Computing Optimal Drone Positions to Wirelessly Recharge IoT Devices

Dimitrios Zorbas and Christos Douligeris
Department of Informatics, University of Piraeus, Greece
email: dzorbas@unipi.gr

Abstract

In an Internet of Things (IoT) environment such as the wireless sensor network the nodes are usually powered by batteries. When the nodes are deployed in a harsh environment the battery replacement may be a hard task or the cost of the maintenance may be very high. In this paper, we explore the possibility of recharging the nodes using drones equipped with chargers that can wirelessly transfer energy to the nodes. We examine the problem of minimizing the number of drone positions by adjusting the drones' altitude since the harvesting power is related to the altitude the drones fly. Indeed, the higher the altitude, the larger the observed area but the lower the harvesting power. The aim is to find the minimum possible drone locations such that all the nodes are charged before the drones run out of energy. We show that this problem can be reduced to the set-cover problem which is NP-Complete. We propose a time-efficient heuristic which can run on devices with low processing capabilities. We, also, solve the problem using Integer Linear Programming (ILP) capable of providing the optimal solution. The evaluation using a set of Monte Carlo simulations shows significant performance gains in terms of execution time compared to the ILP solution while keeping the number of positions close to the optimal.

1 Introduction

The IoT comprises some hundreds of interconnected wireless devices that are capable of sensing, controlling and communicating in order to give us a better understanding of our world. These tiny wireless nodes are often deployed in inaccessible places or in places where there is no permanent power supply. The use of batteries tackles the problem of energy demand, however, their batteries can only last for a few months or years depending on the application. The battery replacement may be a task which incurs a high cost and considerable amount of time especially in sparse large-scale deployments.

To alleviate the energy demands of the nodes, prolong the network lifetime, or in some cases provide energy autonomy, the use of energy harvesting techniques has been recently proposed [1]. Unlike other power harvesting methods (solar, vibration etc.), RF-power harvesting by injecting a constant energy rate, can recharge multiple devices at the same time, and can be used whenever and wherever there are energy needs. RF-power harvesting can be used to power low duty cycle IoT devices such as RFIDs, sensors, and actuators.

RF-power harvesting is divided in two categories; in the first category, the nodes take advantage of the ambient RF signals transmitted by nearby primary devices such as TV towers, cell network antennas, or nearby WiFi's. Ambient harvesting does not require any additional equipment other than the harvesting module, however, the amount of power to be harvested may be extremely low and it varies over time [2]. In the second category, dedicated chargers are used to continuously transmit energy and recharge the nodes. Using dedicated chargers, the received power at the nodes is significantly higher compared to the ambient harvesting but several chargers with high energy reserves needs to be placed across the network.

In this paper, we use drones that act as chargers which periodically fly over the network and fully recharge the nodes. The beam of the chargers is directional towards the ground. Thus, as depicted by Figure 1, the drones can fly at different altitudes to optimize the charging procedure. The higher a node flies, the less the received power, since the signal attenuates with the distance and the harvesting module becomes less efficient. On the other hand, the nodes may have different energy demands depending on their position and their role in the network. For example, nodes close to the sink need to stay active for a longer time to forward data from their neighbors. Thus, they consume much more energy compared to other nodes and require longer recharge times or shorter distances from the charging drones.

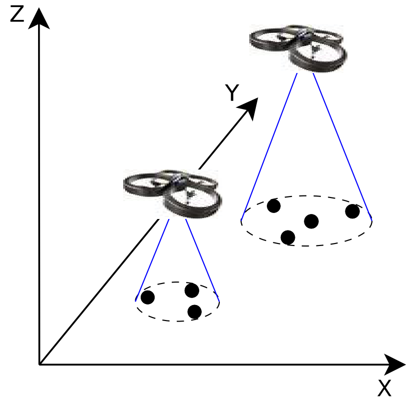


Figure 1: Two drone positions with different altitude charging a number of nodes.

