VESPA: Constrained target coverage by distributed deployment of connected UAVs

Yann Busnel, Christelle Caillouet, David Coudert

VESPA: Constrained target coverage by distributed deployment of connected UAVs

Yann Busnel[1], Christelle Caillouet[2], David Coudert[2]

[1] IMT Atlantique, IRISA, CNRS UMR 6074
[2] Université Côte d'Azur, Inria, I3S, CNRS, France

Abstract

Providing network services access anytime and anywhere is nowadays a critical issue. A natural response to such a need is the use of autonomous flying drones to provide network services. We propose VESPA, a distributed algorithm using only one-hop information of the drones, to discover targets with unknown location and ensure connectivity between them and the sink, auto-organizing in a multi-hop aerial wireless network. We prove that connectivity, termination and coverage are preserved during all stages of our algorithm, and we evaluate the algorithm performances through simulations. A comparison with an existing work has been made as well to validate our approach and show the efficiency of VESPA.

1 Introduction and Background

Providing network services access anytime and anywhere has become an important challenge during the recent years. This challenge is worsen by the environmental context such as vehicular networks, environmental monitoring in harsh conditions or disaster recovery. For the later, providing ground access to network services is almost impossible. However, fair quality of emergency services must be provided. A natural response to such a need is the use of autonomous flying drones or Unmanned Aerial Vehicles (UAV) to provide the needed services to targets, or collect continuously monitored information from the ground [HS18]. The autonomous air system must be able to retrieve data from ground nodes and transfer them to a base station, or sink, located at the center or at the edge of the field. Oppositely, emergency information or temporary network services must be efficiently provided to the ground nodes by the rescue team through the fleet of drones. Network connectivity for air-to-ground and air-to-air communications must be ensured by the deployment while maximizing the monitored area by the drones. In this paper, we consider the following problem : Given a set of targets located on the ground field, the goal is to monitor the area with flying drones to cover as many targets as possible and ensure the connectivity with a base station to continuously report or broadcast information to and from the sink.

The use of a fleet of drones has been studied both from theoretical and practical considerations. Authors of [PGZR16] propose an optimal formulation of the drone location problem to cover a set of targets and a localized algorithm, where each drone autonomously cooperates with neighboring drones in order to minimize the cost of deployment in terms of number of drones. Connectivity considerations between the drones to gather information to a fixed base station has been studied in [CGR19]. Authors propose an optimization framework for optimally deploy a set of drones to cover ground targets while optimizing both the drones altitude and the total deployment cost. A cooperative search and coverage algorithm is presented in [LGF18] where the drones are deployed to explore the environment in order to gather information about it and concentrate the UAVs around targets to capture them as soon as possible. Global connectivity among drones and a sink is not considered.

Covering points of interest (PoI) has also been studied in mobile wireless sensor networks (M-WSN) [HSDH19, MHS17]. In such networks, the goal is to ensure that deployed mobile sensors

provide the required coverage for the area of interest, while ensuring connectivity of the deployed network. In [EFGR19], a decentralized approach deploy the sensors such that they monitor the environment as long as possible and cover a surface as large as possible. The closest work to ours propose the Spread and Shrink (SaS) algorithm that uses only surrounding information and local interactions with mobile robots within range [REZN17]. SaS implements both the discovery phase and the coverage phase in a decentralized manner. In the discovery phase, mobile sensors spread to discover new PoI throughout the field and in the second phase, they deploy to focus only on the discovered PoI. Connectivity is guaranteed during both phases to gather information from the found targets to a central sink.

We develop here an iterative algorithm (called VESPA for "Vehicle Spreading using Self-organized Parallel Algorithm") repeating the detection and the coverage of the targets by the available drones, while guaranteeing the connectivity to the central sink for all the discovered targets. In other words, we extend existing works in the following way :

- We assume that the target locations are unknown so that the drones must fly over the field and detect the targets while flying over.
- We propose a distributed algorithm run by each drone using only information from their neighbors.
- We ensure connectivity between the discovered targets and the sink using drones forming a multi-hop aerial wireless path. Once a target is detected, the drones self-organize to construct the path.
- We then expand again to fully use the available drones and maximize the size of the observed area and the efficiency of the target detection.

The rest of the paper is organized as follows. Next section introduces the model considered. We detail VESPA, our distributed algorithm in Section 3 and prove its correctness and guarantees in Section 4. Results are presented in Section 5 showing the effectiveness of VESPA compared to SaS, and we conclude in Section 6.

2 Model

We consider a population $\mathcal{D}$ of drones, identified by their id d_i, $i \in \{1, \ldots, |\mathcal{D}|\}$, and located in the three dimensional space. Let $p_i = (x_i, y_i, z_i)$ be respectively the position (x_i, y_i) of drone d_i in the 2D plane, and z_i its altitude. The sink $\mathcal{S}$ is located at $p_{\mathcal{S}} = (x_{\mathcal{S}}, y_{\mathcal{S}}, z_{\mathcal{S}})$. Each drone d_i is equipped with an omnidirectionnal antenna. It is able to communicate with other drones located within a ball of radius R_{p_i} (commonly assumed to be in the order of 20 meters). Let $\mathcal{N}_i$ denotes the set of all reachable drones in the ball centered on d_i.

