Unmanned Aerial Vehicles Positioning Scheme for First-responders in a Dynamic Area of Interest

Paulo Alexandre Regis, Amar Nath Patra, Shamik Sengupta
Department of Computer Science and Engineering
University of Nevada, Reno, USA, 89557
{pregis, apatra}@nevada.unr.edu, ssengupta@unr.edu

***Abstract*—In this paper, we explore the problem of finding the initial positions to deploy UAVs to provide service to ground personnel. In this work, we propose a greedy algorithm that finds a feasible solution that guarantees 100% coverage on any given map. We defined subareas of the map which do not need coverage as excluding zones. We considered both excluding zones and the connectivity constraints in the algorithm. By avoiding these zones and keeping all nodes connected to the network we can reduce the total number of UAVs needed to provide coverage without losing the communication capability. We show that the complexity of the algorithm is linear with respect to the input parameters. Simulation results show the behavior of our approach with different maps. We observe a convergence on the number of nodes needed when varying the input map.**

Keywords*—*Positioning, deployment, multi-hop, ad-hoc network

I. Introduction

In recent years we have been experiencing a surge in wireless communication due to the advancement of hand-held equipment and their original pervasive applications. The immense need for wireless connection, coupled with the new promising widespread applicability of lightweight Unmanned Autonomous Vehicles (UAVs) brings a new era of ad-hoc networking. Advancements in hardware, such as micro and small UAVs and Unmanned Ground Vehicles (UGVs), increase the application range of wireless mesh networks and introduce a new dimension to the next generation wireless networking and service provisioning. This evolution makes temporary wireless coverage, environmental monitoring, and other services possible when dedicated infrastructure is unavailable [1]. For example, situations like mission-centric exercises performed by tactical military teams, first-responders, and firefighters, would be possible in disaster-affected zones [2], [3], [4].

The enhancement in wireless technologies enables throughput hungry applications such as video streaming, real-time data transfer, and monitoring. A network of UAVs can, for example, form a temporary backbone network to provide connectivity/coverage to deployed ground teams even when they cannot directly communicate with each other, but by relaying the data through this UAV network [5]. One of the challenges in a service providing network is finding the right deployment positions of the nodes to cover as much area as possible without breaking their inter-connectivity [6]. In a tactical and disaster areas, this temporary backbone network must provide service to the deployed ground teams to avoid isolated personnel [7]. The positioning scheme should maximize the coverage area while minimizing the total number of UAVs needed to avoid unnecessary interference and save resources.

Ideally, deployed nodes should be placed as far as possible to maximize the coverage area, but close enough to allow the communication packets to be relayed with minimal loss. The infinite possible shapes of the map to be covered pose a significant challenge when designing a generic solution that complies such limitations. The area that needs to be covered can have infinite shapes, and it becomes even more complicated if sub-regions in this larger area do not need to be covered. These sub-regions can represent non-interesting areas, unrelated to the mission the network is trying to accomplish, like bodies of water or deserted lands, thus do not need to be covered.

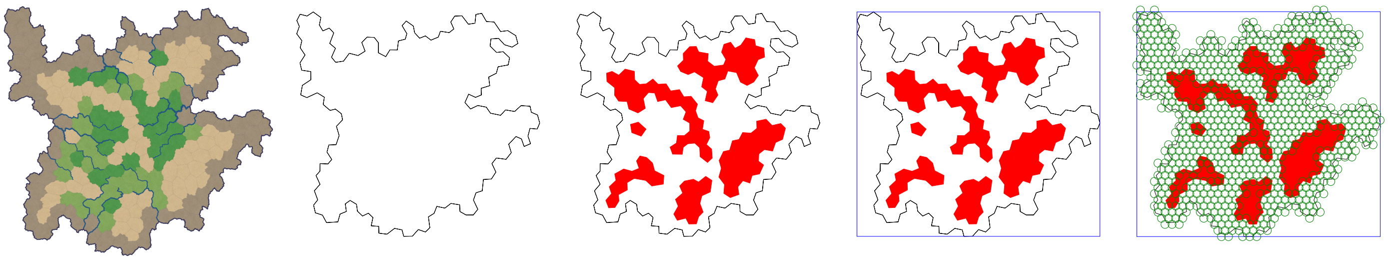
Randomly deploying UAVs on a map might not yield the optimal solution. In a network with limited devices available, finding positions to deploy them in a way that minimizes the number of UAVs needed while providing the same level of coverage, is extremely important because we have the constraint of maintaining the inter-connectivity among themselves. In this paper, we try to minimize the number of nodes needed while guaranteeing maximum coverage on any given map, while considering the network connectivity constraint to maintain every node reachable. We start with a Delaunay triangulation with equilateral sides [8], and then perform various operations and geometric calculations [9]. The algorithm may not find a globally optimal solution but guarantees to generate a solution in a considerably less amount of time.