Although the problem of node energy replenishment using vehicles and wireless power transfer has been already examined in the literature, this is the first work which involves drones that can fly at different altitudes to recharge multiple node at a time. Moreover, given the drone battery capacity, the positions of the nodes, their consumption, and other network parameters we present equations to describe whether wireless power transfer using drones is feasible. We introduce the Optimal Drone Positioning (ODP) problem to find optimal drone positions and minimize the nodes total energy consumption. We show that it can be reduced to the set-cover problem which is NP-Complete. To solve the ODP problem we propose an efficient heuristic with low complexity as well as an Integer Linear Programming approach which evaluates a larger range of eventual drone positions. Simulation results using realistic node and charging parameters confirm the feasibility of using drones in the energy replenishment process. Moreover, they show a lower computation cost of the heuristic approach while it generates solutions close to the optimal in terms of number of positions.

The rest of the paper is organized as follows. Section 2 is dedicated to the related work. In Section 3 we describe the energy consumption and harvesting models and we introduce the ODP problems. Section 4 describes the proposed solutions and Section 5 presents the comparison and evaluation results. Finally, Section 6 concludes the paper and presents directions for future work.

2 Related work

In this section, we briefly cite only works related to the energy replenishment problem with dedicated chargers for IoT devices with limited power resources. The reader can refer to [3] and [4] for other similar optimization problems with drones and sensors.

Several are related to the problem of finding the optimal number of chargers. Zhang et al. [5] examined the problem of maximizing an objective function subject to a power budget. They present approximation algorithms as well as greedy heuristics to find close to the optimal charger positions and their corresponding power transmission levels. The simulation results outperform a random placement strategy. The solution though is limited to 2-dimensional networks. A similar problem is studied by Dai et al. [6]. In this case, a charger positioning problem taking into account the electromagnetic radiation levels is considered. The authors propose a solution which alternates the transmission power of the chargers. The solution guarantees that the electromagnetic radiation levels are safe for every location on the plane. Optimal charger placement problems are also explored in [7] and [8]. The first one presents localized and centralized solutions to place mobile chargers in between a cluster of nodes. The chargers are used to extend the lifespan of as many nodes as possible. In the second work, the optimal number of chargers problem for battery-free nodes is examined. A fast heuristic algorithm is proposed and is compared to the optimal solution. These two last works consider only 2-dimensional networks and omni-directional energy emission.

Another set of studies deals with the problem of network traversing and node energy replenishment by mobile agents. Traversal strategy problems to efficiently recharge nodes using one or more vehicles are examined in [9]. The objective in this kind of problems is how to schedule and, sometimes, take decisions locally in order to periodically recharge a set of nodes. However, additional objectives may be defined, such as the minimization of the recharge operation time [10] and the data gathering [11]. These problems are often formulated as dynamic vehicle routing problems and can be solved to optimality by linear programming solutions. Distributed solutions are, also, proposed [12]. Similar routing vehicle problems and solutions combined with node recharging are also presented in [13] and [14]. The network may also involve mobile nodes [15]. This kind of problems differ from the one examined in the current paper since they are limited to the 2-dimensional space. Apparently, some of the approaches can be revised to involve vehicles in 3-dimensional space as well. These approaches can be combined with our drone altitude adjustment concept to compute minimum vehicle and energy efficient schedules.

Finally, a few studies involve drones in node recharging process although the objectives and the technology of the wireless power transfer are different. Detweiler et al. [16] present a proof-of-concept device to recharge nodes using magnetic resonant power transfer. This is a different technology which exhibits a high power transfer but it requires very short distances (i.e., up to some centimeters) between the transmitter and the receiver as well as accurate coil alignment between the two sides. Basha et al. [17] extend the previous work by considering a mobile sink scheme to further improve the network lifetime. Finally, Najeeb et el. [18] present a localized method using the same recharge technology. They consider a problem where the drone can recharge nodes in one-by-one manner without having a complete knowledge of the network parameters. As a consequence, the drone takes local decisions to visit the next node.

