Trajectory Optimization for Fast Sensor Energy Replenishment using UAVs as RF sources

Igor Dias da Silva, Yann Busnel, Christelle Caillouet

Trajectory Optimization for Fast Sensor Energy Replenishment using UAVs as RF sources

Igor Dias Da Silva
Université Côte d'Azur, France
Inria, CNRS, I3S

Yann Busnel
IMT Atlantique, France
IRISA

Christelle Caillouet
Université Côte d'Azur, France
CNRS, I3S, Inria

***Abstract*—The problem of the lifetime of connected objects, in most use cases (Industrial Internet of Things (IIoT), disaster management, etc.) is an essential element of the proposed solutions. Radio frequency (RF) harvesting of sensor batteries is an attractive solution, however, it does not scale up if it has to be done by human operators, and becomes impossible if the objects are located in unreachable places. An innovative solution consists of using fleets of drones to take care of this regular recharge. In this paper, we focus on the self-organised deployment of a fleet of drones to solve this problem, taking into account the multiple constraints involved. We propose a two-step optimization framework based on an optimal orchestration solution to reduce the recharging time of a complete sensor system, by optimizing the number of drones, the overall flight time and their energy consumption. We illustrate the performance of our framework that ensures the drones avoid conflicts to guarantee a higher energy harvesting efficiency (establishment of optimal drone positions and planning of the global flight plan).**

I. Introduction

The application of sensor networks has grown significantly in recent years. The democratisation of the Internet of Things (IoT) is in fast expansion, in the fields of industry, military, personal (with home automation), ecology (with the surveillance of natural areas), precision agriculture, disaster management (following natural disasters or wars), *etc*. The majority of the aforementioned use cases require the implementation of sensor systems over long periods of time. Therefore, optimising the lifetime of each sensor, and consequently the system itself, is a key element in the deployment of these wireless systems [1].

While electronics manufacturers have made great strides in optimising the power consumption of wireless sensor batteries, the need for regular recharging is paramount to ensure the longevity of the systems and their usability. However, the increasing scale of these systems (both in terms of the size of the monitored area and the growing number of sensors) means that it is no longer possible for human operators to carry out these recharges manually. Therefore, many energy harvesting solutions have been proposed recently, such as, for example, the addition of a local production unit to each piece of equipment (solar panel, micro wind turbine, etc.). However, these solutions are often bulky and costly to implement. Furthermore, they are not constant in production, dependent on climatic conditions, and are not applicable in closed areas (such as large warehouses or assembly lines in the case of Industrial IoT – IIoT – for example).

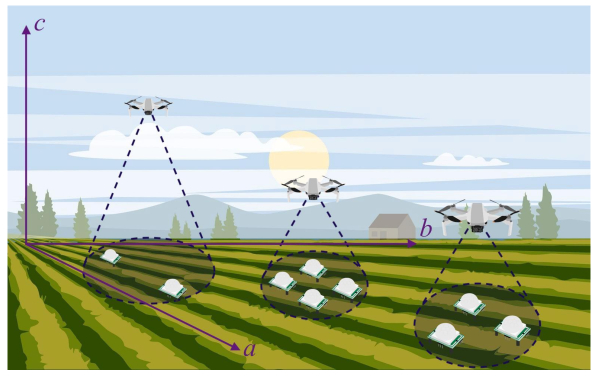


Fig. 1: An example employing 9 nodes and 3 drone positions.

We are therefore focusing on a wireless Radio Frequency (RF) power harvesting solution. The unquestionable advantages of these solutions are flexibility, adaptability and automation. Ones then use mobile vehicles to visit the sensors regularly to recharge them and collect their data. An ideal candidate for this type of recharging are fleets of drones, or Unmanned Aerial Vehicles (UAVs), which are capable of transmitting RF signals to the sensors to recharge their batteries. The objective is to define automatic and autonomous flight plans for these fleet of drones to power up energy-constrained sensor nodes (*cf.* Figure 1). The main energy consumption of UAVs being their engines, it is essential to optimise their flight time, in order to maximise the amount of energy transmitted to the sensors on the one hand and to minimise the time required to recharge the entire system on the other.