We propose a greedy algorithm to find the least number of UAVs needed to provide coverage for a given map. We provide guidelines and assumptions to prepare the input data before using the algorithm. An initial triangulation results in the upper bound of the algorithm, in other words, the maximum number of nodes needed. Following the initial positions, it executes a set of geometric calculations and discards nodes that do not contribute to the coverage objective in the input map. The procedure is repeated for a maximum number of iterations. At each iteration, the positions are slightly shifted horizontally or vertically. The solution found by the algorithm is the calculation that yields the least number of UAVs necessary to provide coverage without breaking connectivity. Remembering, however, that it may not be the globally optimal solution. To the best of our knowledge, this is the first work on temporary UAV backbone positioning in a dynamic area of interest with excluding zones to consider connectivity as constraint [1], [10], [11].

The remainder of the paper is organized as follows. Section II describes the system model and assumptions, as well the problem definition. Section III explains the proposed mechanism. In Section IV, we describe and discuss the numerical results. Finally, Section V concludes the paper.

II. System model

In this section, we describe the assumptions that enable the proposed mechanism and explain the reasoning behind it. As mentioned before, our approach aims to tackle the problem of node placement in a known map. In other words, we want to place nodes throughout a given map in a manner



(a) Original map **(b)** Area of interest contour **(c)** Excluding zones in map **(d)** Bounding box of map **(e)** Computed solution

Fig. 1: From 1a to 1e: extract the map information and model it as a polygon. Extract the areas which should not be covered, representing them as polygons. The algorithm finds a set of nodes that guarantees 100% coverage of the area of interest.

that guarantees coverage on the entire map, while respecting network constraints. We do it by overlapping two layers: the map and the Delaunay equilateral triangulation, and a greedy algorithm that shifts the position of one of the layers. Although the proposed mechanism provides a viable solution, it may not yield the optimal one. However, it does accept any generic map as an input, as long as it follows the guidelines described next.

Area of interest (AoI): in our mechanism, we assume maps can be represented as polygons, regardless if they are regular or irregular, convex or concave. The area of interest of a map is the portion that needs to be covered, we define it as $\mathcal{M}$, representing the sequence of vertices that form the polygon. An example can be seen in Figure 1b. The area of interest might have some non-interesting sub areas, such as deserts or large body of water, that might not require any coverage. Next, we show how we incorporate these sub areas in our system with what we define as the excluding zones.

Excluding zone(s) (EZ): to allow areas inside an area of interest, that do not represent an region that needs coverage, or obstacles like buildings in a city block, we coin the term excluding zones. These zones, similar to the area of interest, are defined as generic polygons. A map may have multiple excluding zones. The only requirements are that all excluding zones are placed inside the area of interest: $e_i \subset \mathcal{M}, \forall e_i \in \mathcal{E}$; and each excluding zone does not overlap with another one, otherwise they could be combined into a single zone: $e_i \cap e_j = \emptyset, \forall e_i, e_j \in \mathcal{E}, i \neq j$. Figure 1c shows both the area of interest and some excluding zones contained in it.

Coverage range: our model assumes an application scenario where UAVs are deployed to provide service to ground users. However, it can be any type of sensor with this characteristic. We model the coverage range as a simple function of the height and the beamwidth angle of a directional antenna used to serve users on the ground. In this paper, we use coverage and sensing range interchangeably.