3 Problem description

3.1 Node consumption model

The energy a node consumes depends on the amount of time it remains active. We consider a modern node configuration where the nodes are parts of an IEEE 802.15.4-TSCH network [19] operating at 2.4GHz. IEEE 802.15.4-TSCH was chosen as the MAC layer protocol since it exhibits significant energy savings combined with a 99% packet delivery ratio compared to traditional CSMA MAC layers [20]. The routes towards the sink are constructed using the RPL protocol [21]. Each node chooses a preferred parent to forward its data according to the RPL functionality. In IEEE 802.15.4-TSCH, the time is divided in slotframes and each slotframe has 101 timeslots of 10ms each. During these 10ms a node can transmit (or receive) a packet and receive (or transmit) an acknowledgment. If a node does not transmit or receive during a timeslot, it remains in sleep mode, saving a significant amount of energy. Depending on the application, a node may send zero to multiple packets per slotframe.

In this paper, we consider networks consisting of N nodes. Each node i generates q_i packets per τ seconds. Thus, the total energy cost of a node in τ, C_i^τ, is computed as follows:

$$\begin{aligned} C_i^\tau &= Q_i 10^{-2} P_{tx} + (Q_i - q_i) 10^{-2} P_{rx} + \\ &+ (\tau - (2Q_i - q_i) 10^{-2}) P_{slp}, \end{aligned} \tag{1}$$

where $Q_i = \sum_{j=1}^{K_i} + q_i$. K_i is the number of predecessor nodes of i. P_{tx}, P_{rx}, and P_{slp} are the transmission, reception and sleep mode power consumptions, respectively.

We must note here that no exact knowledge of the node consumption is required. Since in IEEE 802.15.4-TSCH the nodes wake up at strict time intervals, the number of received packets at the sink can be used to approximate the energy consumption.

3.2 Energy harvesting model

We adopt an energy harvesting model where the harvesting energy depends on the distance between the node and the charger, the propagation model, the efficiency of the harvesting module, and the charging time. More specifically, the energy harvested by a node i in t time while it is in the harvesting range of a charger j is given by (2):

$$H_i^t = \int_0^t P_{rx}^{d_{ij}} f^{d_{ij}} t dt, \tag{2}$$

where $P_{rx}^{d_{ij}}$ is the received power and $f^{d_{ij}}$ is the efficiency of the harvesting antenna at distance d_{ij}[1]. The received power at distance d is given by the following propagation model [22]:

$$P_{rx}^d = P_0 \frac{e^{2\sigma G}}{(\frac{d}{\rho})^{2b}}, \tag{3}$$

where $e^{2\sigma G}$ has a log-normal distribution with a shadowing coefficient σ ($G \sim N(0,1)$). The term $1/(\frac{d}{\rho})^{2b}$ accounts for the far-field path loss with distance d, where the amplitude loss exponent b is environment-dependent. P_0 is the received power at reference distance ρ.

We, also, consider energy conversion losses as a function of the harvesting energy $l_i^t = \lambda H_i^t$, where λ is the power loss factor ($0 < \lambda < 1$).

[1]we use efficiency values provided by the manufacturer Powercast (http://www.powercastco.com)

3.3 Recharge feasibility

In order to achieve energy replenishment using drones, a number of conditions need to be satisfied taking into account the node energy consumption and harvesting models.

First, the final amount of energy after the losses in τ time units must be higher than the node consumption for the same period of time as it is described by Eq. (4):

$$(1-\lambda)H_i^{\tau} - C_i^{\tau} > 0. \tag{4}$$

Second, the recharge time of the nodes cannot exceed the maximum flying time of a drone. If we assume that the recharge process starts after $R\tau$ amount of time where $R \in \mathbb{Z}^*$ and M is the maximum flying time, the total time needed to replenish the energy consumed during this period must not exceed M:

$$\frac{RC_i^{\tau}}{(1-\lambda)H_i^{\tau} - C_i^{\tau}}\tau \leq M. \tag{5}$$

From Eq. (2), (3), and (5) we can compute the maximum distance, d_{max}, a drone can be found away from i in order to recharge it. d_{max} is given by Eq. (6).

$$d_{max} = \sqrt[2b]{\frac{(1-\lambda)P_0 e^{2\sigma G} f^{d_{max}} \tau M}{\tau R C_{\tau}^{i} + M C_{\tau}^{i}}}. \tag{6}$$

Figure 2 illustrates d_{max} using realistic drone and charging parameters as they are given in Section 5.1. Different packet generation intervals and maintenance times are evaluated considering 1 packet per τ minutes for a leaf node. We denote with θ the angle of the beam. The results show that d_{max} remains higher than 4 meters even for dense packet generations and sparse maintenance periods. However, if $d_{max} = 4$m the radius of the covering disk cannot exceed $4\tan\frac{\theta}{2} = 6.93$m limiting the feasibility of recharging multiple nodes with a single drone to dense networks only.

