



# Energy-efficient and fault-tolerant drone-BS placement in heterogeneous wireless sensor networks

Fatih Deniz<sup>1</sup> · Hakki Bagci<sup>1</sup> · Ibrahim Korpeoglu<sup>2</sup> · Adnan Yazıcı<sup>3</sup>

Accepted: 31 October 2020 / Published online: 9 November 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

This paper introduces a distributed and energy-aware algorithm, called Minimum Drone Placement (MDP) algorithm, to determine the minimum number of base stations mounted on resource-rich Unmanned Aerial Vehicles (UAV-BS), commonly referred to as drone-BS, and their possible locations to provide fault tolerance with high network connectivity in heterogeneous wireless sensor networks. This heterogeneous model consists of resource-rich UAV-BSs, acting as gateways of data, as well as ordinary sensor nodes that are supposed to be connected to the UAV-BSs via multi-hop paths. Previous efforts on fault tolerance in heterogeneous wireless sensor networks attempt to determine transmission radii of the sensor nodes based on the already deployed base station positions. They assume that the base stations are stationary and arbitrarily deployed regardless of the position of the sensor nodes. Our proposed MDP algorithm takes into account the desired degree of fault tolerance and the position of ordinary sensor nodes to determine the optimal number of UAV-BSs and their locations. The MDP algorithm consists of two steps. In the first step, each sensor node chooses low-cost pairwise disjoint paths to a subset of candidate UAV-BSs, using an optimization based on the well-known set-packing problem. In the last step, depending on the desired degree of fault tolerance, MDP chooses a subset of these UAV-BS candidates using a novel optimization based on the well-known set-cover problem. Through extensive simulations, we demonstrate that the MDP achieves up to 40% improvement in UAV-connected lifetimes compared to a random and uniform distribution of UAV-BSs.

**Keywords** Fault-tolerance · Energy efficiency · Heterogeneous wireless sensor networks · Unmanned aerial vehicles (UAVs) ·  $k$ -connectivity · Set cover problem

## 1 Introduction

Wireless sensor networks (WSNs) consist of many tiny, low-cost sensor nodes that can perform detection, processing, and transmission of data over wireless channels. After deployment, sensor nodes collaborate in a distributed and self-organized fashion to set up a network, typically in an environment without any infrastructure. These networks are used in numerous applications such as border surveillance [7], underwater habitat monitoring [13] and health-care applications [1]. Because the sensor nodes are battery-powered and have limited energy, it is of very important to prolong network lifetime in WSNs. One approach to extend network lifetime is to form a heterogeneous WSN architecture with supernodes which act as alternative gateways to the monitoring center of a WSN [6]. This means that it is sufficient for the sensory data to reach one of these

---

✉ Fatih Deniz  
fatih.deniz@hacettepe.edu.tr

Ibrahim Korpeoglu  
korpe@cs.bilkent.edu.tr

Adnan Yazıcı  
yazici@ceng.metu.edu.tr

<sup>1</sup> Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

<sup>2</sup> Department of Computer Engineering, Bilkent University, Ankara, Turkey

<sup>3</sup> Department of Computer Science, Nazarbayev University, Nur-Sultan, Kazakhstan

supernodes. A heterogeneous WSN with supernodes is known to be more reliable and has a longer network lifetime than homogeneous counterparts without supernodes. Yarvis et al. [32] reported that heterogeneity can triple the average delivery rate and provide a five-fold increase in network lifetime if supernodes are deployed carefully.

Another critical issue in WSNs is fault-tolerance. Due to the error-prone nature of sensor nodes and wireless communication, faults occur very frequently in WSN applications. Inhospitable and harsh environmental conditions, where sensor nodes are usually deployed, and the multi-hop communication also increase the risk of faulty incidents. Therefore, for particularly mission critical applications that require high reliability, fault tolerance emerges as a very critical issue.

In the literature there are some studies that address fault-tolerance, connectivity and heterogeneity at the same time [5, 9, 14]. These approaches mainly concentrate on the transmission powers of the sensor nodes. Cardei et al. [9] address fault-tolerant topology control in heterogeneous WSNs that consists of a large number of ordinary sensor nodes and a few resource-rich supernodes. They introduce  $k$ -vertex supernode-connectivity where every sensor node is connected to supernodes by at least  $k$  disjoint paths. There, the goal is to minimize total transmission power of the sensor nodes while maintaining  $k$ -vertex supernode-connectivity. Bagci et al. [5] present a distributed algorithm, called the Disjoint Path Vector (DPV) algorithm, for the same problem with a superior performance. More recently, in order to address the dynamic nature of WSNs, Deniz et al. [14] improve the DPV algorithm where sensor nodes dynamically change their transmission powers to adapt the network topology to the node failures, and refer their algorithm as the Adaptive Disjoint Path Vector (ADPV) algorithm. All of these approaches share a common drawback, which they all assume that a certain number of supernodes are initially deployed regardless of where the sensor nodes are located.

