] 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. *Journal of Intelligent & Robotic Systems: 74(1), pp. 221-232, 2014.*
[10] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci. Network connectivity and area coverage for UAV fleet mobility model with energy constraint. *Proc. of IEEE WCNC: 74(1), Doha, Qatar, 2016.*
[11] J. Kim and J. R. Morrison. On the concerted design and scheduling of multiple resources for persistent UAV operations. *Proc. of IEEE ICUAS, Atlanta, USA, 2013.*
[12] E. Bonabeau, M. Dorigo and G. Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. *Oxford University Press*, 1999.
[13] M. Mozaffari, W. Saad, M. Bennis and M. Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communication Letters, 20(8), pp.1647-1650, 2016.*
[14] E. Yanmaz. Connectivity versus area coverage in unmanned aerial vehicle networks. *Proc. of IEEE ICC*, Ottawa, Canada, 2012.
[15] K. Derr and M. Manic. Extended virtual spring mesh (EVSM): The distributed self-organizing mobile ad hoc network for area exploration. *IEEE Trans. on Industrial Electronics*: 58(12), pp. 5424-5437, 2011.
[16] N. Nigam, S.Bicniawski, I. Kroo and J. Vian. Control of multiple UAVs for persistent surveillance: algorithm and flight test results. *IEEE Transactions on Control System Technology*, 20(5), pp. 1236-1251, 2012.
[17] B. Wang, H. Beng Lim, D. Ma. A survey of movement strategies for improving network coverage in wireless sensor networks. *Computer Communications*, 32(1), pp. 1427-1436, 2009.