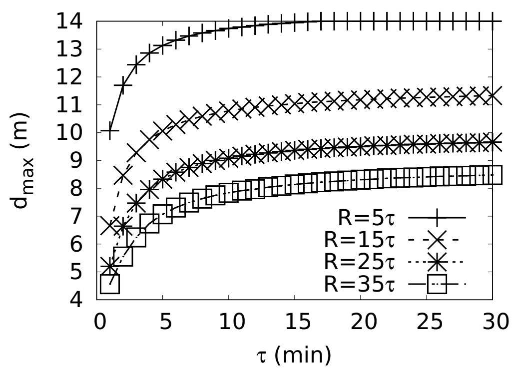


Figure 2: Maximum drone distance d_{max} for different packet intervals τ and maintenance starting times $R\tau$.

Moreover, there is a maximum distance beyond which no power conversion is possible. This distance may vary according to the amount of received power and the attenuation model. As a consequence, there is a maximum drone altitude denoted with h_{max}. We, also, set a lower drone altitude limit, h_{min}, for obstacles avoidance purposes.

A single drone may recharge multiple nodes at the same time. It can achieve this by adjusting its altitude h as it is depicted in Figure 3. It is assumed that the projection of the beam to the

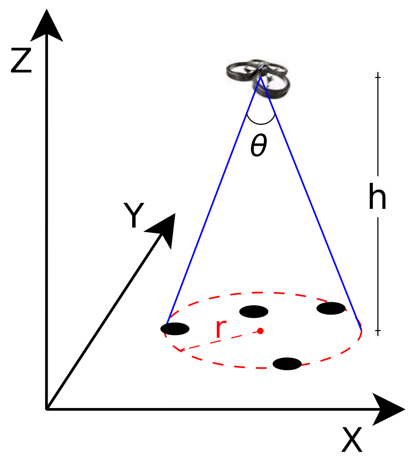


Figure 3: A drone with altitude h and covering radius r.

XY plane is a disk with radius r. Eq. (4) and (5) must be satisfied for each node in the disk, otherwise, the drone needs to lower its altitude with the cost of covering a smaller number of nodes.

3.4 The Optimal Drone Positioning problem

Given a set of nodes $S = \{s_1, s_2, ..., s_N\}$, we define the ODP problem as a minimization problem of the number of positions D so that the conditions described in Section 3.3 are satisfied. We omit the formal formulation of the problem due to the limited size of this paper.

Next, we will show that the ODP problem can be transformed to a set-cover problem which is NP-Complete. Let us assume the example of Figure 3. This set of four nodes can be covered by $2^4 - 1$ different ways using one to four drones. Excluding the subsets that do not satisfy the conditions of Section 3.3, the objective is transformed in finding the minimum number of subsets so that all the nodes are included in these subsets. This is actually the objective of the set-cover problem.

4 The proposed solutions

4.1 Set-cover solution

As it is shown in the previous section, the ODP problem can be easily solved once it is transformed to a set-cover problem. However, to do so, a high number of subsets (node combinations) needs to be generated and tested whether they fulfill the recharge requirements. The total computation cost of processing all the subsets may be extremely high since $2^N - 1$ subsets need to be checked and for each subset two procedures with considerable cost take place. First, the checks described by Eq. (4) and (5) require $\mathcal{O}(n)$ computations, where n is the cardinality of the subset. Moreover, if the subset has three or more elements a Smallest Enclosing Circle (SEC) computation is required to determine the altitude of the drone. A SEC exhibits an $\mathcal{O}(n)$ cost [23].

