2020 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. September 1-4, 2020

Spare Drone Optimization for Persistent Task Performance with Multiple Homes

Erez Hartuv[1] Noa Agmon[1] Sarit Kraus[1]

***Abstract*— In this paper we examine the problem of persistent task performance by a team of multiple drones, where the drones suffer from energy limitations. The drones are required to occupy a set of m locations in order to perform a task, for example surveillance, indefinitely. Since the drones have a limited battery supply, they must be replaced in order to refuel, recharge or change their battery at a fixed set of refueling stations called homes. Therefore, in order to enable the persistent task performance, it is essential to add spare drones to the system that will replace the drones in their task. We examine two problems in this context: determining the minimal number of spare drones that will guarantee that the task will be carried out persistently and indefinitely, and finding a schedule for drone-replacements. The novelty of this work is twofold: (i) Proving that a simple drone replacement schedule is enough with respect to minimizing the number of spare drones, thus reducing the need to cope with $O(m^2)$ pairwise travel costs of the given m locations to only $O(m)$ travel costs between the m locations and the homes; and (ii) The introduction of an innovative approximation approach for the minimum number of spare drones required, and providing a replacement scheduling strategy by combining a Voronoi tessellation with a Bin-Packing variant (Bin Maximum Item Double Packing-BMIDP) for the Multi-homes problem, which is much harder than the single home problem and is NP-Hard even for a single spare drone.**

I. INTRODUCTION

The growing interest in drone technology has brought about the development of new fields of application for it. With the growing technology, many new sensors and actuators are being deployed on drones so that their operation can be highly optimized and they can be used for dedicated applications with high performance. Drones can be used to efficiently monitor and survey areas, and for continuous surveillance of a disaster scene such as flooding and forest fires [1], civil security operations, traffic monitoring [2] and event photography [3]. The main problem in using drones for such applications is their limited flight time. To overcome this severe limitation in persistent task execution, spare drones should be available to replace drones that are running low on battery. The replaced drones could fly to a location where their batteries can be charged or replaced, enabling them to continue in their task [4].

This paper discusses an innovative approach to the problem of persistent task performance in a continuous non-stop manner, for example monitoring by a fleet of drones with several battery replacements or recharging stations called *homes*. Our previous work [5] introduced the problem of persistent monitoring with one home. In this paper we define the problem of Multiple hOmes miNimal Spare drOnes for persistent mOnitoriNg (`MONSOON`), in [6] we introduced an extened abstract of a perliminary version. The drones are given a set of locations for persistent task performance, and it is necessary to guarantee that each of these locations is occupied by some drone at any given time. As the drones suffer from energy limitations and are thus required to fly to some home before their battery drains, it is necessary to add spare drones to the system that will replace the drones and maintain the continuous task performance. To face the drones' energy limitations efficiently and enable drones to replace/recharge batteries at the replacement stations, it is important to identify the minimum number of spare drones necessary to accomplish the persistent monitoring tasks. Therefore our goal is to determine the minimal number of spare drones, as well as finding a schedule of drone replacements that guarantees both that the persistent monitoring tasks are fulfilled indefinitely and that no drone battery is drained. The drones are homogeneous. See an illustration of our simulated environment in Figure 1. We consider two variations of the `MONSOON` problem: (i) An offline version, where the set of locations is given in advance; and (ii) An online version, where the set of locations is given one by one over time. When a location is given, it must be assigned immediately to one of the spare drones, which are added as needed.

While the paper describes the `MONSOON` in the context of persistent monitoring by drones, it is valid for *any* robot type which has energy constraints, performing *any* persistent task, as long as the travel cost of the robots in the environment

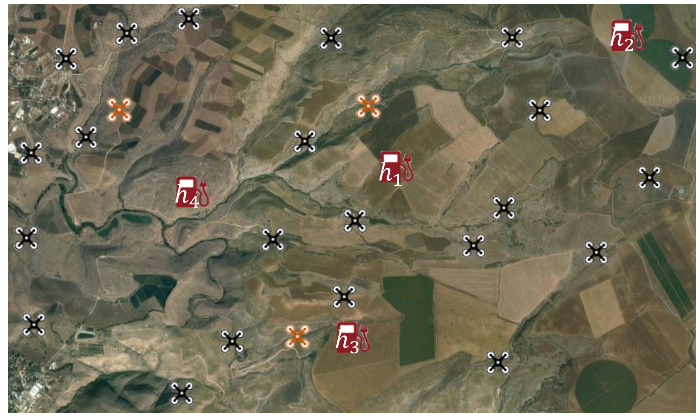

Fig. 1: Several locations requiring persistent monitoring, 4 charging/refueling stations (h_1,... ,h_4) and 3 spare drones (orange).

[1]Erez Hartuv, Noa Agmon and Sarit Kraus are with the Department of Computer Science, Bar-Ilan University, Isreal `erez.hartuv@gmail.com`, `{agmon, sarit}@cs.biu.ac.il`

978-1-7281-4277-7/20/$31.00 ©2020 IEEE

satisfies the triangle inequality.

We consider the problem of deciding whether one spare drone is enough, and prove that in contrast to the single home MSDPM, the multiple home case MONSOON is NP-Hard even for one spare drone, which means that the multiple homes problem is much harder than the single home problem. Moreover, giving an exact solution (intractable) for the multiple homes problem is also much harder since it incorporates the generalized traveling salesman problem, also known as the "traveling politician problem" [7], with Bin Packing [8]. They are interconnected by influencing each other's optimization.

To cope with this harder problem, we introduce a twofold innovative approach: (i) Proving that a simple drone replacement in which the replaced drone always returns directly to one of the homes is not worse than any other replacement procedure with respect to minimizing the number of spare drones. This result is most important since it allows better optimization methods by reducing the need to cope with $O(m^2)$ pairwise travel costs of the given m locations to only $O(m)$ travel costs between the m locations and the homes; and (ii) The latter result also enables us to give a novel approach for both minimizing the number of spare drones and scheduling them by using a Voronoi tessellation that breaks the MONSOON problem into MSDPM sub-problems corresponding to the Voronoi cells. Each sub-problem has an exact (intractable) solution using the Bin Maximum Item Double Packing (BMIDP), which is a bin packing variant where items are packed in each bin as done by regular bin packing with the only exception being that the maximum item of each bin is double packed [5]. A first fit variant greedy approximation algorithm, in which the maximum item of each bin is double packed, is used to solve the BMIDP problem. We show, experimentally, approximation factors of 1.7 and 1.6 for the online and offline cases, respectively.