In this work we propose a two-step optimization framework to determine the UAVs trajectories to plan the fast sensor energy replenishment. Given the sensors positions, their harvesting needs, and a number of available UAVs, we first derive a Mixed Integer Linear Program (MILP) to optimise both the placement of the UAVs and the associated hovering times in order to minimise the overall charging time (excluding moving flight time). Then, in a second step, we propose a greedy algorithm to schedule the flight plans, integrating spatiotemporal constraints and positioning conflicts, leading to less efficient loads (*e.g.*, two UAVs cannot load the same sensor at the same time as efficiently as if they did it successively [2]). The output is a set of UAV conflict-free trajectories fulfilling the positions computed by the MILP, while minimizing the overall duration for recharging the whole system.

We take into account realistic assumptions in terms of *(i)* energy harvesting model for both transmitters and receivers obtained from a RF-power harvesting module manufacturer, *(ii)* 3D UAV moves and positioning to adjust the altitude and line of sight situations to the harvesting power, and *(iii)* sensor capabilities of being charged by only one drone at a time.

The rest of this paper is organized as follows. Section II briefly reports the most recent related work. In Section III we describe our solution. First, we introduce the Energy Harvesting Model in Section III-A, then we present the optimization model for UAV positioning in Section III-B before proposing a greedy algorithm for scheduling conflict-free trajectories of the drones in Section III-C. We then evaluate the performance of our solutions and we discuss their efficiency in Section IV before concluding the paper in Section V.

II. Related work

Recent works have studied the usage of radio-frequency (RF) signals for energy harvesting systems [3]. The use of UAVs as chargers to enhance wireless sensor network lifetime has already gained attention lately. But many works limit their study to deploying only one drone in the sensor network [4], [5], [6], [7], [8], which is not scalable for large sensor networks and wide coverage areas. In such case, the drone, which is also power limited, has to decompose its trajectory into several short time rounds to provide enough energy to all the sensors.

Optimal deployment of UAVs with wireless power capabilities has also been studied in the literature. The study can be limited in a two-dimensional space [9] which decreases the accessibility and the recharge opportunities. Besides, some works focus on one-to-one recharge model in which one drone is dedicated to one sensor [10], or to a cluster-head of a group of nodes [11], [6], [12]. The scenarios require as many drones as sensors or energy harvesting relaying capabilities, and do not take advantage of the drones mobility and high-power omni-directional energy emission.

Globally optimizing the UAV 3D trajectory for wireless power transfer network has been investigated in [8]. Authors show that, to maximize the energy harvested considering the UAV's 3D placement and charging time, the problem has to be decomposed into sub-problems solved separately. However, they limit their study to the case of one drone and 2D positioning, then adjusting the altitude in a second step.

In [13], an optimal linear model determines the set of positions and their associated duration time to ensure a given amount of energy to each sensor provided by a set of mobile UAVs. Authors considered that one drone can harvest energy to multiple sensor nodes that are within its coverage area, and that the amount of energy harvested by a node corresponds to the sum of powers received by all the covering drones. [2] observe that the received power depends on the number of RF sources and channel conditions. And they show, contrarily to [13], that the amount of energy harvested by a node does not equal the sum of harvesting powers received by all the available sources due to spatial diversity. It is more reasonable to limit the number of simultaneous RF sources to 1 to ensure a good behaviour of the energy harvesting capabilities for the sensor nodes. But this assumption enforces to be careful in the trajectory design of multiple UAVs. It is necessary to avoid flying collisions as well as conflict for the RF transmissions. Deriving a complete model computing the UAV 3D trajectories that are conflict-free in terms of avoiding simultaneous RF sources for each sensor is therefore of critical importance.

III. Two-step Optimization Framework

Given a square monitoring area with a set S of sensors and their given 2D locations, we seek to optimally deploy in the 3D-space a set of drones U to provide the required amount of energy to the sensors in a minimum amount of time. Each sensor $s \in S$ has required energy needs of E_s corresponding to an empirical value that has to be sufficient to power-up a node, to allow it to take a measurement, and transmit the data to the drone (or to a base station). We focus here on the recharge time and the flying time, and we neglect the transmission time.