We consider the presence of a set $\mathcal{T}$ of target points, or PoI, on the monitored area. Each target $\tau_i \in \mathcal{T}$ has a fixed position $p_{\tau_i} = (x_{\tau_i}, y_{\tau_i}, 0)$, that is unknown to the drones and the sink. A target is discovered and covered by a drone if it is located inside the area intersecting the communication ball of the drone with the ground. In order to avoid sensing holes and to ensure an optimized coverage, we use a triangular tessellation. Each drone is located in the center of an hexagon, and the communication range R_{p_i} ensures that it can communicate with the 6 drones located at the center of the neighboring hexagons following the triangular tessellation (Figure 1). Also the coverage area of the drone is assumed to encapsulate the hexagon to ensure complete coverage of the area. It does not restrict the drones to fly at the same altitude.

3 VESPA algorithm

All the n drones start at position $p_{\mathcal{S}}$. The absolute geographical positions are unknown, but drones can deduce their relative positions among each other. Each drone runs independently the

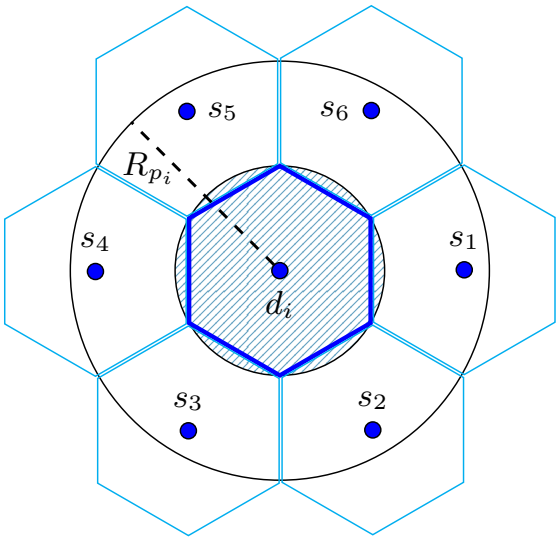


Figure 1: Triangular tessellation model.

algorithm following an alternating of 3 phases: *(i)* expansion, *(ii)* spanning, and *(iii)* balancing. During the process, each drone can take 3 possible states:

- **irremovable**: these drones will stay in their current position. They are essential for ensuring the connectivity of the discovered targets and the aerial path to the sink;
- **border**: these drones are the ones surrounding the covered area, representing the edge of the expansion zone;
- **free**: these drones are those that can move freely, *i.e.*, they are neither irremovable nor border drones.

To ensure connectivity during the successive expansion phases, we allow the drones to be simultaneously irremovable and border as shown in the next sections.

3.1 Expansion phase

In the expansion phase, the drones move away from the sink to detect the targets. They all start in the free state on the initial expansion phase.

Move selection We consider each drone independently and located at spot s_0 corresponding to the center of the hexagon. If it is free, a drone decides if it moves to another location within spots $s_1, \ldots, s_6$ (centers of neighboring hexagons in Figure 1) or remains at location s_0. If there are multiple drones located at s_0, the algorithm ensures that at least one drone will stay at this position to maintain the coverage and connectivity. To do so, each drone is given a priority based on their unique identifier or any other chosen metrics, and the one with the smallest priority is not allowed to move. When a drone can move, it first assigns each spot s_i a value v_i as follows, and then chooses the spot with the smallest value. Let w_i be the number of drones located at spot s_i, and C and ϵ be constants.

- Each unoccupied spot s_i (i.e., such that $w_i = 0$) receives a value $v_i = \frac{d(s_i,\mathcal{S})\times C}{4\times d(d_i,\mathcal{S})}$ that is proportional to the distance of the spot to the sink. This helps avoiding a linear expansion letting uncovered holes closer to the sink.
- Each occupied spot s_i that moves away from the sink is assigned a value v_i picked randomly in $[\omega_i \times C + \epsilon, (\omega_i + 1) \times C[$, where ϵ is a small value preventing the overlap of the generated

values. These spots are chosen as a priority since they favour the expansion of the drones over the area.

- Each occupied spot s_i that is closer to the sink has value $v_i = \infty$ to prevent drones to fly back to the sink if a drone is already on the spot.

If all the spots around drone d_i have infinite value, then the drone stays at s_0.

Termination When a drone d_i is alone on a spot, it senses if the 6 neighboring spots are occupied or not. If there is at least one drone on each spot s_i, $i \in \{1, \ldots, 6\}$, then d_i remains in free state. Otherwise, d_i changes to border state and sends a message to a neighboring drone following the *right-hand rule*. Starting from the empty zone corresponding to the area containing the unoccupied spots, d_i selects the first drone in clockwise direction as destination node for its transmission, indicating its neighbor that it reaches the border state. When a drone receives such a message, it drops it if it is in a free state, or forwards it if it is in border state following the right-hand rule. Following the forwarding process, when d_i receives its message back, this means that it has not encountered a drone in a free state. The drone then sends a broadcast message indicating the end of the expansion phase and wait for a waiting time correlated to the diameter of the expansion area.

Further expansions In addition to the initial expansion phase, which aims to cover the largest possible area around the sink, the subsequent expansion phases must cover the largest area around the last created border. Thus, we must slightly modify the expansion rules, forbidding drones to fly back from a spot of the border to a spot in the region surrounded by the border (information deduced from the termination of the previous expansion phase). The termination of this phase and the discovery of the new external border, are similar to the first expansion phase.