Another outcome of the simple drone replacement result (i) is the ability to efficiently optimize the home placement by using facility location analysis approximation [9], [10].

II. Related Work

There are many works on task performance that take energy restriction into consideration [11], varying from theoretical analysis related to the vehicle-routing problem [12] to practical deployment of multi-robot teams in continuous missions with energy-aware solutions, e.g., [13], [14], [15]. Recently [5] were the first to consider minimizing the number of drones required when providing a persistent continuous (non-stop) service, but with only *one* recharging station (home). They were followed by [16] who consider minimizing the number of drones with *multiple* recharging stations. Our approach is completely different and significantly outperforms the work of [16] as discussed in the experimental results section.

The feasibility of swapping a group of small multi-rotor unmanned aerial systems during flight for persistent perimeter surveillance was studied in [17]. The thesis presented a system prototype that consisted of multiple quad-copters that were programmed to fly in different predefined patterns over a specified area. The number of quad-copters that fly simultaneously is determined by the need to monitor the area within the perimeter. Once an airborne quad-copter has depleted its battery life to the predetermined level, one of the standby quad-copters is activated as a replacement. The number of stand-by quad-copters at the launch site (home) is determined by the duration of the recharging. Although [17] treats a persistent task performed by several drones in the air, they all monitor near each other and use one home. It *does not* examine the question of several distant locations for monitoring, nor do they consider multiple homes as examined in this paper.

A Coverage Path Planning (CPP) problem is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles [18]. Such missions consist of five phases: takeoff, cruise, hovering, turning and landing. Taking energy consumption into consideration is necessary for their success. In order to correctly estimate the energy requirement in the five phases, a new route-based optimization model with column generation that can trace the amount of energy required for all different mission phases is presented in [19]. It uses numerical simulations to study the effectiveness of the proposed method for both a single UAV and multiple UAV scenarios for CPP problems.

An important persistent task is carried out by a group of UAVs that serve as the mobile Base Stations (BSs) [20], [21]. In order to fulfill such a task, there is a need to design a control solution for the UAV's navigation to fly around a target area in order to provide long-term communication coverage for the ground mobile users, taking energy limitations into consideration. Liu et. al [20] proposed a decentralized deep reinforcement learning (DRL)-based framework to control each UAV in a distributed manner. Their approach is to maximize the temporal average coverage score achieved by all UAVs in a task, maximize the geographical fairness of all considered points-of-interest (PoIs), and minimize the total energy consumption while keeping the UAVs connected and not flying out of the area's borders.

The Vehicle Routing Problem (VRP) [22], [23], [24] seeks to generate routes for a team of agents leaving a starting location referred to as the depot, which visit a number of goal locations and return back to the depot. Many works have presented algorithms for the VRP, also aiming to minimize energy consumption. For example, in [25], in addition to the attempt to minimize cost by minimizing overall traveling distance, the model also incorporates energy minimization. However, their goal is to satisfy the latest requirements of green logistics and not, as in our case, facing strong energy constraints that are needed for fulfilling the tasks.

While in our work the drones are required to occupy a set of m locations in order to perform a persistent task, Troudi et al. [26] examine a different problem: managing a large fleet of drones which are ready to deliver parcels to costumers. They present a VRP model to solve the sizing of a fleet of drones for urban parcel delivery logistics, taking into account the issues of autonomy and energy consumption related to the drone's technical specification. One of the proposed policies tries to make a compromise between the distance the drones will fly and the number of drones involved. They focus on policies in which the battery of a drone should be charged to 100% of its capacity for every mission.

Another related problem is the Continuous Monitoring Problem with the goal of maximizing the visiting frequency of targets, taking fuel constraints into consideration. Papers that consider this problem such as [14], [27], [28], [4] *do not* examine the question of minimal necessary UAVs, nor do they consider multiple (static) targets for monitoring, as examined in this paper.

Burdakov et al. [29] consider the problem of replacing security UAVs performing the task of surveillance along a perimeter and focus on the optimal replacement problem. They also *do not* examine the question of minimal necessary UAVs, rather they assume that the number of UAVs is given.

III. The MONSOON Problem Definition

In this section we formally define the Multiple hOmes miNimal Spare drOnes for persistent mOnitoriNg problem, MONSOON. Given a set L of m locations, $L = \{l_1, l_2, \ldots, l_m\}$, at least one drone should be present at each location at all times. Drones must be replaced and return, *before* their battery drains, to one of the homes in the set H of n homes, $H = \{h_1, h_2, \ldots, h_n\}$, for recharging or battery exchange. Therefore, the required number of drones necessary to ensure persistent monitoring is greater than m. We refer to the p extra drones, that is, the drones used for replacing the drones in the monitoring task, as *spare drones*. All drones are assumed to be homogeneous, have the same velocity v, and each one of them is initially fully charged q, and located in one of the n homes. A drone must be at one of the homes to exchange its battery. We assume that battery exchange/refuel time is negligible in order to simplify the calculations, although our results are also valid when it is not negligible. Following standard assumptions [29], a drone's batteries' charge decreases linearly with time. c denotes the rate of discharge per time unit.

Definition 1: Multiple hOmes miNimal Spare drOnes persistent mOnitoriNg-MONSOON problem. Given a set L of m locations that require persistent monitoring, a set of $m+p$ homogeneous drones, $p>0$, with velocity v, maximal battery capacity q, discharge rate c, and a set H of n home locations in which the drones replace batteries. Determine whether a schedule of drone replacements using p spare drones exists and find one, such that each location is monitored indefinitely by at least one drone, and no drone's battery will drain unless it is in one of the homes.