A. Energy Harvesting Model

Each sensor node is equipped with an RF-power harvesting module capable of converting the RF-power from transmitted signals to DC power. We use UAVs as chargers with a directional antenna facing the ground to continuously emit power to the network and recharge the sensors. A drone can adjust its altitude to recharge multiple sensors at the same time, whereas each sensor cannot be charged by more than one drone simultaneously.

The efficiency of the energy harvesting capabilities is affected by several parameters, such as environmental conditions and the distance between the source (charger on the drone) and the sensor. The amount of power harvested by a sensor s when it is located within the line of sight of the charger u is [14]:

$$P_{h_s} = P_{\text{rx}}^{d_{us}} f^{d_{us}}, \tag{1}$$

where $P_{\text{rx}}^{d_{us}}$ is the received power and $f^{d_{us}}$ is the efficiency of the harvesting antenna at distance d_{us}. The received power at distance d is given by the following propagation model [15]:

$$P_{\text{rx}}^{d} = P_0 \frac{e^{2\sigma G}}{d^{2b}}, \tag{2}$$

where $\frac{e^{2\sigma G}}{d^{2b}}$ has a log-normal distribution with a shadowing coefficient σ ($G \sim N(0,1)$) and b is the amplitude loss exponent. P_0 is the received power at reference distance.

The minimum harvested power received by a sensor to recharge its battery depends on the efficiency of the converter for the corresponding received power at this particular distance. We denote by Γ this minimum harvesting threshold, which is a hardware-depended constant. Formula (3) defines the harvested energy of a sensor s for a given time period t:

$$H_s^t = \int_0^t P_{h_s} t dt. \tag{3}$$

This energy is stored directly in a super-capacitor with some leakage properties expressed as ηH_s^t, where $\eta \in (0, 1)$.

B. Optimal positioning : MILP approach

The first step of our framework seeks to determine the optimal number of drones needed to recharge the sensors in the minimum amount of time. We determine for each UAV the list of positions and their associated hovering time at each location. To do so, we discretize the monitored area and derive a set P of possible 3D positions. Given the energy needs E_s of each sensor $s \in S$ and the minimum harvesting threshold Γ defined in the previous section, we associate to each sensor s a set of possible positions P_s from which Γ is reached. We define the following binary variables for UAV positioning :

- $x_u = 1$ if the drone $u \in U$ is used, 0 otherwise.
- $y_p^u = 1$ if the drone u is deployed at position $p \in P$, 0 otherwise.

To each position, we then want to compute the drone hovering time to recharge enough battery power to the sensor. We define a set t_p^u of continuous variables representing the time duration of UAV u at position p. We also set τ the maximum total recharge time of a drone that practically represents a lower bound of its maximum flying time. We want to minimize the number of used drones to recharge the sensors in a minimum amount of time to ensure the required harvested energy E_s. However, using less drones forces them to go to several positions in order to recharge all the sensors, and minimizing the number of used drones is an orthogonal objective of minimizing the total recharge time. Indeed, having enough drones brings us deploy one drone for one sensor, lasting enough time to recharge all E_s entirely. On the contrary, limiting the number of positions per drone therefore splits the harvested time between all the available UAVs, and they all go to the same positions but for a limited duration such that they will all charge a fraction of amount of E_s and exchange positions. These two antagonistic objectives make us develop a bi-objective optimization problem in which we want to limit the total recharge time period, and the number of positions :

1) $\min \max_{u \in U} \sum_{p \in P} t_p^u$
2) $\min \max_{u \in U} \sum_{p \in P} y_p^u$

(1) seeks to minimize the time needed to harvest and store the required quantity of energy from the deployed drones to the sensors, and (2) minimizes the number of visited positions. As said, these two objectives are antagonistic since visiting more positions allow to split the recharge time between the UAVs and so increase the total recharge time, while limiting the recharge time tends to make the drone stay at only one position during the longest duration to reach E_s for all the reaching sensors. So we want to study the balance between the number of visited locations and the total flying time of the UAVs in order to derive satisfying solutions.