3.2 Spanning phase

At the end of the expansion phase, each spot is either empty or populated by exactly one drone. Each drone located on a spot including a PoI, becomes irremovable and will seek both to create the shortest multi-hop communication path to the sink, and to connect to the nearest edge drone to ensure the connectivity of the network as a whole.

Path construction Each drone, which has changed from a free state to an irremovable state (*i.e.*, discovered a PoI in its current zone), sends a message to its neighboring drone located in the zone closest to the sink. If the latter is already an irremovable node, then the path construction process stops but the message still follows the existing path until it reaches the sink. Otherwise, it also switches to the irremovable state and iterates the process again. Since we have a finite number of drones, the process ends when the sink is reached. In parallel, the first drone of the process also sends another message, but in the opposite direction to the previous message to the sink. Thus, it transmits a message to the drone whose position is furthest from the sink in its neighbourhood. The process will iterate again until it reaches the edge of the current expansion zone (i.e. a drone in border state). This ensures that the entire border remains connected to the sink at the end of the phase.

Termination When the sink receives the end-of-expansion phase message, it begins to wait for a waiting time. During this time, if it receives a message indicating the discovery of a target, then the time is reset and the sink starts waiting again from the beginning. When the waiting time is over, if no discovery message has reached the sink, then it randomly selects a nearby drone and tells him to find a path as far away from it as possible, thus ensuring the construction of a path from the sink to the edge. The drones on the path turn into irremovable state and an acknowledgement message is then sent from the border drone until the sink. Once the acknowledgement is received, the sink broadcasts a message indicating the end of the spanning phase. If at least one discovery

message has been received, the sink directly broadcasts the end-of-spanning phase message at the end of the waiting time.

Further spanning Again, except for the first spanning phase which ensures that a direct path exists between any PoI and the sink (by fully covering the initial expansion phase), the following spanning phases should seek to connect any new PoI found with an existing irremovable drones path. Thus, we must slightly modify the spanning rules as follows. A drone located on newly discovered PoI becomes irremovable and seeks to establish a path to the sink.

- If a drone that became irremovable during the previous spanning phase occupies a neighboring spot, we are done.
- If there is an occupied spot s_i closer to the sink (the closest if there is a choice), it sends a message to the drone occupying that spot that becomes irremovable and continue the process.
- If there is no occupied spot closer to the sink, i.e., we reached the previous border, a right-hand-side process is used to follow the border until a spot occupied by an irremovable drone from the previous phase is reached.

In parallel, a path to the new border is created. Finally, paths to the border of the previous phase are extended to reach the new border. Hence, at the end of the spanning phase, at least one path of irremovable drones exists between the sink and the border of the explored area.

3.3 Balancing phase

Only free drones are allowed to move in this phase. Each free drone chooses randomly a neighboring spot that goes away from the sink. This process is repeated until the drone reaches a spot occupied by a border drone. We then balance the number of drones per spot at the border so that the number of drones on two neighboring spots of the border differ by at most one. Observe that some spots of the border might be occupied by irremovable drones and that these spots might now also be occupied by some free drones.

Termination When a free drone reaches a border drone, it will follow the same process as in the expansion phase. It sends a broadcast message indicating the end of the balancing phase and starts a waiting time to terminate the phase in case of another free drone answering that it has not reach the border yet.

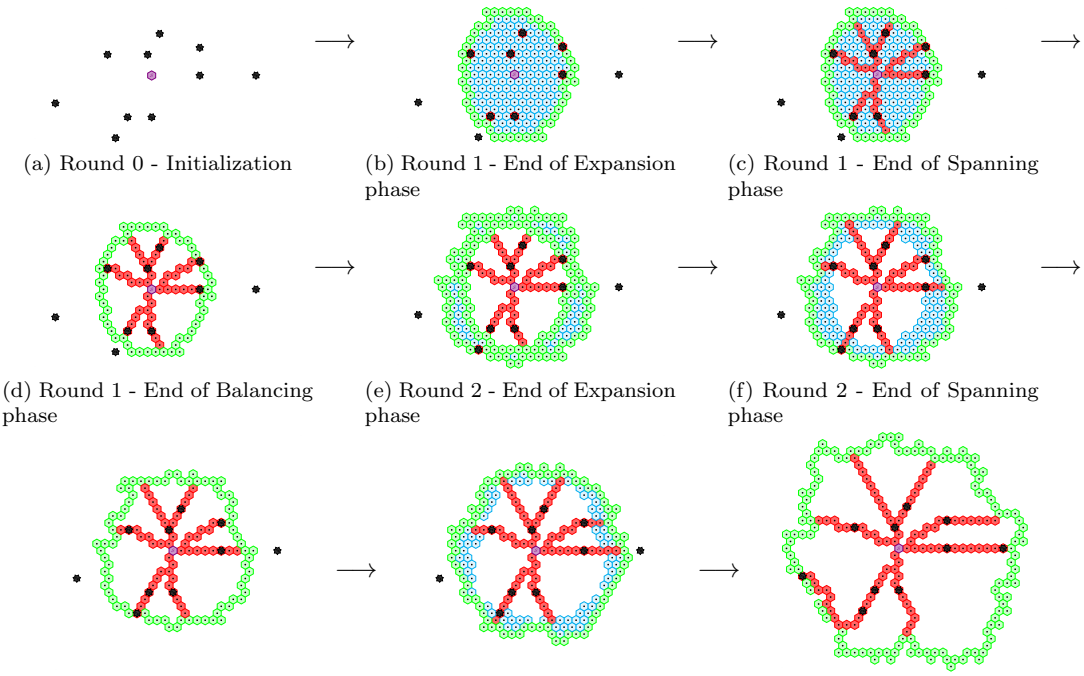
Finally, we repeat the 3 phases balancing, expansion, spanning until all drones are either irremovable and/or border.