The goal is to find a schedule of drone replacements with a minimal number of spare drones (minimal number p) satisfying the persistent monitoring task, that is, guaranteeing both that the persistent monitoring tasks are fulfilled indefinitely and that no drone battery is drained. While referring to a drone that is located at l_i, we use d_i or just i ($1 \leq i \leq m$) interchangeably, and sd_i ($1 \leq i \leq p$) will be used for denoting a spare drone. We use $dist(a, b)$ to denote the distance of

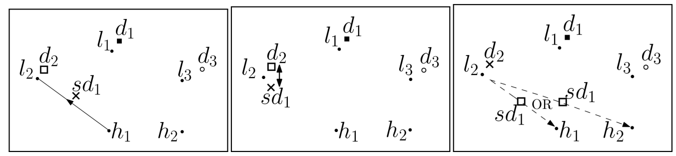

Fig. 2: $\boldsymbol{A_{oo'}}$**:** (a) sd_1 travels from home h_o o=1 to location l_2. (b) When it arrives at l_2, drones d_2 and sd_1 exchange roles and names. (c) The former drone d_2 (which is now sd_1) travels from l_2 directly to home $h_{o'}$, where o'=1 or o'=2.

a path between two points, a and b. In our analysis the paths between points are not necessarily straight lines, as long as the travel cost satisfies the triangle inequality. A drone's travel cost (number of charge units) between location l_i and location l_j is $c \cdot \dfrac{dist(l_i, l_j)}{v}$, and between home h_j and location l_i is $c \cdot \dfrac{dist(l_i, h_j)}{v}$. The travel cost between a location l_i and its nearest home plays an important role in the analysis (sections VI, VII) and is denoted $c \cdot \underline{t}_i$, where $\underline{t}_i = \dfrac{dist(l_i, nearest\ home)}{v}$.

Each drone i must remain at location l_i until it is replaced by one of the spare drones sd_j. When the replaced drone leaves l_i, it exchanges names with the spare drone that has just arrived: the spare drone becomes d_i and the replaced drone becomes sd_j (figures 2-5). Thus, we do not refer to a specific drone by its identity, rather by the task it performs. That is, the location in which it performs its monitoring task. This approach enables us to use a formalism ("replacement schemes"), presented in the next section, to mathematically denote all drone replacement patterns.

A drone replacement pattern is a series of drone replacements. We will also use the term *switch* for a single drone replacement which is an exchange of two drones. Thus a drone replacement pattern is a series of *switches*. In order to maintain persistent task performance, all of the switches must be made by a spare drone replacing a task drone, thus in each location there is a task drone at all times and the task is persistently executed. The following observation is the basis of the formalism: any *switch* (two drones which perform an exchange) must be one of exactly 4 types, which describe all possible drone actions regarding persistent task performance energetic optimal replacements at location l_i. We denote these 4 switch types by: $\boldsymbol{A_{oo'}}, \boldsymbol{B_o}, \boldsymbol{C}, \boldsymbol{D_{o'}}$. The subscripts o, o' of switch types ***A,B,D*** encode the recharging homes. Figures 2-5 illustrate an example for each switch type with three locations l_1, l_2, l_3, one spare drone sd_1 and two homes h_1, h_2. In the following section (Theorem 4.1) we prove that, in order to minimize the number of spare drones, it is enough to focus only on switches of type $\boldsymbol{A_{oo'}}$, where each drone that is replaced by a spare drone sd_j that arrived from some home h_o should return directly to some home $h_{o'}$ in order to replace (or recharge/refuel) its battery (o may or may not be equal to o').

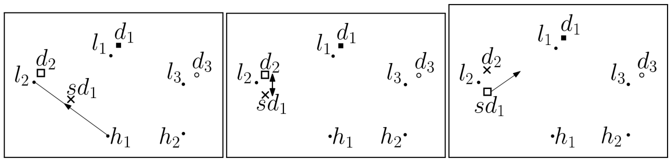

Fig. 3: $\boldsymbol{B_o}$: (a) sd_1 travels from h_o, where o=1 to l_2. (b) When it arrives at l_2, drones d_2 and sd_1 exchange roles and names. (c) The former drone d_2 (which is now sd_1) travels to location l_1.

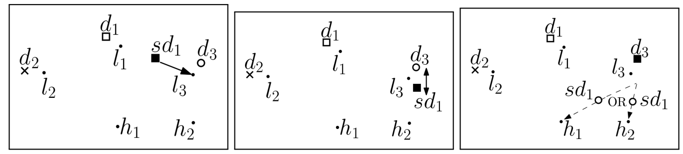

Fig. 5: $\boldsymbol{D_{o'}}$: (a) sd_1 travels from l_1 to l_3. (b) When it arrives at l_3, drones d_3 and sd_1 exchange roles and names. (c) The former drone d_3 (which is now sd_1) travels from l_3 to home $h_{o'}$, where o'=1 or o'=2.

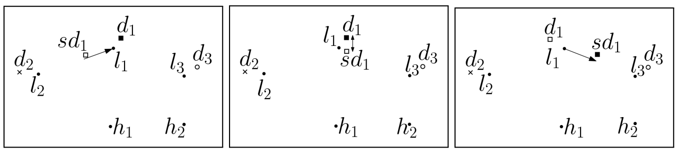

Fig. 4: $\boldsymbol{C}$: (a) sd_1 travels from l_2 to l_1. (b) When it arrives at l_1, drones d_1 and sd_1 exchange roles and names. (c) The former drone d_1 (which is now sd_1) travels to location l_3.

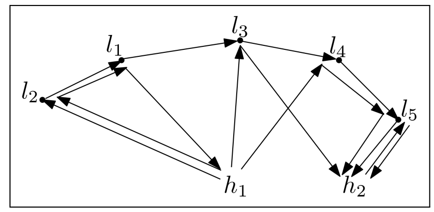

Fig. 6: An illustration of a replacement scheme.

IV. Replacement Schemes

In order to encompass all possibilities of drone replacement patterns, we define the mathematical notation of a replacement scheme for multi-homes.

Definition 2: Replacement scheme $R = (i_1, i_2, \ldots, i_{r_1})$ is a series of time consecutive drone switches until all of the batteries of all m drones are replaced at the homes in H. A drone exchange over location l_{i_j} of switch type s_j at time t_j is denoted by $i_j \in \{1, 2, \ldots, m\}$ for $1 \le j \le r_1$, $\quad r_1 \ge m$. The series of replacement timings, denoted by $R^T = (t_1, t_2, \ldots, t_{r_1})$, is non-decreasing. The series of replacement switch types is: $R^S = (s_1, s_2, \ldots, s_{r_1})$, with switch types $s_j \in \{\boldsymbol{A_{oo'}}, \boldsymbol{B_o}, \boldsymbol{C}, \boldsymbol{D_{o'}}\} \quad 1 \le o, o' \le n$.