Since we want to compute $\min \max$, we define 2 continuous variables modeling the worst time λ_1, and worst number of positions λ_2. We can combine the bi-objective into only one function $\min a \cdot \lambda_1 + (1-a) \cdot \lambda_2$, where a is an input parameter to balance the impact of the two objective functions. Then,

$$\sum_{p \in P} t_p^u \leq \lambda_1, \forall u \in U \tag{4}$$

$$\sum_{p \in P} y_p^u \leq \lambda_2, \forall u \in U \tag{5}$$

We ensure that the number of used drones do not exceed the budget B (in terms of number of available drones $\leq |U|$):

$$\sum_{u \in U} x_u \leq B \tag{6}$$

The harvested energy of each sensor s must be fulfilled :

$$\sum_{u \in U} \sum_{p \in P_i} (1-\eta) H_s^{t_p^u} \geq E_s, \forall s \in S \tag{7}$$

Then, we must verify the possibility of a drone to move around. We can place each drone at different locations during the operation, but the total length of its flying time cannot exceed the imposed time limit (i.e., τ) (8). If a drone is not used, then it cannot be placed in any position ((9) and (10)). And if the drone is not located in a given position, then the associated time duration should be 0 (11).

$$\sum_{p \in P} t_p^u \leq \tau x_p, \forall u \in U \tag{8}$$

$$t_p^u \leq \tau y_p^u, \forall u \in U, p \in P \tag{9}$$

$$x_u \leq \sum_{p \in P} y_p^u, \forall u \in U, p \in P \tag{10}$$

$$y_p^u \leq x_u, \forall u \in U, p \in P \tag{11}$$

To avoid collision and interference we do not place drones above other drones. So we limit the total time for positions in the same z-axis to λ_1. This ensures that the drones that need to use the same z-axis have enough time to visit it consecutively:

$$\sum_{u \in U} t_p^u + \sum_{p' \in P, (x_p, y_p) = (x_{p'}, y_{p'})} \left(\sum_{u' \in U} t_{p'}^{u'} \right) \leq \lambda_1, \forall p \in P \tag{12}$$

Finally, in order to derive a feasible schedule for the UAV trajectories, we limit the total time spent on the set of charging positions P_s of sensor s to the longest flying time of a drone.

$$\sum_{u \in U} \sum_{p \in P_i} t_p^u \leq \lambda_1, \forall s \in S \tag{13}$$

C. Trajectory optimization : The Wait Time algorithm

The output of the previous linear model gives us, for each UAV, the set of positions and the associated hovering time at each position to optimally harvest energy to sensors. The goal here is to construct near-optimal UAV trajectories going through the selected positions, and integrating spatio-temporal constraints and positioning conflicts. More specifically, the output of the previous linear model is a set of tasks, where a task k is given by a drone u_k, a position p_k, a duration t_k in seconds and a set of sensors S_k that are charged during the task's execution. Each drone can have multiple tasks to complete, and our goal here is to determine the order in which the drones complete their respective tasks without conflicts.

There is a *conflict* between two tasks k and k^* if they use the same drone ($u_k = u_{k^*}$), if they use the same position ($p_k = p_{k^*}$), or if they charge some common sensors at the same time ($S_k \cap S_{k^*} \neq \emptyset$) (to avoid the multi-charger constraints [2]). A task is said to be *conflict-free* at a given moment if it has no conflict with any of the tasks being currently executed.

Even if we ignored the possible conflicts between tasks, determining the fastest order of tasks for each drone is the same as solving the Traveling Salesman Problem (TSP) that is known to be NP-hard. We therefore present a greedy algorithm that assign the tasks in order to minimise the total time the drones have to wait hovering to avoid conflicts. More precisely, the algorithm runs as follows :

1) We order the tasks in increasing order of the wait time, that is, how long the UAV have to wait until it can execute the task (including the time of flight to reach p_k). If the task k is conflict-free, then the wait time corresponds to the time of flight (ToF), that is, the necessary time for drone u_k to reach p_k from its previous position. However, if the task k is in conflict with some task k^* being currently executed, then its wait time is $max(ToF, f_{k^*})$, where f_{k^*} is equals to the remaining duration before k^* ending. In some cases, when k^* uses the same drone as k, the wait time is $f_{k^*} + ToF$, as the drone u_k will only be able to move when k^* is finished.
2) Once the tasks are ordered, if a task is conflict-free, we assign it immediately. This changes the wait time of the remaining tasks, and they must be reordered. If a task k is not conflict-free and u_k is not executing any task, then u_k moves to p_k in advance so that it can execute k as soon as it becomes conflict-free.