Next, we describe an efficient algorithm to decrease the number of generated subsets and, thus, the computation cost. The proposed algorithm does not solve the ODP problem but it is used as an efficient way to generate and test all the possible drone position combinations that

Algorithm 1: ODP to set-cover transformation

```
sets = ∅;
foreach i ∈ [1..N] do
    find max distance d_max using Eq. (6);
    reachable = {i};
    foreach j ∈ [1..N] do
        if |ij| ≤ r_max then
            reachable := reachable ∪ {j};
    foreach subset combination sub from reachable do
        skip sub if cardinality higher than 3 or if i ∉ sub;
        compute drone position x, y, h using SEC;
        if h < h_min then  h = h_min;
        check = 1;
        if h > h_max then  check = 0 ;
        foreach j ∈ sub do
            if j does not satisfy Eq. (4), (5) then
                check = 0;
        if check == 1 then
            foreach j ∈ reachable do
                if distance of j to (x, y) ≤ h tan(θ/2) then
                    if j satisfies Eq. (4), (5) then
                        sub := sub ∪ {j};
            sets := sets ∪ sub;
return sets;
```

will be later used as input for the set-cover solution. The algorithm (see Algorithm 1) takes advantage of the following SEC properties; (a) the minimum enclosing circle can be determined by at most three points which lie on the boundary of the circle and (b) if it is determined by only two points, then the line segment joining those two points must be a diameter of the SEC. This means that only the subsets with one, two or three elements need to be checked. The computed circle (i.e., disk) will also contain any other nodes covered by the examined drone.

The algorithm picks one node i per iteration from the N available nodes and it checks whether any other node is found in distance $r_{max} = d_{max} \sin(\theta/2)$ from it as it is described by Eq. (6). All these nodes are placed in set `reachable`. They are used to generate a set of subsets, each of them corresponding to potential drone positions capable of recharging i and other neighboring nodes. Each subset is checked whether it satisfies the criteria defined by Eq. (4) and (5) once the position of the drone is computed using a SEC algorithm. If yes, other nodes located in the computed smallest enclosing disk are also checked. If they satisfy the recharge criteria, they are also added in the examined subset. The algorithm terminates by returning the collection of subsets.

Algorithm 1 requires at most $N(\binom{N}{1} + \binom{N}{2} + \binom{N}{3})$ computations. The worst case appears when all N nodes are used to generate the subsets of `reachable`.

Algorithm 2: Drone Positioning Heuristic

```
covered = ∅;
POS = ∅;
sort nodes in descending consumption order and store them in sorted;
foreach i ∈ sorted do
    if s ∈ covered then skip i;
    covered := covered ∪ {i};
    ctemp := ctemp ∪ {i};
    pos = (0, 0, 0);
    search_neighborhood = 1;
    while search_neighborhood == 1 do
        select a node s ∉ covered as the shortest to i node;
        if defined s then
            ctemp := ctemp ∪ {s};
            compute drone position x, y, h using the nodes in ctemp;
            check = 1;
            foreach j ∈ ctemp do
                if j does not satisfy Eq. (4), (5) then
                    check = 0;
            if check == 1 then
                pos = (x, y, h);
                covered := covered ∪ {s};
        else
            search_neighborhood = 0;
    POS := POS ∪ pos;
return POS;
```

4.2 Drone positioning heuristic

The Drone Positioning Heuristic (DPH) (see Algorithm 2) is a fast alternative solution for the ODP problem. DPH examines only a subset of the possible drone positions starting from the nodes with higher consumption.

In its first step, DPH ranks the nodes in descending energy consumption order. It then successively picks the first ranking node and it tries to expand the recharge range covering more nodes with a single drone. To achieve this, it finds the closest neighbor and checks if conditions (4) and (5) are met. It abandons if the next shortest neighbor does not pass the recharge criteria or if no further expansion is possible. In this case, DPH stores the current drone position and initiates a new drone position `pos(x,y,z)`. Already covered nodes are excluded from future selections. The algorithm terminates by returning the set of positions `POS`.

DPH exhibits a low complexity since at the worst case it requires N^2 computations.