R=(2, 1, 3, 4, 5, 2, 1, 3, 4, 5, 5) is the replacement scheme illustrated in Figure 6, and R^S=$(B_1, C, C, C, D_2, B_1, D_1, A_{12}, B_1, D_2, A_{22})$ is the corresponding series of replacement switch types. R^T=(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110) can be the corresponding series of replacement timings, if all travel times equal 10 time units.

Definition 3: Proper replacement scheme $R = (i_1, i_2, \ldots, i_m)$ is a replacement scheme in which drones only go back and forth to homes (switch type $A_{oo'}$).

$R = (2, 1, 3, 4, 5)$, R^S=$(A_{12}, A_{11}, A_{11}, A_{12}, A_{22})$ is the proper replacement scheme illustrated in Figure 7. The length of a proper replacement scheme is m, and it is a permutation of $(1, 2, \ldots, m)$.

A *cycle* is a sub-series of a replacement scheme that starts with a spare drone which leaves some home h_o in order to replace a drone which, in turn, may replace another drone and so on until the last replaced drone in the sub-series returns to a home h'_o.

Definition 4: Drone replacement cycle (*cycle* in short) is a sub-series of a replacement scheme $R = (i_1, i_2, \ldots, i_{r_1})$ and can be either a simple cycle, which is a drone replacement of switch type $\boldsymbol{A_{oo'}}$, or a compound cycle, which has drone replacements starting with switch type $\boldsymbol{B_o}$ followed by at least one switch type $\boldsymbol{C}$ followed by an ending switch type $\boldsymbol{D_{o'}}$.

A simple cycle over location l_x is denoted by C_{xx}, and a compound cycle over a set of locations l_v, l_w, l_x, l_y, l_z is denoted by C_{vwxyz}.

Note that any replacement scheme, $R = (i_1, i_2, \ldots, i_{r_1})$, has a corresponding series of drone replacement cycles: $R^C = (C_1, C_2, \ldots, C_a)$, where $m \le a \le r_1$. When $a = m$ there is no redundancy, i.e., there are exactly m (number of locations) battery replacements at the homes in H during R. If we have several spare drones, the concatenation of all cycles gives R up to order of elements due to the simultaneous independent performance of cycles by several spare drones. With one spare drone, the elements of a cycle are consecutive in R, the concatenation of all cycles gives exactly R.

If the minimum number of spare drones p is achieved by some series of drone replacements which solves the `MONSOON` problem, then it can be done with a proper replacement scheme which uses the same minimum number of spare drones p. The following theorem proves that.

Theorem 4.1: If p spare drones are needed in order to maintain persistent monitoring on location set L, using the set of homes H and the set of $m + p$ drones with battery charge capacity q, and performing any series of drone replacements, then it can also be achieved by a concatenation of proper replacement schemes.

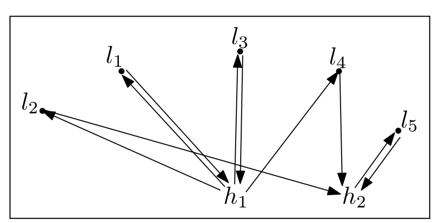

Fig. 7: An illustration of a proper replacement scheme.

Proof: Given any series of drone replacements, we notice that it is a concatenation of replacement schemes: $R_1 = (i_1, i_2, \ldots, i_{r_1})$, $R_2 = (i_{r_1+1}, i_{r_1+2}, \ldots, i_{r_1+r_2})$, $R_3, R_4 \ldots$ etc. R_1 has a corresponding series of drone replacement cycles (def 4): R_1^C=$(C_1, C_2, \ldots, C_m)$. The total number of cycles is m (no redundancies). R_1 ends with drone replacement i_{r_1}, i.e. at time t_{r_1} the last drone that has not yet replaced its battery arrives at one of the homes $h_{o'} \in h$ after leaving location $l_{i_{r_1}}$. This last drone was replaced at $l_{i_{r_1}}$ by a spare drone which came during the last cycle. If it is a simple cycle of switch type $\boldsymbol{A_{oo'}}$ then the spare drone came directly from one of the homes $h_o \in H$. If it is not simple, i.e. it consists of switch types $B_o, C, D_{o'}$, then the spare drone came from some location l_z if the cycle is $C_{xyzi_{r_1}}$, for example. We now carefully examine the situation at the end of R_1. If we sort the locations according to the time of entry from the homes in h of the replacement spare drones, we get: $l_{j_1}, l_{j_2}, \ldots, l_{j_m}$ (a permutation of the locations $l_1, \ldots, l_m$). Over l_{j_1} is the drone with the earliest entry time: it entered from some home (w.l.o.g) $h_1 \in h$ during the first cycle, but not necessarily to its final location l_{j_1}. Similarly, during the second cycle, the drone entered which eventually arrived at l_{j_2} during R_1, and so on. The drone over l_{j_1} had to travel along a path from h_1 to l_{j_1}, for example: $P_{h_1xyzj_1}$ with edges $\{(h_1, x), (x, y), (y, z), (z, j_1)\}$ that are scattered among the cycles of R_1. Although different cycles can be performed in parallel, those edges of $P_{h_1xyzj_1}$ cannot be performed in parallel because it is the same drone performing them. After performing $P_{h_1xyzj_1}$ alternately, this same drone has to wait the sequential performance of the suffix of the cycle containing the last edge path (z, j_1) of $P_{h_1xyzj_1}$. This suffix starts with the replaced drone leaving l_{j_1} and ends with some drone arriving w.l.o.g. at $h_2 \in h$, denoted $P_{j_1uvwh_2}$. Combining these two paths, $P_{h_1xyzj_1}$ and $P_{j_1uvwh_2}$, we get a cycle $\widetilde{C}_{xyzj_1uvw}$, therefore, by the triangle inequality and by applying the same arguments to drones over locations $l_{j_2}, \ldots, l_{j_m}$, we know that battery charge q, which is sufficient for R_1, is also sufficient for the proper replacement scheme defined as follows: $R_1^{'} = (j_1, j_2, \ldots, j_m)$. Repeating the above process on all other replacement schemes $R_2, R_3, \ldots$ completes the proof by transforming the given series of drone replacements into a concatenation of proper replacement schemes with the same number p of spare drones. ■

An immediate consequence of Theorem 4.1 is that solving the MONSOON problem requires using only the $O(m)$ travel costs between the m locations and the homes instead of all pairwise $O(m^2)$ travel costs. Another consequence is that we can optimize the locations of homes to minimize travel costs by using facility location methods. Although in this paper the homes are part of the input, we prove this theorem and leave the usage of it for optimizing home placement to future work. The usage of this theorem is important for example in dynamic scenarios where locations to be monitored are changing or when the drones are not homogeneous and the travel cost varies between different types of drones.