An illustration of the Wait Time algorithm is presented in Figure 2. The linear program has computed 12 tasks assigned to 4 drones to cover 40 sensors. The lines represent the time a drone spends flying, and the rectangles represent the period of execution of the tasks when the drones send RF signals to recharge the sensor batteries.

All drones start at the same location corresponding to a base station located at position $(0, 0)$. The tasks are ordered following the Wait Time algorithm. Task 4 has the shortest wait time (at the beginning the wait time is simply the time of flight), so it is the first one to be assigned. Now we must reorder the remaining tasks because their wait time may have changed if they have a conflict with Task 4. For example, the wait time for Task 7 is no longer its time of flight, but f_4, the time remaining for Task 4 to finish. Tasks 0 and 11 once again have the same wait time because they are both conflict-free and have the same time of flight. Once Task 0 is assigned, we reorder the tasks and, as the wait time for Task 11 remains the same and the smaller one, it is then assigned to drone 3.

The tasks are reordered once more, and, Task 8 is the one with the smallest wait time since it only needs to wait for Task 0 to finish. Task 2 also needs to wait for Task 0 to finish, but since they use the same drone, it can only travel after Task 0

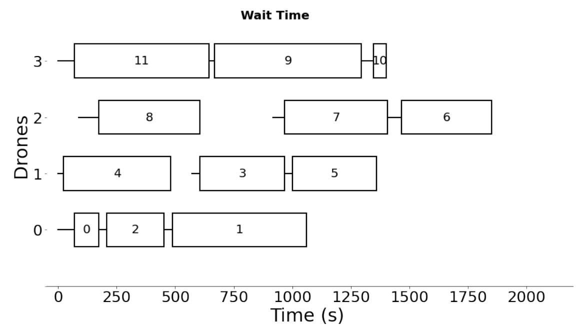


Fig. 2: Example using the Wait Time algorithm to schedule the tasks in a 5x5 grid with 40 sensors and 4 drones.

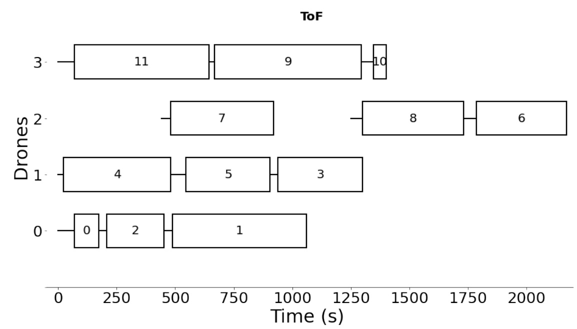


Fig. 3: Example using the ToF algorithm to schedule the tasks in a 5x5 grid with 40 sensors and 4 drones.

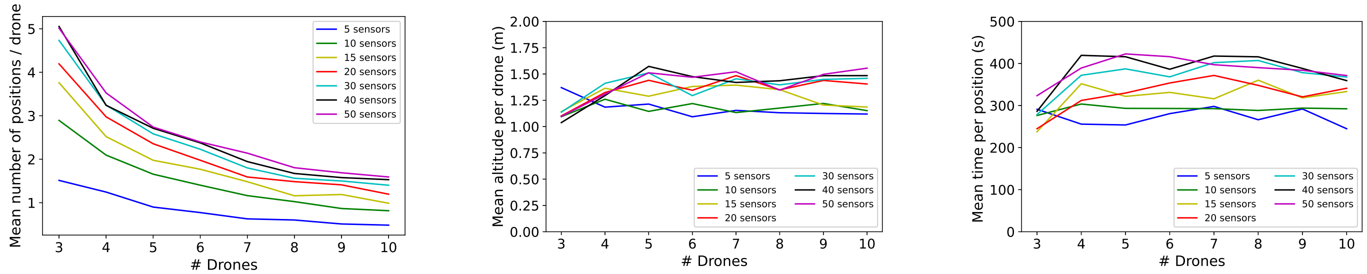
finishes, so its wait time is larger than Task 8. The execution of the algorithm continues consecutively, until the chronogram shown in Figure 2 is obtained.