Directional antennas are known to have a main lobe, which in our case is pointed towards the ground by the UAV. Figure 2 shows an example of such pattern and illustrates the physical meaning of the sensing/coverage range in this paper ($sensing\_range = f(h, \theta) = h \tan(\theta)$). Based on the radiation pattern, we define the aperture angle θ as half the beam-width of the main lobe. The sensing range is extracted as the maximum distance from the peak effective radiated power to the half beam distance ($-3dB$), based on a sensitivity of the receiver, and the latitude of the UAV. In this case, for example, if someone is located in the shaded area, the communication might not be possible if the UAV is flying too high (the gain in the shaded zone, due to path loss, might not meet the sensitivity

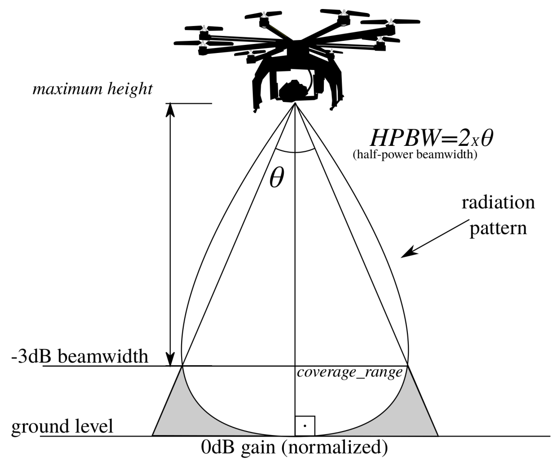


Fig. 2: Sensing/coverage range based on the normalized radiation pattern.

of the receiver), but it would be possible if the receiver was located closer to the center of the lobe.

The interference between cells is out of the scope of this project. For example, in a multi-channel multi-radio network, interference can be avoided by assigning different non-interfering channels to neighboring nodes [12]. In this paper we consider that neighboring UAVs can transmit in orthogonal channels, avoiding interference.

Since we consider each node to be a UAV, it is possible to adjust the height of the node, thus increasing or decreasing the coverage range. However, there is a maximum limit for the height. In our model, the maximum height depends on the irradiated power and sensitivity of the receiver. We consider all UAVs to be in the same altitude, and the fleet to be homogeneous (i.e. they are all equal, have the same capabilities).

Network constraint: in our system model, we consider a fixed communication range between the backbone (UAV) network. To avoid interference, we once again assume the UAV network is established in a different frequency from the service provisioning frequency (i.e. the directional antenna operates in a different channel). This communication range will directly affect the selection of the radius used by our algorithm to compute the placement of the nodes. We define the $radius = \min(\sqrt{3}sens\_range, comm\_range)$. Where $\sqrt{3}sens\_range$ represents the distance between two nodes in a Delauney distribution using equilateral triangulation method with $sens\_range$ value. In other words, it is the side of the equilateral triangle. By deciding between the lower value, we assure that the resulting placement maintains the network connectivity while covering the entire area of interest. In the future we want to consider directionality in the backbone UAV network, either to improve network performance or avoid possible attacks to the network [13].

A. Problem

The problem of finding the optimal positioning of nodes in a particular area has been studies and is notoriously difficult [14]. We define the objective of our algorithm as an optimization problem in terms of the areas of the map, the excluding zones, and the solution. The goal is to minimize the difference between the total area covered by the positioning of the nodes and the area of interest of the map. The objective and constraints are as follows.

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \quad & f(\mathbf{X},\mathbf{Y}) = |\mathbf{X}| = |\mathbf{Y}| \\ \text{subject to} \quad & A_{\mathcal{M}} - \sum A_e \subset \bigcup A_{\mathcal{C}} \\ & A_{e_i} \subset A_{\mathcal{M}}, \forall e_i \in \mathcal{E} \\ & distance(x_i, y_i, x_j, y_j) \leq radius, |i-j| = 1 \end{aligned}$$

Where **X** and **Y** are matrices with the (x, y) positions of each node. $|\mathbf{X}| = |\mathbf{Y}|$ represents the total number of nodes needed. $A_{\mathcal{M}}$ is the set of points inside the area of interest, $\sum A_e$ is the set of points in all the existing excluding zones, and $A_{\mathcal{C}}$ is the set of points inside the covered area by nodes.

The constraints mean that each adjacent pair of nodes is in communication range. It also means that each point contained inside the AoI that is not in an excluding zone must be inside one of the nodes coverage areas. This constraint will ensure that the entire area of interest is covered by the UAVs while minimizing the number of nodes needed to do that.