Theorem 4.2: Based on Theorem 4.1 we can use the Multi-Source Weber problem [10] to find optimal places for the homes of the MONSOON problem.

Proof: An immediate consequence of Theorem 4.1 is that solving the MONSOON problem depends only on the travel costs between the m locations and the homes because the minimal number of spare drones can be achieved by a proper replacement scheme in which drones only follow paths between homes and locations but not between locations or between homes. Minimizing the travel cost between homes and locations therefore yields better minimization of spare drones. The multi-source Weber problem in the plane requires finding locations for facilities in the plane to provide service to a set of m given demand points, each with an associated weight $w_i > 0$. Each demand point gets its service from the facility closest to it. The objective is to minimize the total sum of weighted minimum distances to the facilities. Let $d_i(X_j)$ be the Euclidean distance between demand point i and facility j located at $X_j = (x_j, y_j)$. The vector of unknown locations is $X = (X_1, \ldots, X_n)$, thus the objective function to be minimized is:

$$F(X) = \sum_{i=1}^{m} w_i \min_{1 \le j \le n} d_i(X_j)$$

Let the facility vector X represent the n homes, the demand points represent the m locations requiring persistent monitoring, $w_i = \frac{c}{v}$ for all m locations (drones are homogeneous), and $d_i(X_j) = dist(l_i, h_j)$, thus $w_i d_i(X_j)$ is the drone travel cost between h_j and l_i. Optimizing $F(X)$ yields the required optimal homes placement for the MONSOON problem. ■

V. MONSOON IS NP-HARD

Theorem 5.1: MONSOON is NP-Hard even for a one spare drone problem.

Proof: We shall show a reduction from the Traveling Salesman Problem (TSP) to MONSOON.

TSP instance [30]: given a finite set $C = \{c_1, c_2, \ldots, c_m\}$ of cities, a distance $dist(c_i, c_j) \in Z^+$ between each pair of cities $c_i, c_j \in C$, and a bound $B \in Z^+$ (Z^+ denotes the positive integers). Determine whether there is a tour of all the cities in C having a total length of no more than B. Given an instance of TSP, the corresponding instance of MONSOON has a set of homes $h = \{h_1, h_2, \ldots, h_m\}$ that represents the set of cities C. For any two homes $h_i, h_j \in h$ the inter-home distance is defined to be the same as the inter-city distance $dist(h_i, h_j)$=$dist(c_i, c_j)$. The set of locations $l = \{l_1, l_2, \ldots, l_m\}$ (one location for each home) such that for all $i = 1, 2, \ldots, m$: $dist(l_i, h_i) := \epsilon \cdot \frac{1}{2m} \cdot \min_{1 \le i,j \le m} dist(h_i, h_j)$, where $\epsilon < \frac{1}{1000}$ (each location l_i is very close to its attached home h_i and very far from other homes and locations). The battery capacity q equals B. c=1, v=1, therefore $dist(l_i, h_i)$ is the number of charge units it takes a drone to get from h_i to l_i (or vice versa). After creating the (obviously polynomial) MONSOON instance as above, let MONSOON determine whether a schedule of drone replacements using one spare drone exists and find one, such that each location is monitored indefinitely by at least one drone, and that no drone's battery will drain unless it is in one of the homes. According to Theorem 4.1, if q is enough to maintain persistent monitoring with one spare drone for the given MONSOON instance, then it can also be achieved by a series of drone replacements which has only switch type A_{xy}. Thus an optimal tour of drone replacements goes through a permutation of the homes $h_{\pi(1)}, h_{\pi(2)}, \ldots, h_{\pi(m)}$ with the replacement scheme: $R = (\pi(1), \pi(2), \ldots, \pi(m))$ $R^S = (A_{\pi(1)\pi(2)}, A_{\pi(2)\pi(3)}, \ldots, A_{\pi(m)\pi(1)})$ and continues indefinitely by repeatedly performing R. $\widetilde{L}$ denotes the total number of charge units needed to complete one tour of R.

$$\widetilde{L} = \sum_{i=1}^{m-1} \left[dist(h_{\pi(i)}, l_{\pi(i)}) + dist(l_{\pi(i)}, h_{\pi(i+1)}\right] +$$

$$+dist(h_{\pi(m)}, l_{\pi(m)}) + dist(l_{\pi(m)}, h_{\pi(1)}) =$$

$$= m\cdot\epsilon\cdot\frac{1}{2m}\cdot \min_{1\leq i,j\leq m} dist(h_i, h_j) + \sum_{i=1}^{m-1} \left[dist(l_{\pi(i)}, h_{\pi(i+1)})\right] +$$

$$+dist(l_{\pi(m)}, h_{\pi(1)}) \leq \frac{\epsilon}{2} \cdot \min_{1\leq i,j\leq m} dist(h_i, h_j) +$$

$$+m \cdot \epsilon \cdot \frac{1}{2m} \cdot \min_{1\leq i,j\leq m} dist(h_i, h_j) +$$

$$\sum_{i=1}^{m-1} \left[dist(h_{\pi(i)}, h_{\pi(i+1)})\right] + dist(h_{\pi(m)}, h_{\pi(1)})$$

$$\lim_{\epsilon\to 0} \widetilde{L} = \sum_{i=1}^{m-1} dist(h_{\pi(i)}, h_{\pi(i+1)}) + dist(h_{\pi(m)}, h_{\pi(1)}) =$$

$$= \sum_{i=1}^{m-1} dist(c_{\pi(i)}, c_{\pi(i+1)}) + dist(c_{\pi(m)}, c_{\pi(1)})$$