IV. Performance evaluation

We consider a 2-dimensional area of size $50m \times 50m$ where sensors are randomly placed. We divide the area as a 5×5 grid with 5 possible altitudes, totaling 125 possible positions entirely covering the area to deploy the drones. The number of sensors varies between 5 and 50 while the number of drones available to charge them varies from 3 to 10. Each sensor requires 150 mJ and the energy model parameters are $P_0 = 10mW$, $b = 1.05$, $\sigma = 1$, and $\eta = 0.3$ [13]. The maximum recharging time of a drone is set to 1 hour and $\alpha = 0.1$. The MILP model is implemented in Java and solved using IBM CPLEX solver on an Intel Core i9-10900K CPU, 3.70GHz, 64 Gb RAM computer with Fedora system, release 35.

A. Analysis of the optimal positioning

The first step of our framework is the MILP determining the optimal positions that should be visited by the drones and for how long to recharge the battery of the sensors using RF-signals. As one can observe in Figure 4a, the more sensors to charge, the more positions the drones must visit. On the other hand, the more drones we have, the fewer positions they need to visit on average. The altitude of the UAV positioning is also an important factor. Indeed, the energy harvesting capability is strongly related to the distance between the RF emitting source and the destination. The closer in line of sight, the



(a) The mean number of positions each drone must visit.

(b) The mean altitude of the positions occupied by drones.

(c) The mean time each drone spends in the same position (tasks' mean duration).

Fig. 4: MILP results: Positioning analysis.

better. As depicted in Figure 4b, our MILP model seeks to place the drones as low as possible to maximize the energy harvested by the sensors in the shortest amount of time. So the lowest positions located at 1 meter are the most used because the recharging phase is faster. However, as we increase the number of sensors, either the model adds new drones flying at 1 meter, or it places some drones on higher positions to cover more sensors at once. This trade-off between coverage and recharging time is also illustrated in Figure 4c, depicting the average time a drone spends in the same position. With a higher density of sensors, each drone can charge more sensors at once, but it requires a higher altitude to provide more coverage, and consequently more time in each position.

B. Validation of the trajectory optimization algorithm

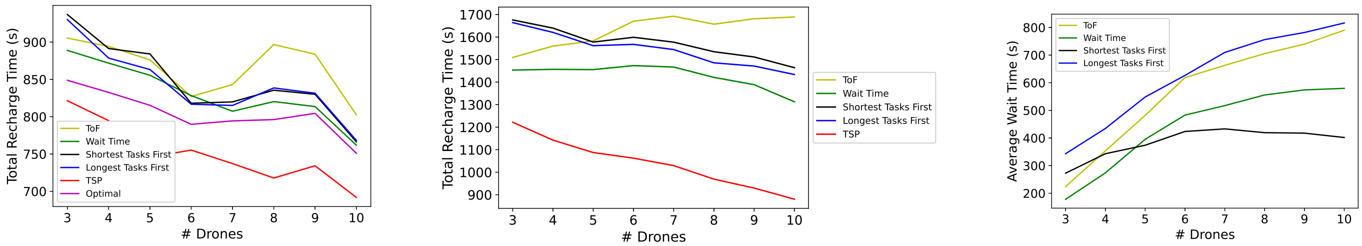
Given the results of the MILP, we then apply the Wait Time algorithm presented in Section III-C to schedule the trajectory (e.g. order of tasks) for the drones in order to minimise the total time of the energy replenishment problem (including time of flight and waiting time due to conflict avoidance).