5 Evaluation & discussion of the results

5.1 Setup

We assume a scenario with a fixed size square terrain of 50 meters side and a variable number of nodes randomly and uniformly scattered on the terrain. We compare DPH's performance to the optimal solution derived by solving the ODP problem as it is transformed by Algorithm 1. To do so, we use an ILP set-cover solution that uses the GLPK[2] library. We measure the minimum number of positions as well as the execution time of the two approaches. Due to the presence of random values, we generate 25 instances per scenario and we run each instance 10 times; the average results are presented. The 95% confidence intervals are, also, displayed.

Regarding the node, charging and drone characteristics, we consider the following values: $P_{tx} = 24mA$, $P_{rx} = 20mA$, $P_{slp} = 1.45\mu A$, $\tau = 15$ minutes, $q \in [1..5]$ every τ minutes, $R = 12$, $M = 30$ minutes, $h_{min} = 3$m, $h_{max} = 10$m, $\sigma = 1$, $P_0 = 8mW$, $\theta = 120$ degrees, $\rho = 1$m, $\lambda = 0.4$ and $b = 1$. The transmission, reception and sleep mode power consumptions refer to Zolertia RE-Mote nodes[3]. Regarding the charger and the harvesting efficiency we used the values provided by the Powercast corporation for the P2110B harvesting receiver and a 3W effective isotropic radiated power charger operating at 915MHz.The algorithms were developed using the Perl programming language whereas the experiments were carried out on an Intel i7 2.7GHz CPU with 8GB RAM running Linux. No parallel processing was performed.

5.2 Results

Figure 4 depicts the number of positions for different node populations. We can observe that the number of positions increases almost linearly with the number of nodes. Additionally, DPH produces a slightly higher number of locations but close to the optimal solution. Indeed, the difference between the two approaches varies between 2 and 15%.

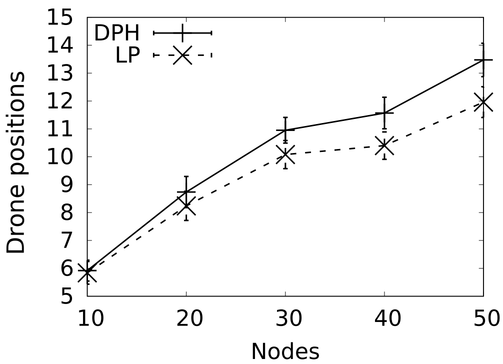


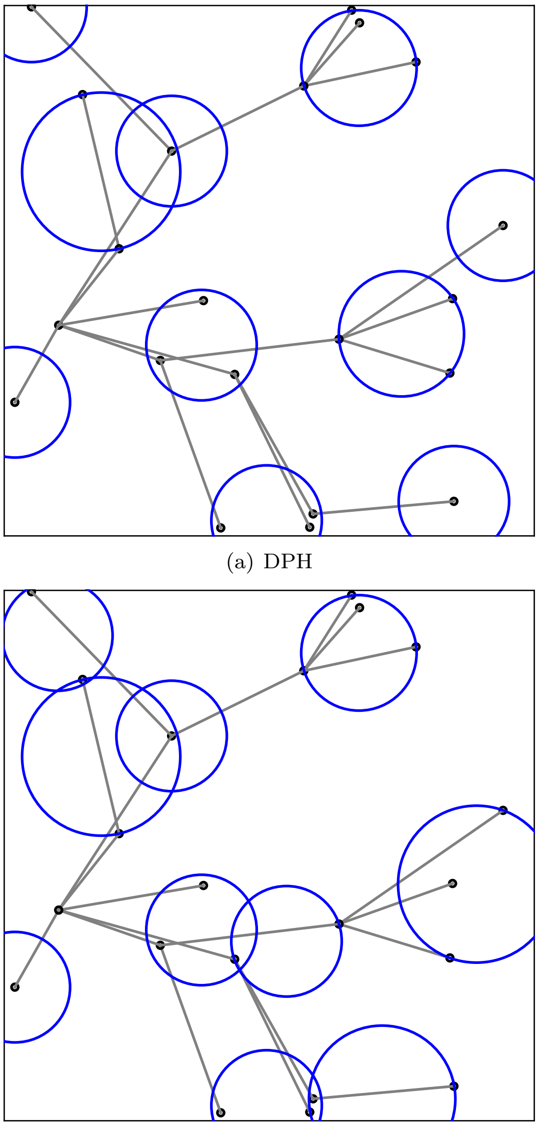
Figure 4: Number of positions for different node populations.