Therefore q is enough for the MONSOON to be true with one spare drone (and find a schedule) $\iff$ there is a tour (and find one) of all the cities in C having a total length of no more than B. ■

VI. Approximation Algorithm

As discussed in Theorem 4.2, an immediate consequence of Theorem 4.1 is that solving the MONSOON problem depends only on the travel costs between the m locations and the homes but not between locations or between homes. Therefore minimizing the travel cost between homes and locations yields better minimization of spare drones. This is the motivation to suggest that spare drones would arrive from the home closest to each location. Therefore, given a MONSOON instance, we apply Voronoi tessellation in order to break the MONSOON problem into MSDPM sub-problems with one home. For each home h_j a Voronoi cell is defined to be the set $h_j^{'}$ of all locations l_i which are closer to h_j than to any other home. Each Voronoi cell corresponds to a sub-problem. Each sub-problem has an exact (intractable) solution by applying BMIDP (Bin Maximum Item Double Packing which is a Bin Packing variant). The items to be packed are the travel costs between locations and the single home of the Voronoi cell sub-problem (double packing of maximum item to accommodate the need of the first replacement drone to wait for all others and replace again). The resulting solution is equivalent to finding the minimal number of spare drones p (the number of bins) and a scheduling of replacements (the packing). A First Fit (FF) variant greedy approximation algorithm (MIDFF), in which the maximum item of each bin is double packed, is used to approximate the solution. To each Voronoi cell we apply the MIDFF algorithm in order to efficiently approximate the BMIDP solution. The sum of the approximated number of spare drones over all Voronoi cells approximates the minimal total number of spare drones needed to solve the given MONSOON instance with several homes, as shown in Algorithm 1. The same arguments and

---

Algorithm 1: Solve MONSOON using MIDFF.

Initialize: $SpareDrones \leftarrow 0$
for `Each home of MONSOON` $h_i = h_1, h_2, ..., h_n$ **do**
    create INSTANCE of MSDPM with one home h_i
    INSTANCE locations:$\{l_j | l_j \in$ Voronoi cell of $h_i\}$
    SpareDrones=SpareDrones+MIDFF(INSTANCE)
end for

---

algorithm also apply to the offline MONSOON using MIDFFD instead of MIDFF. MIDFFD is the corresponding First Fit Decreasing (FFD) variant greedy approximation algorithm in which the maximum item of each bin is double packed; it is used to efficiently approximate the BMIDP solution for the offline MONSOON.

VII. Experimental Results

We conducted extensive experiments with various parameter settings in order to find an estimation of the approximation factor of both online MIDFF and offline MIDFFD. Note that battery units are used for distance measurement. While searching for the estimation for MIDFF and MIDFFD approximation factors, we also checked the influence of the parameters on it and on the number of spare drones. We wanted to compare the algorithm's estimation results with the actual optimal solution, OPT. However, in order to avoid intractable computation of the bin packing variant (BMIDP), we used the minimal number of spare drones = $\frac{\sum_{\forall i} 2c\cdot\underline{t}_i}{q-\min_{\forall i}\{2c\cdot\underline{t}_i\}} < OPT$, instead of OPT ($2c\cdot\underline{t}_i$ is the number of charge units it takes a drone to travel back and forth between l_i and its nearest home).

Therefore, the experimental approximation factor is a strict upper bound of the real approximation factor. Thus dividing by a value lower than OPT gives a higher approximation factor, which means that 1.6 (offline) and 1.7 (online) are upper bounds, so our algorithm gives approximation factors that are not worse than 1.6 and 1.7 but may be much better. Moreover we show asymptotic convergence of the approximation factors and we report the asymptotic worst case values of 1.6 and 1.7 (while we get better values in Figures 8 and 10 of 1.4 and 1.5 respectively). In all cases shown in Figures 8 - 10, we iterate 1000 times and report the MEAN values with error-bars of ± 1 STD. In each iteration we generate uniformly random samples of m location distances in the range (min dist, max dist). For both versions of the BMIDP problem, those are the item sizes to be packed in bins that represents spare drones. For the online version, the item ordering is kept random as in the random sample, simulating locations that are given one by one over time with no constraint on the order as in the MONSOON online version problem. For the offline version, items are sorted by decreasing order, simulating a set of locations given in advance as in the MONSOON offline version problem. We compute the number of spare drones needed and the approximation factor, and report the average over the 1000 iterations. Each of the figures presents a unique setup of three parameters as constants, and presents the influence of the fourth parameter as it changes on the approximation factor and on the number of spare drones. In all of the graphs,

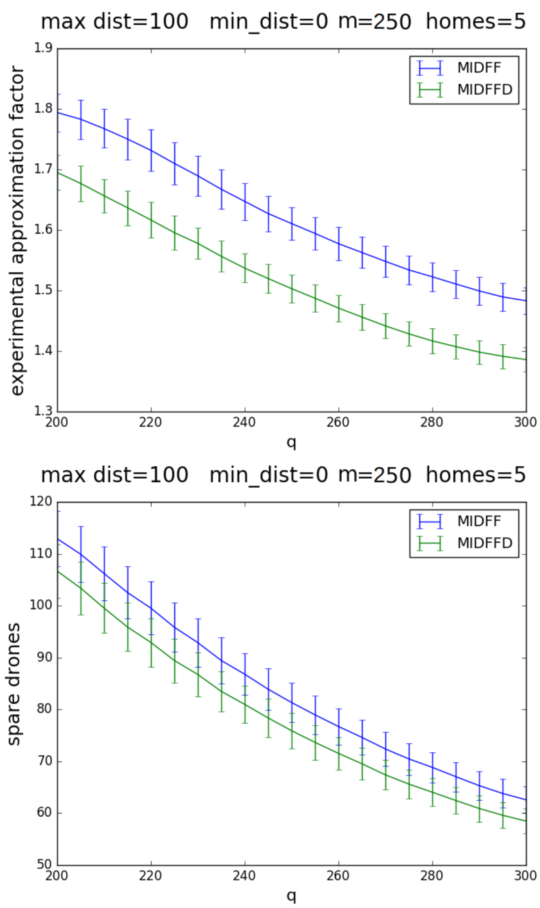

Fig. 8: Influence of increasing battery capacity q.