3.4 Example

To illustrate the round-by-round operation of VESPA, Figure 2 shows the scroll on an example with 217 drones and 10 targets to cover.

On initialization, all drones are in the free state and located on the sink (*cf.*, Figure 2a). The process starts with a first expansion phase, allowing to cover the widest possible area centered on the sink. At the end of the phase, all the drones located at the edge of the expansion zone change their state to `border` (*cf.*, Figure 2b). Once the termination process is completed, the drones covering an area including a target switch to an `irremovable` state and initiate the spanning phase, in order to guarantee connectivity between the sink and the border (*cf.*, Figure 2c). The 3rd phase of VESPA then allows all drones still in a `free` state to reach the border, in order to prepare for the next expansion phase (*cf.*, Figure 2d). Figures 2e to 2i illustrate the continuation of the VESPA execution, requiring 11 additional rounds to reach the maximum expansion. Once the final round is reached (no more drone is in free state), all the drones not necessary for the connectivity of the targets with the sink return to their starting point.

Observe that Figure 2c also illustrates the output of SaS [REZN17].



(a) Round 0 - Initialization

(b) Round 1 - End of Expansion phase

(c) Round 1 - End of Spanning phase

(d) Round 1 - End of Balancing phase

(e) Round 2 - End of Expansion phase

(f) Round 2 - End of Spanning phase

(g) Round 2 - End of Balancing phase

(h) End of round 3

(i) End of last round

Figure 2: Example of execution of the VESPA algorithm, with 10 targets and 217 drones.

4 Algorithm Analysis

We now formally prove the characteristics of our distributed algorithm. For each phase, we show that, in all cases, the guarantees of connectivity, coverage and termination are respected.

4.1 Expansion phase

Theorem 1 (Connectivity)**.** *During the expansion phase, if at time t the network is connected, then at any time $t', t' > t$, the network remains connected.*

Proof. During the expansion phase, only free drones can move. Indeed, the border and irremovable drones are forced to remain in place. Let us now consider all the drones in free state. At least one drone (free or not) will remain on each spot considered (see Section 3.1). Thus, no spot occupied at time t will be unoccupied at time $t' > t$ of the expansion phase. If a drone is allowed to leave the given spot s_i, it will move to a neighboring spot. Therefore, its final distance to the current spot will be less than R_{p_i}, guaranteeing the connectivity of this drone with the remaining drones on spot s_i. Since the network is connected at time t, by induction, after each movement, the network connectivity is maintained. Then, at any time during the expansion phase, the network is connected. □

Let's now denote by $\mathcal{A}_t$ the total area covered by the drone network at time t. More formally,

$$\mathcal{A}_t = \cup_{i \in \{1,\dots,N\}} \mathcal{B}(d_i, r_s)$$

where $\mathcal{B}(d_i, r_s)$ is the ball centered at the coordinate of drone d_i, with a radius equals to r_s.

Theorem 2 (Coverage)**.** *During the expansion phase, if at time t the discovered area is equal to $\mathcal{A}_t$, then at any time $t', t' > t$, the discovered area $\mathcal{A}_{t'}$ is not decreased, that is $\mathcal{A}_t \subseteq \mathcal{A}_{t'}$.*

Proof. Since no spot can be freed by a drone covering this spot previously, the number of occupied spot after each movement is either stable or increased. In addition, in a given expansion phase, spots covered at a time t will be covered at a time $t' > t$ by definition (guarantee of connectivity and stability of positions already covered, *cf.*, Section 3.1). Also, the area covered by the drone cannot be reduced between time t and t' of expansion. This implies that $\mathcal{A}_t \subseteq \mathcal{A}_{t'}$. □

Lemma 3. *At the end of the expansion phase, there is a cycle composed only of drones in border state (i.e., the stabilized border is contiguous).*

Proof. In this proof, we separate the analysis into two steps: the first expansion phase and the subsequent spanning phases.

1. During the first expansion phase, all the drones will cover the largest continuous area around the sink. Theorem 2 shows that the maximum coverage is reached (monotonous growth until convergence, *i.e.*, when there is only one drone left per spot). In addition, the expansion rules presented in Section 3.1 ensure that any unoccupied spot, whose distance from the sink is less than the current distance of the considered drone from the sink, will be occupied as a priority in the next movement. Thus, since a spot occupied at a time t cannot be unoccupied in the same expansion phase at a time $t' > t$, and since all the UAVs start from the same starting point (*i.e.*, the sink), the coverage area at the end of the first expansion phase will be complete (*i.e.*, no unoccupied spot inside the area).