III. Positioning mechanism

In this section, we present and discuss the proposed mechanism. The algorithm assumes the input is sanitized, meaning that the area of interest $\mathcal{M}$ and each excluding zone $e \in \mathcal{E}$ are polygons with no intersecting edges. Each excluding zone must not overlap with one another: $e_i \cap e_j = \emptyset, \forall e_j \in \mathcal{E}, i \neq j$.

The first step is to compute the bounding box of the given area of interest. This step only takes a single iteration over the input map coordinates. We can see the resulting bounding box for our example map in Figure 1d, represented by the blue rectangle surrounding the map.

Next, given the input deployment radius, the maximum number of nodes max_nodes needed to cover the map in the worst case scenario (if the map is a rectangle-shaped with no excluding zones) is calculated, i.e., the number of rows and the number of nodes per row, num_rows and per_row respectively, multiplied.

Then it iterates for a specified number of times ($MAX\_X \times MAX\_Y$). At each iteration, the initial positioning is shifted by a fraction of the radius input. Given the symmetry of the initial triangulation, the shifted positioning will never go beyond the point where a node's position is the same as another node's initial position when $MAX\_X = MAX\_Y = 0$. For each iteration the algorithm calculates the number of nodes needed to cover the entire area of interest, discarding the nodes fully contained in any excluding zone. The algorithm keeps track of the minimum number of nodes needed to cover the map and returns the set of nodes positions $\mathcal{N}$. We show the entire algorithm in Algorithm 1 and a summary list of the symbols used in Table I.

Algorithm 1: Positioning

Input: Polygon of interest $\mathcal{M}$, excluding zones list $\mathcal{E}$, deployment radius R, MAX_X and MAX_Y
Output: List of nodes positions $\mathcal{N}$

$\mathcal{N} \leftarrow \emptyset$
$box \leftarrow bounding\_box(\mathcal{M})$
$num\_rows \leftarrow \frac{height(box)}{\frac{\sqrt{3}}{2}R}$
$per\_row \leftarrow \frac{width(box)}{R}$
$max\_nodes \leftarrow num\_rows \times per\_row$
for $\Delta x \in \{0, 1, ..., MAX\_X\}$ **do**
    for $\Delta y \in \{0, 1, ..., MAX\_Y\}$ **do**
        $\eta \leftarrow \emptyset$
        $\delta x \leftarrow \frac{R}{MAX\_X}\Delta x$
        $\delta y \leftarrow \frac{\frac{\sqrt{3}}{2}R}{MAX\_Y}\Delta y$
        for $n \in \{0, 1, ..., max\_nodes\}$ **do**
            $row = \lfloor \frac{n}{per\_row} \rfloor$
            $x \leftarrow row \mod 2\frac{R}{2}$
                $+R(n - row \times per\_row) - \delta x$
            $y \leftarrow \frac{R\sqrt{3}row}{2} - \delta y$
            $C \leftarrow circle((x, y), R)$
            if $\mathcal{M} \cap C \neq \emptyset$ **then**
                if $C \not\subset e, \forall e \in \mathcal{E}$ **then**
                    $\eta = \eta \cup \{C\}$
        if $|\eta| < |\mathcal{N}|$ *or* $|\mathcal{N}| = 0$ **then**
            $\mathcal{N} \leftarrow \eta$

TABLE I: Glossary for Algorithm 1

Symbol	Description
$\mathcal{M}$	the map, defined by a list of vertices
$\mathcal{E}$	excluding zones, a list of lists of vertices
$\mathcal{N}$	a list or coordinate tuples, representing the placement of each node
$e \in \mathcal{E}$	a single excluding zone, defined by a list of vertices
$C = circle((x, y), r)$	circle object defined by a position tuple (x, y), and radius r
$bounding\_box(p)$	function that returns the bounding box of a polygon
δx, δy	the distance variation calculated for each iteration
Δx, Δy	the total number of iterations in each axis (x and y)

A. Analysis of the algorithm

The complexity of the algorithm is: $O(\frac{2hw}{\sqrt{3}R^2}(|\mathcal{M}| + |\mathcal{E}|\bar{e}))$ Where h and w are the height and width of the map $\mathcal{M}$, defined by the maximum and minimum values in each axis, or in other words, the sides of the bounding box of the polygon. The growth is always linear with respect to the input data. $|\mathcal{E}|$ represents the number of excluding zones, and $\bar{e}$ is the average size of the excluding zones: $\bar{e} = \frac{\sum^{\mathcal{E}} |e|}{|\mathcal{E}|}$.