the upper line is the MIDFF results and the lower (better) one is the MIDFFD. Figure 8 describes the influence of battery capacity q on the approximation factor and the number of spare drones. The parameters setup: (i) m - 50 locations $\times$ 5 homes; (ii) Maximum distance - 100; (iii) Minimum distance - 0 (iv) q - varies between 200 and 300. As q grows we need fewer spare drones, because more locations can be allocated to each spare drone. The approximation factor also gets better with increased capacity because packing gets easier as more efficient packing opportunities are available, thus less penalty of greedy sub-optimal decisions. Figure 9

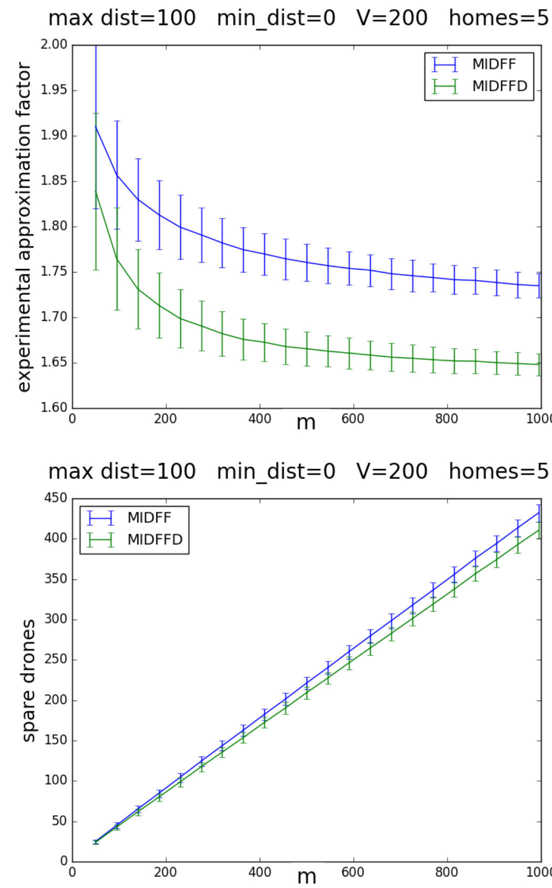

Fig. 9: Influence of increasing m, the number of locations.

reports the influence of m, the size of the location set L on the approximation factor and the number of spare drones. Parameters setup: (i) m - varies from 10 locations $\times$ 5 homes to 200 locations $\times$ 5 homes; (ii) Maximum distance - 100; (iii) Minimum distance - 0; (iv) q - 200. As m grows we need more spare drones because we must visit more locations with the same energy capacity q. The approximation factor gets better as m increases, because of the degrading influence of spare drone free residual space which is left after packing in each Voronoi cell and is not exploited to pack in other Voronoi cells. There is an asymptotic limit which evolves from the real approximation factor of the one home MIDFF and MIDFFD. Figure 10 describes the influence of the minimum distance on the approximation factor and the number of spare drones. Parameters setup: (i) m - 50 $\times$ 5 homes; (ii) Maximum distance - 100; (iii) Minimum distance - varies from 0 to 50; (iv) q - 200. As the minimum distance grows, the random samples of m locations' distances in the range (min dist, max dist) have higher distance values, therefore the items to pack are bigger. We need more spare drones, since each drone can visit fewer locations that are more distant. The approximation factor gets better with an increased minimum distance, because the items become more homogeneous as the samples are from a smaller range and we get more uniform item sizes.

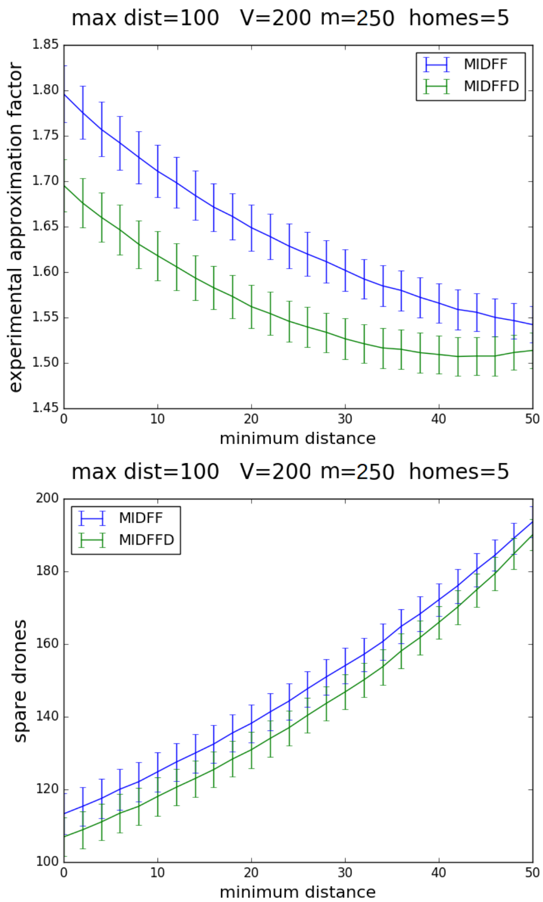

Fig. 10: Influence of increasing the minimum distance from h_i.

Finally, as mentioned in the related work section, we emphasize that our approach significantly outperforms the completely different approach reported in [16]. They can only deal with small-sized problems, as they report that when the problem size becomes large, large variations occur. Therefore they gave numerical results of up to 40 locations to be monitored, while we can deal with thousands of locations to be monitored (with stable results) as seen in Figures 8-10. Moreover, their solution yields much more than twice the spare drones compared to our solution, for example with 40 locations and 12 charging stations (4 central stations + 8 refueling stations), 53 spare drones are needed, while in our solution with 50 locations and 5 refueling stations (much fewer refueling stations, which makes the spare drones' optimization harder), only 24 spare drones are needed.