In WSNs, sensor nodes can be carefully placed at engineered positions or thrown in bulk at random positions [20]. The placement of sensor nodes obviously has a significant effect on the resource management of WSNs [15]. However, it is not always possible to place thousands of tiny sensor nodes at known locations. Firstly, generally the number of sensor nodes to be deployed is very large and secondly, the application environment is usually not fully accessible. For these reasons, especially for regular sensor nodes, it is usually preferable to sacrifice a little more resources than working on the placement issue. For resource-rich supernodes, on the other hand, placement at engineered positions is less expensive due to their smaller numbers and adds huge advantage to the connectivity and fault-tolerance properties of the network. In this study, we

assume that ordinary sensor nodes are launched in bulk at random locations and we consider determining the minimum number of supernodes required and possible locations based on these randomly distributed sensor nodes. Most of the existing studies on placement of sensor nodes aim to minimize the number of sensor nodes required, while preserving application-specific constraints such as coverage, connectivity or fault-tolerance [23, 42]. We also have the constraints of preserving connectivity, fault-tolerance factors and at the same time balance the relaying among the sensor nodes.

For the placement of supernodes, we use drones (UAVs) mounted on supernodes, which have recently been widely studied [10, 21, 34]. Due to their inherent attributes such as rapid deployment, mobility and flexibility, they can be easily placed at predefined positions, recharged or replaced as needed. It has been shown that correct placement of UAV-BSs can effectively prolong the lifetime of mobile ad hoc networks [3].

In this paper, we propose a novel distributed and energy-aware algorithm, namely the Minimum Drone Placement (MDP) algorithm, to locate the minimum number of UAV-BSs mounted supernodes in a network of sensor nodes using an optimization based on the well-known set-cover problem, so that the resulting topology becomes  $k$ -vertex UAV-connected.

The remainder of the paper is organized as follows: In Sect. 2, we discuss related studies and give some background information. We present our proposed UAV-BS placement algorithm in Sect. 3. We present the results of our experiments in Sect. 4 and finally, we give our conclusions in Sect. 5.

## 2 Related work

The deployment phase is an important aspect of WSNs. To ensure certain levels of connectivity and fault tolerance for sensor nodes, deployment solutions have been widely used. In this section we provide a brief overview of some prominent recent work on multiple sink, UAV-BS and relay node placement strategies in WSNs.

### 2.1 Multiple sink placement

Due to the scarcity of energy supply of WSNs, there is a need to design energy-efficient architectures and optimize energy consumption. In small networks, sensors can send their data directly to the sink node. In larger networks, multi-hop communication is required and sensors are also used for data relay. In particular, sensors within one-hop distance from the sink have to relay data for other sensor nodes and most of their energy is devoted to data relaying.

Therefore, in order to utilize energy efficiently, it is important to shorten number of hops a packet has to travel until it reaches the sink and to balance the relaying job between sensor nodes. Deploying multiple sinks is a method to shorten these hop counts, balance relaying job and prolong network lifetime [8, 16]. Multiple sink usage also improves various network performance including average data delivery latency [46] and system throughput [38].

WSNs experience failure problems due to various factors such as power depletion, environmental impact, radio interference, dislocation of sensor node and collision. The problem of missing sensor node and communication link errors are inevitable in WSNs. In addition to failures of sensor nodes, sinks may also fail due to different reasons such as hardware failure, software failure or intentional attacks [26, 39]. Therefore, fault-tolerance is an important design factor that should be considered during the deployment of sink nodes. Deployment solutions are mainly designed to ensure coverage and connectivity requirements of WSNs. In multiple sink placement problem, the aim is to ensure network's connectivity while keeping the maximum hop-count constraint. Together with fault-tolerance requirements, sinks are placed so that all the sensor nodes are connected to multiple sinks or connected to the set of sinks with fault-tolerant paths. In this way, multiple sink placement solutions support fault-tolerance by ensuring the existence of alternative routes to the sinks when failure occurs.

In general, finding minimum number of required sink nodes is NP-hard [40] and determining locations for these sinks is NP-complete [8]. Since optimal sink placement has proved to be NP-complete, several sub-optimal heuristics were proposed with the objective of balancing energy consumption [45], reducing packet delivery latency [46], minimizing transmission radius of sensor nodes [25] and meeting fault-tolerance requirements [43]. Xu and Liang [45] and Oyman and Ersoy [35] aim to minimize number of deployed sink nodes, while other mentioned approaches get number of sink nodes as an input and deploy sinks into precomputed locations. Sitanayah et al. [43] and Xu and Liang [45] are the only two approaches that take maximum hop-count into consideration and Sitanayah et al. [43] is the only approach that considers fault-tolerance in the multiple sink placement problem. None of the multiple sink placement approaches take into account the residual energy levels of the sensor nodes and none of the approaches mentioned are distributed.