   During the termination phase of the expansion phase, only drones located at the border of the coverage area will therefore have unoccupied spots in their vicinity. They will therefore all declare themselves in a border state. Since the area is continuous, each border drone has at least two border neighbours. Otherwise, it would mean that either

   (i) he has no neighbours in a border state (*i.e.*, all his neighbours would then be surrounded by occupied spots, which is impossible, because itself would then be surrounded by neighbours, and could therefore not be in a border state, or he would have no neighbours, which contradicts the continuous zone assumption), or

   (ii) only one of his neighbours is in a border state. In the latter case, it is either that it has only one neighbour (*i.e.*, it is at the end of a drone line), in which case we consider that this same neighbouring drone plays the role of left and right neighbour of the considered drone, or an unoccupied spot would be in the vicinity of a drone which is not in border state (which is also impossible).

   Finally, by applying a right hand path algorithm, it is always possible to advance from one drone in border state to the next in border state. The area being continuous and closed (the space is considered infinite), it is therefore guaranteed to fall back on the starting drone. This therefore implies the existence of a border drone cycle, surrounding the coverage area.

2. In the subsequent expansion phases, we can apply a similar reasoning. Since all the free drones are located on the border of the previous round (*cf.*, Theorem 9). Thus, the same arguments apply, allowing to conclude that there will be a complete coverage between the border of the previous round (whose presence is guaranteed until the end of the expansion phase) and the new border defined by the drones that have moved furthest from the sink. Thus, at the end of the expansion phases, all non-irremovable drones are located in a complete annulus, surrounded by the frontier of the previous expansion phase and the new extended frontier.

□

Theorem 4 (Termination)**.** *Any expansion phase will eventually terminate.*

Proof. Each spot hosting at least one drone during an expansion phase will remain occupied until the end of the expansion phase. Also, given the rules for moving drones, no movement towards an

occupied spot towards the sink is allowed. The free drones that are not required for connectivity will therefore definitely move towards the outer edge of the covered area, until they find a free spot, or become the drone that covers the spot it has reached. In the first case, this drone will no longer be able to move, at least until another drone joins it on this spot. In the second case, the drone that was previously covering the spot will be able to move towards the outside of the area. So, the number of drones that can move according to the rules of the Section 3.1 can only decrease. By combining these properties with Theorems 1 and 2, the number of drone being able to move will reach 0 with a probability equal to 1, which concludes the proof. □

In addition, unlike the former article [REZN17], we propose a deterministic algorithm to detect the termination, allowing to ensure the direct transition from the expansion phase to the spanning phase, without relying on a timeout process or too strong a synchronization.

4.2 Spanning phase

Theorem 5 (Connectivity)**.** *During the spanning phase, if at time t the network is connected, then at any time $t', t' > t$, the network remains connected.*

Proof. Since no drone can move during the spanning phase, and thanks to Theorem 1, connectivity is maintained throughout the spanning phase. □

Theorem 6 (Coverage)**.** *For any PoI discovered during a previous expansion phase, a path composed of irremovable drones will exist between this PoI and the sink at the end of the spanning phase. In addition, at the end of the spanning phase, there is a connected path between any drone in border state with the sink.*

Proof. In this proof, we separate the analysis into two steps: the first spanning phase and the subsequent spanning phases.

1. At the end of the first expansion phase, the coverage area is related and the boundary is continuous (*cf.*, Theorem 1 and Lemma 3). Thus, all the PoI located inside the border will be covered by a drone. The latter will therefore de facto pass into an irremovable state and launch the connection procedure to the sink (*cf.*, Section 3.2). Since the coverage is exhaustive, the iterative procedure of creating a path of irremovable drones to the sink is deterministic and will terminate (the number of drones being finished and each step reducing the distance between the probe and the sink).

   In parallel, a probe will be routed in the opposite direction to the sink. Since the boundary is continuous and encompassing (*cf.*, Lemma 3), and the number of drones is finite, the probe will reach the edge of the coverage area in a finite time.

   Thus, after the first phase of spanning, each PoI will be connected by paths composed of irremovable drones, to the sink on the one hand and to the edge on the other hand.

2. During the subsequent spanning phases, exhaustive coverage around the sink is no longer guaranteed. However, this complete coverage is guaranteed between the border established during the expansion phase of the previous round and the new border established during the expansion phase of the current round. Thus, for any PoI discovered during the current round, the procedure explained in the Section 3.2 results in creating a path composed of non-removable drones between the two boundaries mentioned above, passing through the PoI (following the same reasoning as before). Then, the probe arriving at the inner edge of the area will attempt to connect to an irremovable path created in the previous round (*cf.*, Section 3.2). The previous spanning phase ensures that there is at least one irremovable drone on any external boundary of the previous round. The border being continuous (*cf.*, Lemma 3), this implies by transitivity the connection of any new PoI discovered with the sink, as well as any drone located on the current external border with the sink.