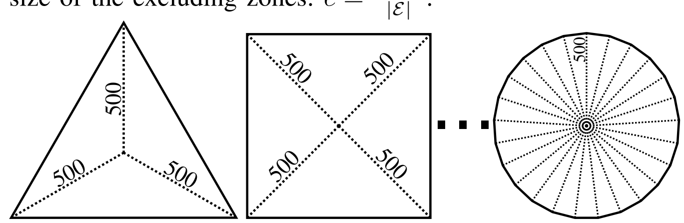


Fig. 3: The variation of number of sides of the AoI

TABLE II: Parameters used when varying the area of interest

Parameter	Value
MAX_X	100
MAX_Y	100
$coverage\_range$	$100m$
$comm\_range$	$100\sqrt{3}m$
Benchmark radius	$100\sqrt{3}m$
Number of sides	3 to 25

IV. Simulations

We test our algorithm in two distinct yet correlated scenarios. First, we assume a regular polygon as the area of interest, in this case, there is no excluding zones. First, we vary the number of sides of the area of interest from 3 to 25 and radius of $500m$, but maintain the same radius throughout different shapes as seen in figure 3. The 25 sides result in a polygon with side $125.33m$, less than the triangulation side: $100\sqrt{3}$. The sensing/coverage range can be calculated based on a height of $173.2m$ using the Friis path loss equation, a beamwidth of $\frac{\pi}{3}$, and a radiation pattern with the peak gain of $0dB$ [15]. We use transmission power of $1W$ from the directional antenna that provides connectivity to ground teams. The sensitivity of the receiver is $-60dBm$. Table II lists the parameters used.

In the second scenario, we define the area of interest as a square side of $1000m$. Inside the square, we place a single excluding zone. Then we vary the excluding zone shape as we did in the first case. Figure 4 exemplifies this scenario.

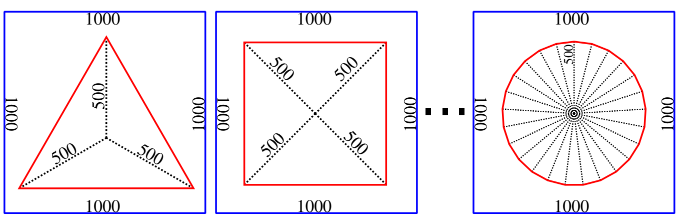


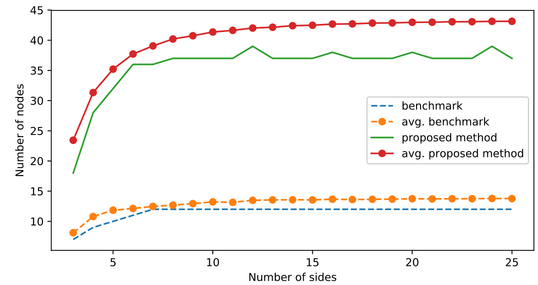
Fig. 4: The variation of number of sides of the EZs

Benchmark: we compare our results with the hexagonal tiling circle packing algorithm that provides the highest density possible [14]. We use it as the lower bound of the number of nodes. Since it expects gap areas between nodes, it does not comply with our requirements. However, it provides a good comparison as it will always yield fewer nodes than the optimal NP-hard solution [14]. This comparison provides an insight on how close or far from the optimal solution our method is.

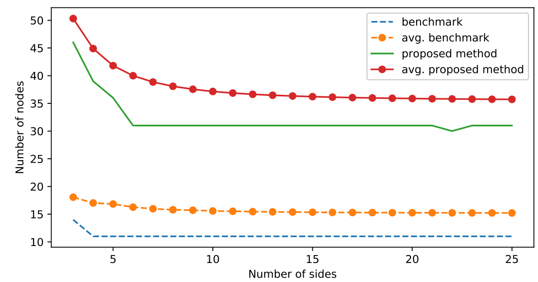
A. Results and discussion

We implemented the system d in Python 3.6.1, and simulations performed on an Intel Core i7 (3.4 GHz), 16 GB ram computer, running Ubuntu 16.04. We compare the final result of each algorithm, as well as the average result between all the iterations ($MAX\_X * MAX\_Y$) for each case. We compare our method with the benchmark according to the performance indicators as follows.