An example with the solutions of the two approaches for a scenario with 20 nodes is depicted in Figure 5. The small circles denote the nodes, the big circles are the projections of the drone positions onto the XY axes, and the lines represent the routes from each node to the sink (as they are computed by the RPL protocol using the preferred parent). The sink is the only uncovered node in the network. Both DPH and LP produce the same number of positions. However, LP

[2]https://www.gnu.org/software/glpk/

[3]https://github.com/Zolertia/Resources/raw/master/RE-Mote/Hardware/Revision%20B/Datasheets/ZOL-RM0x-B%20-%20RE-Mote%20revision%20B%20Datasheet%20v.1.0.0.pdf

sometimes double covers some nodes increasing the flying time. This happens because the only objective of the LP solver is the minimization of the number of positions producing sometimes overlapping drone projections.



(a) DPH

(b) LP

Figure 5: Solutions of the two approaches for a scenario with 20 nodes.

Figure 6 depicts the execution time of the two approaches. DPH requires up to 0.01 seconds for all the examined scenarios while the computation cost of the LP approach increases exponentially with the number of nodes. This makes DPH's execution feasible on devices with low power capabilities such as ARM-based drones. Indeed, we tested DPH on a Raspberry Pi Zero with 1GHz ARM CPU and the results show that the execution time does not exceed half of a second even for scenarios with many nodes (see Figure 7).

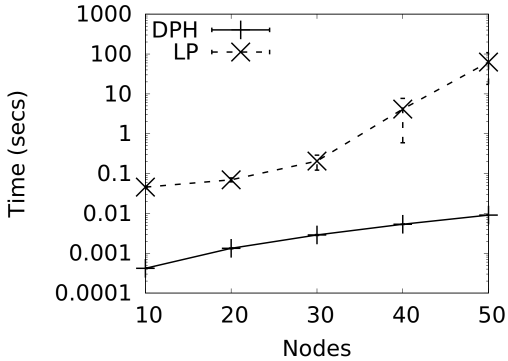


Figure 6: Execution time of the two approaches.

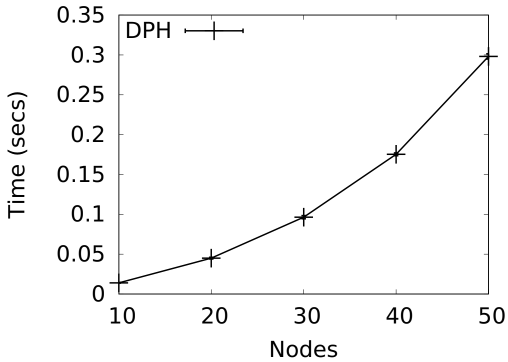


Figure 7: Execution time of DPH running on a Raspberry Pi Zero.

6 Conclusion & Future work

In this paper we studied the problem of computing the optimal drone positions needed to recharge an IoT network consisting of RF-power harvesting nodes. We showed that the problem can be transformed to the set-cover problem which is NP-Complete. We proposed an efficient algorithm for this purpose. We, also, proposed a lower complexity heuristic algorithm capable of running on hardware with limited processing power if there is such a need. Simulation results showed that the heuristic solution is faster while it performs close to the optimal. In the future we intend to evaluate the proposed solution using a real experimental environment and assess the impact of a noisy environment as well as of the presence of obstacles. Finally, we are interested in computing optimal drone tours combined with the multiple drone altitude concept and optimize both network lifetime and the number of vehicles.

Acknowledgments

This paper was carried out within the action “Strengthening Post Doctoral Research” of the “Human Resources Development Program, Education and Lifelong Learning”, 2014-2020, which is being implemented from IKY and is co-financed by the European Social Fund – ESF and the Greek government.

It has also received funding from the European Union’s Horizon 2020 Research and Innovation

Programme under Grant Agreement No 740477 – SAURON.