The first step represents the initialization of the proof by induction, while the following steps represent its recurrence. □

Theorem 7 (Termination)**.** *Any spanning phase will eventually terminates.*

Proof. In the spanning phase, there is no drone movement. Termination is therefore based solely on the termination of the communication and state change process. Two scenarios can occur: (i) no new PoI are discovered during this phase, so the sink will automatically trigger the balancing phase after a waiting time; (ii) if a new PoI is discovered, the sink will be informed before the timeout, and will wait again a waiting time to ensure that the propagation process to the external border is achieved, then will trigger the balancing phase. In both cases, the process will therefore end in a finite time. □

4.3 Balancing phase

Theorem 8 (Connectivity)**.** *During the balancing phase, if at time t the network is connected, then at any time $t', t' > t$, the network remains connected.*

Proof. During the balancing phase, none of the drones in border and irremovable state will move. So, since the border is continuous and encompassing (*cf.*, Lemma 3), all the drones in the free state inside the border will end up on an occupied spot on the border. Theorem 6 ensures that the other drones that haven't moved are still connected. Concerning the drones that have moved, they are connected by transitivity with the border drone located on their new location. □

The objective of this phase being to move all the free drones to the edge of the previously covered area, the study of the coverage is useless here. However, the final locations of the free drones for the next expansion phase is interesting and is addressed by the following theorem.

Theorem 9 (Drone End-phase Localization)**.** *At the end of the balancing phase, all drones that are not in irremovable state are localized on a spot on the border.*

Proof. By Lemma 3, we know that at the end of an expansion phase, there is a cycle composed only of drones in border state. Let us call the set of spots, including the sink, that is surrounded by this cycle the "*inside*". All free drones are located in the inside. Hence, if a free drone follows any path going away from the sink, it will eventually reach a spot occupied by a drone in border state, otherwise contradicting Lemma 3, i.e., the fact that the set of drones in border state forms a cycle. □

Theorem 10 (Termination)**.** *Any balancing phase will eventually terminates.*

Proof. In a similar way to the termination of the expansion phase, all the drones in free state will move away from the sink until they reach a spot located on the outer edge of the coverage area. So, since the border is continuous and encompassing (*cf.*, Lemma 3) and the number of drone in free state being finished, all these drones will end up on an occupied spot on the border. Given the process described in Section 3.3, each drone arriving on the border, and not being surrounded by border spots of a size that differs more than one, transmits a termination message and waits for a waiting time. If this drone has to move once again to balance the distribution of drones along the border, it will then relaunch its probe message. Then, as the number of moving UAVs is finite and decreasing irrevocably, convergence will be achieved in a finite time. The last drone to move will then send a termination message and after a waiting time, all drones will switch to the next expansion phase. □

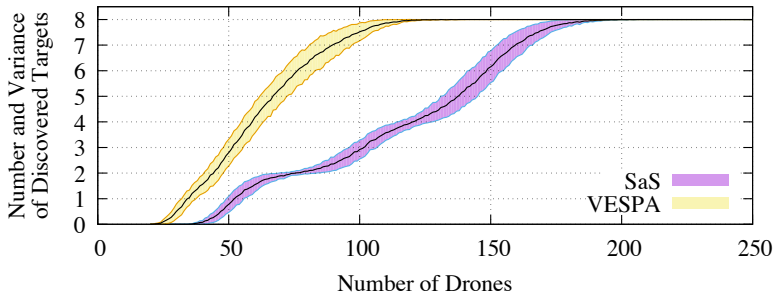


Figure 3: Average number of discovered targets, enveloped by variance value.

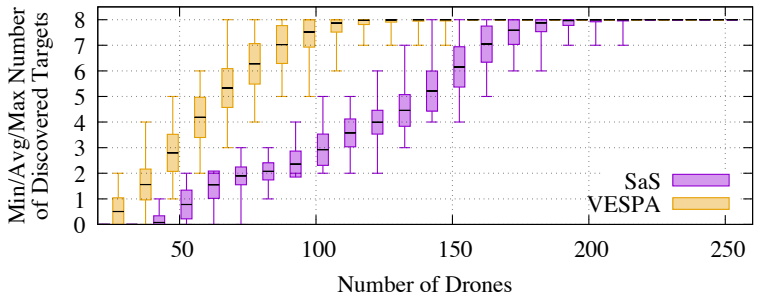


Figure 4: Whisker-box of number of discovered targets, with min, max, average and standard deviation values.

5 Simulations

We simulated the distributed pattern of our algorithm in the following scenario. One sink is located in the center of the area to monitor. We randomly deploy targets around the sink with unknown coordinates to the drones and the sink, and we try to discover them with a set of drones. We run the algorithm on thousands of random topologies, making the number of targets vary from 1 to 30, and the number of drones from 10 to 250. As in [REZN17], we set the drones communication range R_{p_i} to 20m, constant C to 100 and ϵ to 20.

We validate VESPA through extensive simulations and compare its performances with the Spread and Shrink (SaS) algorithm [REZN17] in terms of number of discovered targets, and number of drones needed to cover and ensure connectivity to the targets.

5.1 Validation of VESPA

To ensure a fair comparison with SaS, we first define the following scenario used in [REZN17]. In this case, 8 targets are deployed on the area at coordinates:

$$\{(0,85);(0,-85);(85,0);(-85,0);(75,75);(75,-75);(-75,75);(-75,-75)\},$$

assuming that the sink is located at coordinate $(0,0)$. We run VESPA on this configuration with various number of drones and depict the number of discovered PoI in Figure 3. The average number of discovered PoI in function of the number of drones used is enveloped by the variance value obtained from all the tested topologies. We can see from the figure that the discovery of all the targets depends on the number of drones. Indeed, the more drones, the larger monitored area, and so the larger number of discovered targets. Since VESPA guarantees connectivity and coverage at each stage of the process, the number of free drones allowing to move along and enlarge the monitored area decreases with time.