- **Number of nodes**: size of the solution yielded by the algorithms
- **Uncovered area**: areas in the AoI that are uncovered by the solution. The benchmark allows gaps between nodes, while our method guarantees this to be *zero*
- **Wasted coverage**: the total area covered by the solution and not contained in the AoI
- **Time to compute**: time taken to compute the solution. We run the algorithm 10 times and plot the average of these values to reduce the effect of external variables



(a) Number of nodes needed for regular polygon AoI



(b) Number of nodes needed for regular polygon EZ

Fig. 5: Number of nodes

Figure 5 shows how the number of nodes needed by each algorithm converges as the number of sides of the AoI or the EZ increases. Each spike in the plot 5a suggests that there might be a better positioning solution. However, as the number of sides increases, we observe that the trend stabilizes, not deviating too much from the mean.

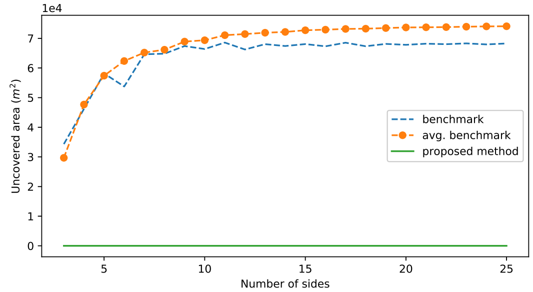
We see in Figure 6 how much area the benchmark scenario leaves uncovered. While our proposed method guarantees 100% coverage of the AoI, the benchmark has in the order of hundreds of thousands of m^2 without coverage for the same AoI. In contrast, Figure 7 reveals how much coverage is wasted by the solution. The vast amount of wasted area yielded by our mechanism is a direct consequence of the larger number of nodes needed to cover the entire AoI.

Figure 8 compares the time taken to compute the solution for each method. We observe that the difference is minimal, in the order of milliseconds.

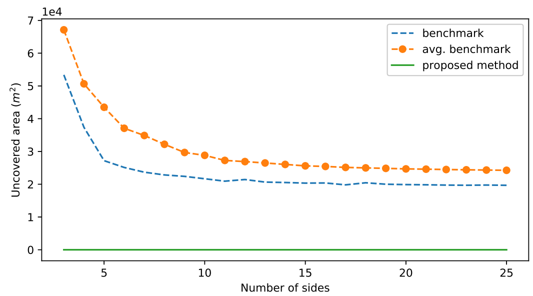
V. Conclusion and future work

We presented an algorithm to find a feasible solution that guarantees coverage of a given area. The algorithm has linear complexity, allowing fast computation. It accepts any arbitrary map as an input modeled according to the provided guidelines. The algorithm allows sub areas inside the map where coverage is unnecessary. We compare the results to a benchmark scenario. It yields more nodes than the benchmark; however, it guarantees coverage of the AoI. The computation time slightly increases when compared to the benchmark results.

In the future, we want to extend this solution to allow dynamic maps. If a map is modified (i.e., a wildfire region is expanding, coverage/position must change) we want to reorganize the nodes without computing the entire solution for the map, but only for the modified part. It would enable self-positioning intelligent network in the field. We also want to extend to allow 3D positioning, allowing nodes to be placed in different heights while still connected.