References

[1] A. Z. Kausar, A. W. Reza, M. U. Saleh, and H. Ramiah, "Energizing wireless sensor networks by energy harvesting systems: Scopes, challenges and approaches," *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 973 – 989, 2014.

[2] M. M. Butt, I. Krikidis, A. Mohamed, and M. Guizani, *Energy Management in Wireless Cellular and Ad-hoc Networks.* Springer, 2016, ch. RF Energy Harvesting Communications: Recent Advances and Research Issues, pp. 339–363.

[3] F. Guerriero, A. Violi, E. Natalizio, V. Loscri, and C. Costanzo, "Modelling and solving optimal placement problems in wireless sensor networks," *Applied Mathematical Modelling*, vol. 35, no. 1, pp. 230 – 241, 2011.

[4] L. Di Puglia Pugliese, F. Guerriero, D. Zorbas, and T. Razafindralambo, "Modelling the mobile target covering problem using flying drones," *Optimization Letters*, vol. 10, no. 5, pp. 1021–1052, Jun 2016.

[5] 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 *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2344–2352.

[6] H. Dai, Y. Liu, A. X. Liu, L. Kong, G. Chen, and T. He, "Radiation constrained wireless charger placement," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[7] D. Zorbas, P. Raveneau, and Y. Ghamri-Doudane, "On optimal charger positioning in clustered rf-power harvesting wireless sensor networks," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '16. New York, NY, USA: ACM, 2016, pp. 225–228.

[8] D. Zorbas, P. Raveneau, Y. Ghamri-Doudane, and C. Douligeris, "On the optimal number of chargers in battery-less wirelessly powered sensor networks," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, July 2017, pp. 1312–1317.

[9] C. M. Angelopoulos, J. Buwaya, O. Evangelatos, and J. Rolim, *Strategies for Wireless Recharging in Mobile Ad-Hoc Networks.* Springer, 2016, pp. 305–330.

[10] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Optimal charging in wireless rechargeable sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 278–291, Jan 2016.

[11] M. Zhao, J. Li, and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2689–2705, Dec 2014.

[12] A. Madhja, S. Nikoletseas, and T. P. Raptis, "Distributed wireless power transfer in sensor networks with multiple mobile chargers," *Computer Networks*, vol. 80, no. Supplement C, pp. 89 – 108, 2015.

[13] S. H. Chen, Y. C. Cheng, C. H. Lee, S. P. Wang, H. Y. Chen, T. Y. Chen, H. W. Wei, and W. K. Shih, "Extending sensor network lifetime via wireless charging vehicle with an efficient routing protocol," in *SoutheastCon 2016*, March 2016, pp. 1–2.

[14] S. A. Chowdhury and A. Benslimane, "Recharging of wireless sensor network using kmec with dynamic active zone strategy," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2016, pp. 1–6.

[15] C. M. Angelopoulos, J. Buwaya, O. Evangelatos, and J. Rolim, "Traversal strategies for wireless power transfer in mobile ad-hoc networks," in *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '15. New York, NY, USA: ACM, 2015, pp. 31–40.

[16] C. Detweiler, M. Eiskamp, B. Griffin, J. Johnson, J. Leng, A. Mittleider, and E. Basha, *Unmanned Aerial Vehicle-Based Wireless Charging of Sensor Networks.* Springer, 2016, pp. 433–464.

[17] E. Basha, M. Eiskamp, J. Johnson, and C. Detweiler, "Uav recharging opportunities and policies for sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 824260, 2015.

[18] N. W. Najeeb and C. Detweiler, "Extending wireless rechargeable sensor network life without full knowledge," *Sensors*, vol. 17, no. 7, 2017.

[19] IEEE Std 802.15.4e-2012, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," 2012.

[20] S. Duquennoy, A. Elsts, A. Nahas, and G. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2015)*, Ottawa, Canada, Nov. 2017.

[21] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and A. R., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, 2012.

[22] 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.

[23] S. Xu, R. M. Freund, and J. Sun, "Solution methodologies for the smallest enclosing circle problem," *Computational Optimization and Applications*, vol. 25, no. 1, pp. 283–292, Apr 2003.