However, even with very few drones (20, see Figure 4), it is possible to discover at least one point of interest. Figure 4 also shows that with 70 drones it is possible to find all the PoI. With 160 drones and above, all the PoI are always discovered during all the simulations, and with 120 drones, all PoI are discovered more than 99% of the time. This behaviour shows that the randomness of the movement decisions in the algorithm is managed and controlled by VESPA.

As expected, our multi-stage algorithm allows to discover more targets with less drones than SaS. We indeed use the full potential of the available drones to discover as many targets as possible. With the same number of drones at the beginning, our algorithm will cover between two to three times more PoI on average than SaS (e.g. 3 times more with 50 drones, and 2.5 times more with 100 drones, see Figure 3).

In return, VESPA will repeat expansion-spanning-balancing phases to discover new targets at the edge of the previous round of these 3 phases, if free drones are still available. In Figures 5 and 6 , we present the number of repetitions of the 3 phases of VESPA in function of the number of drones. The number of rounds decreases when the number of drones increases. The expansion phase allows to cover a larger region and to discover more targets. Indeed, when the number of drones is large enough, then all the targets are found in the first round (with more than 180 drones, see Figure 6). With small number of drones, the number of 3-phase rounds can vary, depending on the targets location, leading to a large variance value, but the average number of rounds usually stays lower than 6.

5.2 Performance evaluation

We now compare the results on random topologies where the targets are randomly deployed on the monitored area.

Figure 7 depicts the number of needed drones to cover all the PoI. The number of required drones increases with the number of targets for small values: 45 drones must spread the area to cover one target, while 90 drones are required to discover 10 targets on average. This phenomenon is also found in SaS behavior, while the number of needed drones is more important (twice as many drones for 5 PoI). When the number of targets is large, the number of drones is always below 140 drones on average. Indeed, VESPA computes connected path gathering several targets on the same path, therefore limiting the number of additional drones required in the solution.

This is confirmed in Figure 8 where the number of used drones at the end of the algorithm, that are kept to ensure the connectivity between the found targets and the sink, is presented in function of the number of targets. The figure shows that less than 10 drones are needed on average to cover a target (depending on its distance to the sink), and less than 70 drones can cover and connect 30 targets. Compared to SaS (see also Figure 13 in [REZN17] for the scenario of previous section), the number of drones in the solution is more important for small number of PoI because longer paths can be computed by VESPA during the several spanning phases, but for more than 10 targets the number of used drones becomes equivalent for both algorithms. This validates the spanning and balancing phases of VESPA that manages the size of the solution.

VESPA reduces by 40% the number of drones required to discover the same number of PoI than SaS, while optimizing the size of the connected paths to the sink in the solution.

6 Conclusion

In this paper, we presented a distributed algorithm using only one-hop information of the drones, to discover targets with unknown location and ensure connectivity between them and the sink, auto-organizing in a multi-hop aerial wireless network. We prove that connectivity and coverage are preserved during all stages of our algorithm, and that all phases eventually terminate. We evaluate the algorithm performances through simulations and shown its effectiveness compared to an existing algorithm from the literature both in terms on number of drones used in the solution and number of discovered targets. In the next steps of this work, we would like to evaluate our algorithm in an event-based simulator and extend the algorithm to deal with target mobility.

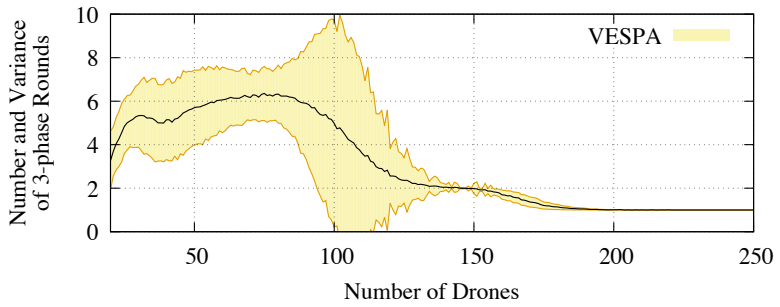


Figure 5: Average number of 3-phase rounds, enveloped by variance value.

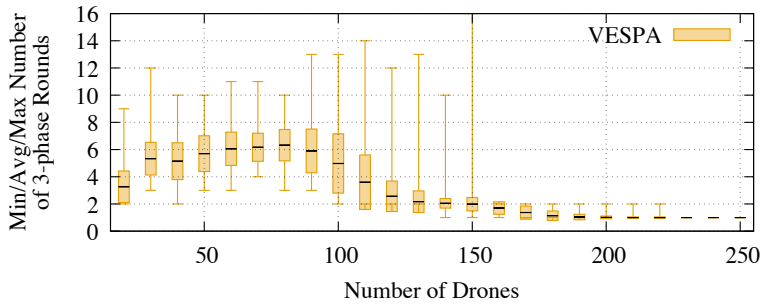


Figure 6: Whisker-box of number of 3-phase rounds, with min, max, average and standard deviation values.