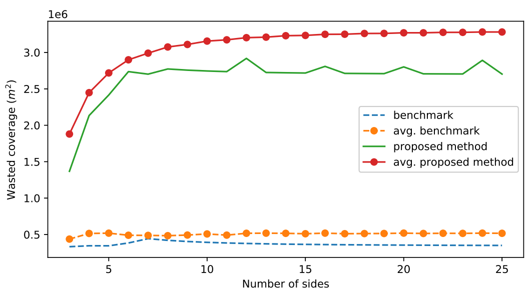


(a) Area uncovered in the regular polygon AoI

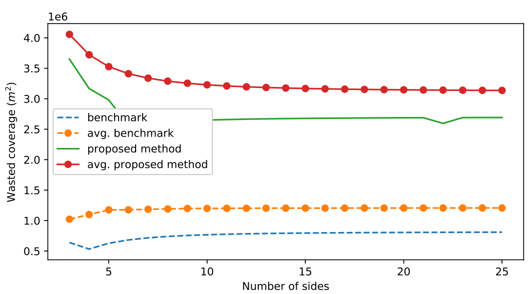


(b) Area uncovered in the regular polygon EZ

Fig. 6: Uncovered area



(a) Wasted coverage area in the regular polygon AoI



(b) Wasted coverage area in the regular polygon EZ

Fig. 7: Wasted coverage area

ACKNOWLEDGMENTS

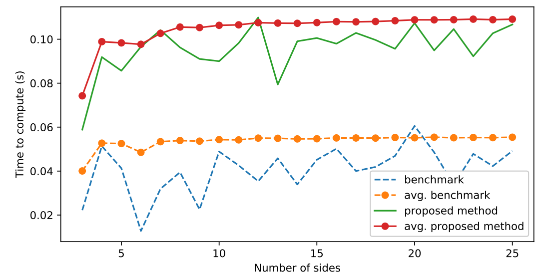
This research was partially supported by the NSF Partnership for Innovation Award #1430328, and CAPES Brazil 13184-13-0.

REFERENCES

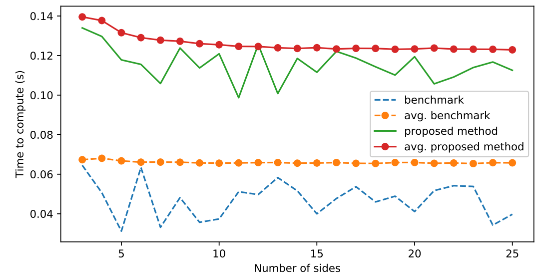
[1] A. N. Patra and S. Sengupta, "Dynamic deployment of uav-enabled floating access points for serving hot zones," in *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pp. 1–8, July 2017.

[2] A. Merwaday and I. Guvenc, "Uav assisted heterogeneous networks for public safety communications," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*, pp. 329–334, IEEE, 2015.

[3] E. P. de Freitas, T. Heimfarth, C. E. Pereira, A. M. Ferreira, F. R. Wagner, and T. Larsson, "Evaluation of coordination strategies for



(a) Time to compute the solutions for the regular polygon AoI



(b) Time to compute the solutions for the regular polygon EZ

Fig. 8: Time to compute final solution

heterogeneous sensor networks aiming at surveillance applications," in *Sensors, 2009 IEEE*, pp. 591–596, IEEE, 2009.

[4] A. T. Erman, L. van Hoesel, P. Havinga, and J. Wu, "Enabling mobility in heterogeneous wireless sensor networks cooperating with uavs for mission-critical management," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 38–46, 2008.

[5] M. F. Pinkney, D. Hampel, and S. DiPierro, "Unmanned aerial vehicle (uav) communications relay," in *Military Communications Conference, 1996. MILCOM'96, Conference Proceedings, IEEE*, vol. 1, pp. 47–51, IEEE, 1996.

[6] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A los map approach," in *Communications (ICC), 2017 IEEE International Conference on*, pp. 1–6, IEEE, 2017.

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

[8] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 7, 2007.

[9] D.-T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.

[10] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Dean, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *arXiv preprint arXiv:1803.07926*, 2018.

[11] H. X. Pham, H. M. La, D. Feil-Seifer, and L. Van Nguyen, "Cooperative and distributed reinforcement learning of drones for field coverage," *arXiv preprint arXiv:1803.07250*, 2018.

[12] A. Raniwala, K. Gopalan, and T.-c. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, pp. 50–65, Apr. 2004.

[13] S. Bhunia, P. A. Regis, and S. Sengupta, "Distributed adaptive beam nulling to survive against jamming in 3d uav mesh networks," *Computer Networks*, 2018.

[14] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the first workshop on Hot topics in software defined networks*, pp. 7–12, ACM, 2012.

[15] H. T. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, pp. 254–256, May 1946.