VIII. Conclusions

In this paper we have examined the problem of determining the minimal number of drones necessary for performing continuous tasks in m locations, considering the energy limitations of the drones and finding the associated schedule for drone replacements. We have proven (Theorem 5.1) that if there are multiple possible recharging stations, the problem is NP-Hard even in its simplest version: determining whether a schedule of drone replacements using *one* spare drone exists and find one, guaranteeing that all tasks are carried out indefinitely. We show (Theorem 4.1) that a simple back-and-forth replacement pattern of the drones (between the task locations and the recharging stations) is optimal with respect to finding the minimal number of spare drones. Combining this with a variant of the Bin Packing problem, we suggest a heuristic algorithm that is shown empirically to result in an approximation ratio of 1.6 and 1.7 for the offline and online versions of the problem, respectively. Directions to pursue in the future are: (i) Heterogeneous drones; (ii) Dynamic scenarios where both monitored locations and recharging stations are added and deleted over time; (iii) Devising an exact solution (intractable) in order to find theoretical bounds on the approximation factor; and (iv) Using Theorem 4.2 to find optimal placement for charging stations in all above mentioned future directions.
Main contributions of this paper:

- Proving Theorem 4.1 is the base of paper (simple replacement pattern is optimal).
- Providing an approximation algorithm to minimize the number of spare drones and schedule drones' replacements, both online and offline.
- Proving Theorem 4.2 to find optimal home (recharging station) locations.
- Our algorithm significantly outperforms published research results for the same problem [16] (see section VII).

References

[1] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, B. Rinner, *et al.*, "Networked uavs as aerial sensor network for disaster management applications," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[2] K. Kanistras, G. Martins, M. J. Rutherford, K. P. Valavanis, *et al.*, "Survey of unmanned aerial vehicles (uavs) for traffic monitoring," in *Handbook of unmanned aerial vehicles*. Springer, 2015, pp. 2643–2666.

[3] D. D. Willens, C. Proudlove, T. Miller, *et al.*, "Drone away: Bombs, photographs, pizzas, hearts, and headaches," *The Brief*, vol. 46, no. 4, 2017.

[4] 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. of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 221–232, 2014.

[5] E. Hartuv, N. Agmon, and S. Kraus, "Scheduling spare drones for persistent task performance under energy constraints," in *AAMAS*, 2018, pp. 532–540.

[6] E. Hartuv, N. Agmon, and S. Kraus, "Scheduling spare drones for persistent task performance with several replacement stations - extended abstract," in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2019, pp. 95–97.

[7] G. Laporte, H. Mercure, and Y. Nobert, "Generalized travelling salesman problem through n sets of nodes: the asymmetrical case," *Discrete Applied Mathematics*, vol. 18, no. 2, pp. 185–197, 1987.

[8] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: A survey," in *Approximation algorithms for NP-hard problems*, 1996, pp. 46–93.

[9] K. Jain and V. V. Vazirani, "Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation," *Journal of the ACM (JACM)*, vol. 48, no. 2, pp. 274–296, 2001.

[10] Z. Drezner, J. Brimberg, N. Mladenović, and S. Salhi, "New local searches for solving the multi-source weber problem," *Annals of Operations Research*, vol. 246, no. 1-2, pp. 181–203, 2016.

[11] S. K. Debnath, R. Omar, and N. B. A. Latip, "A review on energy efficient path planning algorithms for unmanned air vehicles," in *Computational Science and Technology*. Springer, 2019, pp. 523–532.

[12] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European journal of operational research*, vol. 59, no. 3, pp. 345–358, 1992.

[13] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Transactions on robotics*, vol. 22, no. 3, pp. 507–522, 2006.

[14] V. Mersheeva and G. Friedrich, "Multi-uav monitoring with priorities and limited energy resources." in *ICAPS*, 2015, pp. 347–356.

[15] T. H. Nguyen, F. M. Delle Fave, D. Kar, A. S. Lakshminarayanan, A. Yadav, M. Tambe, N. Agmon, A. J. Plumptre, M. Driciru, F. Wanyama, *et al.*, "Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games," in *International Conference on Decision and Game Theory for Security*. Springer, 2015, pp. 170–191.

[16] H. Park and J. R. Morrison, "System design and resource analysis for persistent robotic presence with multiple refueling stations," in *ICUAS*. IEEE, 2019, pp. 622–629.

[17] C. P. J. Teo, "Persistent perimeter surveillance using multiple swapping multi-rotor uas," Ph.D. dissertation, Monterey, CA; Naval Postgraduate School, 2018.

[18] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[19] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, "Energy-constrained multi-uav coverage path planning for an aerial imagery mission using column generation," *Journal of Intelligent & Robotic Systems*, pp. 1–15, 2019.

[20] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-uav navigation for long-term communication coverage by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, 2019.

[21] H. Park and J. R. Morrison, "On the resources required to provide persistent robotic service: Multiple immobile customers and a single service station," in *APIEMS'14*, 2014, pp. 1–8.

[22] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, vol. 99, pp. 300–313, 2016.

[23] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1472–1483, 2009.

[24] J. R. Montoya-Torres, J. L. Franco, S. N. Isaza, H. F. Jiménez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Computers & Industrial Engineering*, vol. 79, pp. 115–129, 2015.

[25] S. F. Ghannadpour and A. Zarrabi, "Multi-objective heterogeneous vehicle routing and scheduling problem with energy minimizing," *Swarm and evolutionary computation*, vol. 44, pp. 728–747, 2019.

[26] A. Troudi, S.-A. Addouche, S. Dellagi, and A. Mhamedi, "Sizing of the drone delivery fleet considering energy autonomy," *Sustainability*, vol. 10, no. 9, p. 3344, 2018.

[27] D. Mitchell, M. Corah, N. Chakraborty, K. Sycara, and N. Michael, "Multi-robot long-term persistent coverage with fuel constrained robots," in *ICRA*, 2015, pp. 1093–1099.

[28] 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," *Journal of Intelligent & Robotic Systems*, pp. 1–13, 2013.

[29] O. Burdakov, J. Kvarnström, and P. Doherty, "Optimal scheduling for replacing perimeter guarding unmanned aerial vehicles," *Annals of Operations Research*, vol. 249, no. 1-2, pp. 163–174, 2017.

[30] M. R. Garey and D. S. Johnson, "Computers and intractibility," 1979.