We validate our approach in comparison with :

- **TSP:** Given the set of visited positions for each drone provided by the MILP, we compute the optimal TSP solution by brute-force. This solution gives us a lower bound for the global minimum time to recharge the sensors. However, this solution is not conflict-free, meaning that the computed trajectories can allow multiple simultaneous UAV chargers for a sensor, therefore limiting the efficiency of the harvesting system.
- **Optimal:** We consider all possible permutations of tasks and, for each permutation, we assign the tasks in order. Then we select the permutation that leads to the best result. For each task k in a given permutation, k is assigned to start as soon as possible, as done in the wait time algorithm. If the task k is in conflict with a current task, the UAV times its departure such that it arrives at p_k as soon as k is conflict-free. If the UAV u_k is executing a current task, then we wait until u_k is free to assign k. In the end, we select the scheduling with the minimal recharge time. When a task is assigned, we don't reorder the remaining tasks, as done in the wait time algorithm, because eventually all possible sequences are verified.
- **Time of Flight (ToF):** This greedy approach is similar to the Wait Time algorithm except that the tasks are ordered only by their time of flight in increasing order. We then assign the first task that has no conflict with the current ones: the drone leaves its current position immediately and starts executing the task as soon as it reaches the next position. If there is a conflict, we look for the next task with the shortest time of flight and repeat the process. If we find no task for a given UAV that is conflict-free, we choose the closest task and program the moving in advance so that, once this task becomes conflict-free, the UAV can immediately start executing the task.
- **Shortest/Longest Tasks First:** We order the tasks by their duration in ascending/decreasing order. For each task, if it is conflict-free, we assign it immediately. Otherwise, the drone moves at the last moment to start charging as soon as possible (*i.e.* when it becomes conflict-free).

Figure 3 presents the result of the ToF algorithm on the same scenario as Fig. 2. One can observe that Wait Time gives a better solution, because the Task 7 has a higher priority than Task 8 with the Wait Time, and can thus be scheduled before. Since then Tasks 3 and 8 are in conflict, the scheduling of Task 8 based on the ToF has to be delayed later, giving a longer solution for the trajectory of Drone 2.

Figure 5b compares the performance of the different algorithms. The figure illustrates how long it takes to charge all sensors by scheduling the tasks with each algorithm. As expected, we see that no greedy algorithm can charge the sensors as fast as the TSP because the TSP solution allows conflicts. But then, the Wait Time algorithm outperforms all the other naive greedy approaches. The previous example (Fig. 2 and 3) is confirmed here on all the 2800 tested scenarios for various number of drones and sensors. Out of the 2800 scenarios, we manage to solve the Optimal algorithm on 870 instances with less than 10 tasks because taking into consideration all possible tasks permutations consumes a lot of time (Fig. 5a). Considering the 3,628,800 possible permutations for 10 tasks can take 28 minutes in our experiments. In these 870 scenarios, the optimal algorithm charges all sensors in 803.92 seconds on average. The Wait Time algorithm takes on average only 3.28 % longer to charge the sensors than the optimal scheduling.



(a) Total recharge time: 870 scenarios with < 10 tasks and $5 \leq$ sensors ≤ 15.

(b) Total recharge time for all 2800 scenarios. Optimal cannot be depicted.

(c) Average time drones wait idle in each algorithm in all 2800 scenarios.

Fig. 5: Validation of our approach. Considering a 5x5 grid, with a number of sensors varying between 5 and 50.

The ToF algorithm takes 7.87 % longer on average, while the Longest Tasks First and Shortest Tasks First take 4.80% and 5.39% longer, respectively. In fact, we observed that all algorithms can reach the optimal solution in some instances. The Wait Time algorithm reaches the optimal solution in 63.9 % of the instances. The ToF algorithm reaches the optimal solution in 59.54% of the instances, while the Longest Tasks First and Shortest Tasks First reach the optimal solution in 51.14 % and 47.01 % of the cases.

So the Wait Time algorithm has the best performance in our tests even though, despite its name, it was not the smallest average wait time. Figure 5c illustrates that the drones remain more idle on average with the Wait Time algorithm than with the Shortest Tasks First algorithm. However, in the Shortest Tasks First there is usually one drone that pays the price and has to wait more than the others, because it has the largest tasks. Consequently, the largest task might have to wait for the second-largest task to finish at the end, which pushes the total recharge time further. The Wait Time algorithm, on the other hand, has a larger averaged wait time because small tasks sometimes need to wait for big tasks to complete, but it doesn't penalize the drones with the largest tasks. Hence, it provides the strongly better total recharge times on average.