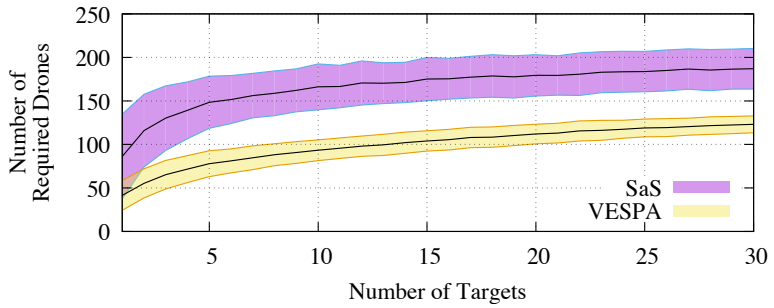


Figure 7: Average number of drones required to discover all the randomly positioned targets.

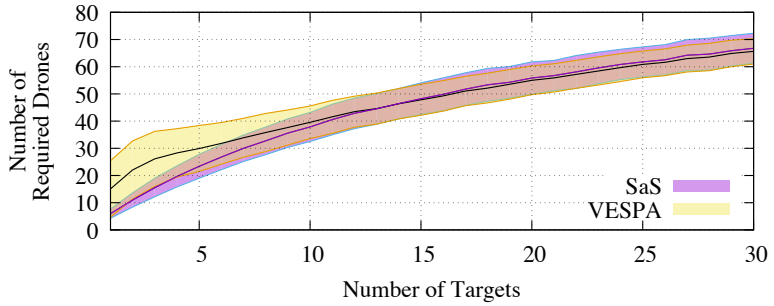


Figure 8: Average number of drones at the end of the algorithm, to cover and connect all the discovered targets.

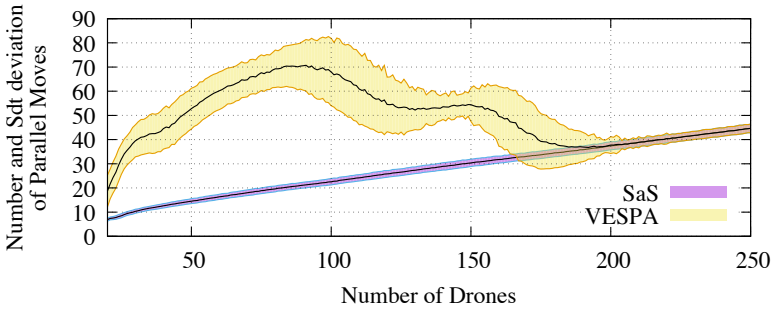


Figure 9: Average number of parallel moves, enveloped by standard deviation value.

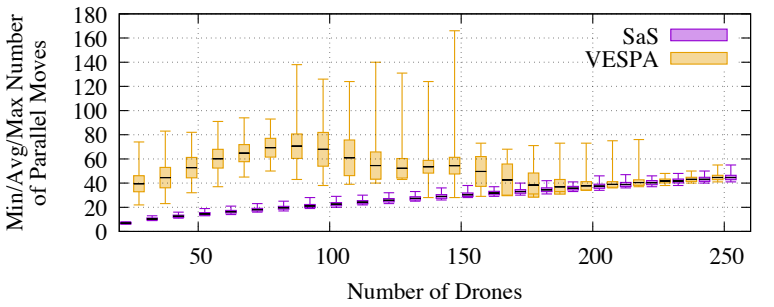


Figure 10: Whisker-box of number of parallel moves, with min, max, average and standard deviation values.

Acknowledgements

This work has been partially supported by action RESCOM of GDR ASR of CNRS, the French National Research Agency (ANR), through the UCA$^{\text{JEDI}}$ Investments in the Future project with the reference number ANR-15-IDEX-01.

References

[CGR19] Christelle Caillouet, Frédéric Giroire, and Tahiry Razafindralambo. Efficient data collection and tracking with flying drones. *Ad Hoc Networks*, 89:35 – 46, 2019.

[EFGR19] Jean-Matthieu Etancelin, Andr Fabbri, Frédéric Guinand, and Martin Rosalie. Dacyclem: A decentralized algorithm for maximizing coverage and lifetime in a mobile wireless sensor network. *Ad Hoc Networks*, 87:174 – 187, 2019.

[HS18] Hailong Huang and Andrey V. Savkin. Towards the internet of flying robots: A survey. *Sensors*, 18(11), 2018.

[HSDH19] Hailong Huang, Andrey V. Savkin, Ming Ding, and Chao Huang. Mobile robots in wireless sensor networks: A survey on tasks. *Computer Networks*, 148:1 – 19, 2019.

[LGF18] Zhong Liu, Xiaoguang Gao, and Xiaowei Fu. A cooperative search and coverage algorithm with controllable revisit and connectivity maintenance for multiple unmanned aerial vehicles. *Sensors*, 18(5), 2018.

[MHS17] Shaimaa M. Mohamed, Haitham S. Hamza, and Iman Aly Saroit. Coverage in mobile wireless sensor networks (M-WSN): A survey. *Computer Communications*, 110:133 – 150, 2017.

[PGZR16] L. Di Puglia Pugliese, F. Guerriero, D. Zorbas, and T. Razafindralambo. Modelling the mobile target covering problem using flying drones. *Optimization Letters*, 10(5):1021–1052, 2016.

[REZN17] Tahiry Razafindralambo, Milan Erdelj, Dimitrios Zorbas, and Enrico Natalizio. Spread and shrink: Point of interest discovery and coverage with mobile wireless sensors. *Journal of Parallel and Distributed Computing*, 102(Supplement C):16 – 27, 2017.