Overall, increasing the number of drones allows recharging the sensors faster, however it also increases the amount of conflicts between tasks since we have more drones charging the same sensors. Reducing the amount of conflicts between tasks is important to allow the use of several drones simultaneously, and therefore, benefit even more from a larger UAV fleet.

V. Conclusion and Future Work

We presented a two-step framework for optimal energy replenishment of a set of sensors, using UAV as RF sources. We first solve a mixed integer linear program to determine the drones' placement in a 3D space such that it minimizes the time required to charge all ground sensors. Then, we proposed a Wait Time heuristic to compute quickly the best order in which the drones should visit the deployment positions. The algorithm ensures that the drones will not be in conflict at any moment and take advantage of the parallelism between tasks. Avoiding these conflicts, such as multiple drones charging the same sensors simultaneously, ensure a higher energy harvesting efficiency. In the future, we would like to take the number of conflicts between tasks as a metric to be minimized in the MILP itself, to push for more parallelism which would allow us to benefit even more from a higher number of drones.

References

[1] S. Rajendran and G. Nagarajan, "Network lifetime enhancement of wireless sensor networks using EFRP protocol," *Wireless Personal Communications*, vol. 123, p. 1769–1787, 2022.

[2] D. Altinel and G. Karabulut Kurt, "Energy harvesting from multiple rf sources in wireless fading channels," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 8854–8864, 2016.

[3] F. Sangoleye, N. Irtija, and E. E. Tsiropoulou, "Smart energy harvesting for internet of things networks," *Sensors*, vol. 21, no. 8, 2021.

[4] Z. Yang, W. Xu, and M. Shikh-Bahaei, "Energy efficient uav communication with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1913–1927, 2020.

[5] S. Arabi, E. Sabir, H. Elbiaze, and M. Sadik, "Data gathering and energy transfer dilemma in uav-assisted flying access network for iot," *Sensors*, vol. 18, no. 5, 2018.

[6] A. Saif, K. Dimyati, K. Noordin, D. G C, N. Shah, Q. Abdullah, and M. Mohamad, "An efficient energy harvesting and optimal clustering technique for sustainable postdisaster emergency communication systems," *IEEE Access*, vol. 9, pp. 78 188–78 202, 2021.

[7] Y. Xu, Z. Liu, C. Huang, and C. Yuen, "Robust resource allocation algorithm for energy-harvesting-based d2d communication underlaying uav-assisted networks," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 17 161–17 171, 2021.

[8] W. Feng, N. Zhao, S. Ao, J. Tang, X. Zhang, Y. Fu, D. K. C. So, and K.-K. Wong, "Joint 3d trajectory design and time allocation for uav-enabled wireless power transfer networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9265–9278, 2020.

[9] S. Zhang, Z. Qian, F. Kong, J. Wu, and S. Lu, "P3: Joint optimization of charger placement and power allocation for wireless power transfer," in *IEEE Infocom*, April 2015, pp. 2344–2352.

[10] J. C. Park, K.-M. Kang, and J. Choi, "Low-complexity algorithm for outage optimal resource allocation in energy harvesting-based uav identification networks," *IEEE Communications Letters*, vol. 25, no. 11, pp. 3639–3643, 2021.

[11] D.-T. Do, A.-T. Le, Y. Liu, and A. Jamalipour, "User grouping and energy harvesting in uav-noma system with af/df relaying," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 855–11 868, 2021.

[12] D. Zorbas, P. Raveneau, and Y. Ghamri-Doudane, "On optimal charger positioning in clustered rf-power harvesting wireless sensor networks," in *19th ACM MSWiM*, 2016, pp. 225–228.

[13] C. Caillouet, D. Zorbas, and T. Razafindralambo, "Optimal placement of drones for fast sensor energy replenishment using wireless power transfer," in *Wireless Days Conference*, 2019.

[14] D. Zorbas, P. Raveneau, and Y. Ghamri-Doudane, "Assessing the cost of rf-power harvesting nodes in wireless sensor networks," in *IEEE GLOBECOM*, Washington, DC, USA, Dec 2016.

[15] M. Z. Win, P. C. Pinto, and L. A. Shepp, "A mathematical theory of network interference and its applications," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 205–230, Feb 2009.