In [37], Poe and Schmitt discuss four different sink placement strategies. These are: Random Sink Placement (RSP), Geographic Sink Placement (GSP), Intelligent Sink Placement (ISP) and Genetic Algorithm-based sink placement (GASP). Among these, GSP and GASP are most

efficient strategies. More recently, Pardesi and Grover [36] improve GSP and propose a new strategy, namely I-GSP, which divides the network in concentric circular rings around the central circular region. In [11], Dandekar and Deshmukh propose an algorithm named Optimal Multiple Sink Placement (OMSP) that divides the network into clusters for the given number of sink nodes and calculates locations for them using particle swarm optimization. Das et al. [12] propose two algorithms, namely Candidate Location with Minimum Hop (CLMH) and Centroid of the Nodes in a Partition (CNP), and they also compare their results with Geographic Sink Placement (GSP) strategy. Das et al. assume a partitioned network, where partitions is given as an input and try to find locations for the cluster heads that will minimize transmission delay and extend network lifetime.

In this study, we present an energy-aware and distributed solution to determine sink positions and form a fault-tolerant network topology. In terms of the intended network topology, our aim differs from all existing solutions. Only one of the existing multiple sink placement solutions addresses fault-tolerance. However, that study only considers sink failures, but not sensor node failures, which is obviously much more probable and aims to connect each node to multiple sinks [43]. Greedy-MSP and GRASP-MSP are the two approaches mentioned in [43] to calculate multi-hop paths to multiple sinks. In this study, we consider both sink and sensor node failures and also take residual energy levels of the sensor nodes into account to calculate number of required sink nodes and their locations and achieve a more robust network topology.

The use of UAV-BSs has been realized as a promising addition to the conventional WSNs [24, 34] and make a valuable contribution to the deployment of sink nodes. They can be used for data collection from the sensor nodes, and also support connectivity and fault-tolerance properties [47]. Owing to their inherent attributes like rapid deployment and flexibility, they can be placed to predefined positions, recharged or replaced as needed. They can be used to serve as assisting relays to improve the connectivity of the WSNs [29]. When placement of the sink nodes is static, it leads to performance instabilities and fortunately, UAVs can provide an effective leverage to improve the environment adaptation in WSNs. This paper provides a UAV-BSs deployment plan, while minimizing number of them, it also ensures certain levels of connectivity and fault-tolerance.

## 2.2 Relay node placement in WSNs

Another well-studied subject for the placement problem in WSNs is the placement of relay nodes. Relay nodes in WSNs are generally nodes with higher transmitting power.

The goal in the relay node placement is to place minimum number of relay nodes in a region where sensor nodes are already deployed so that the resulting topology is connected and/or fault-tolerant. In the literature there are several such studies [31, 33]. There are also some recent studies such as [17, 41] to determine the minimum number of relay nodes to maintain multi-hop paths between every pair of sensor nodes.

As stated by Liu et al [30], existing solutions to obtain fault-tolerance aim to obtain  $k$ -vertex connectivity between any two sensor nodes and achieve that using least number of relay nodes. Although such optimization is a very challenging work and proven to be NP-hard [28], it does not conform with the general objective of WSN applications. For WSN applications, general objective is to send the received data to the sink nodes. Therefore, it is more convenient to have fault-tolerant paths from sensor nodes to the sinks. Our aim differs from the existing solutions at this point. We aim to provide fault-tolerant disjoint paths between every sensor node to the set of supernode mounted UAV-BSs, but not between every pair of sensor nodes. In other words, we aim to place UAV-BSs, where the traffic is destined to, so that network becomes  $k$ -vertex UAV-connected.

Recently, Azharuddin and Jana [4] proposed a method to place minimum number of additional relay nodes to achieve fault-tolerance for heterogeneous WSNs. However, they do not allow sensor-to-sensor communication and thus locate relay nodes in such a way that there is at least  $k$  relay nodes within the transmission range of each sensor node. Our approach, on the other hand, allows sensor-to-sensor communication and hence potentially decreases the required number of UAV-BSs to be placed to achieve a certain degree of fault-tolerance. More recently, Hanh et al. [17] proposed two heuristics, namely PGA and SGA, that focus on minimizing the number of sensor and relay nodes in the existence of multiple sink nodes. GRASP-ARP [42] is another relay node placement approach that aims to deploy minimum number of relay nodes, so that in the resulting network each sensor node is connected to the sink nodes by at least  $k$  disjoint paths. In terms of the resulting network topology, GRASP-ARP has similar intentions with our approach, however being a centralized approach restricts its usage in WSN applications.

### 3 Minimum drone placement algorithm

In this section, we present our novel distributed and energy-aware algorithm, MDP, which aims to determine the minimum number of UAV-BSs and their possible locations to construct a  $k$ -vertex UAV-connected network topology. In this section, we first discuss the associated

network model and define the problem, then describe our algorithm in detail.

#### 3.1 Network model and problem definition

Consider a mission critical border surveillance system that is integrated with a two-tiered heterogeneous WSN. In this network, sensor nodes are randomly and uniformly deployed on the border line and initially there are no supernode mounted UAV-BSs as shown in Fig. 1(a). After MDP algorithm determines UAV-BS locations, UAV-BSs are deployed as shown in Fig. 1(b). When sensor nodes sense a potential intrusion activity, they inform the base station by forwarding the data to the UAV-BSs and base stations decide whether to take an action or not. Since it is common to lose some sensor nodes because of energy depletion, harsh environmental conditions or hostile activities of intruders, it is desired for every sensor node to have more than a certain number of independent paths to

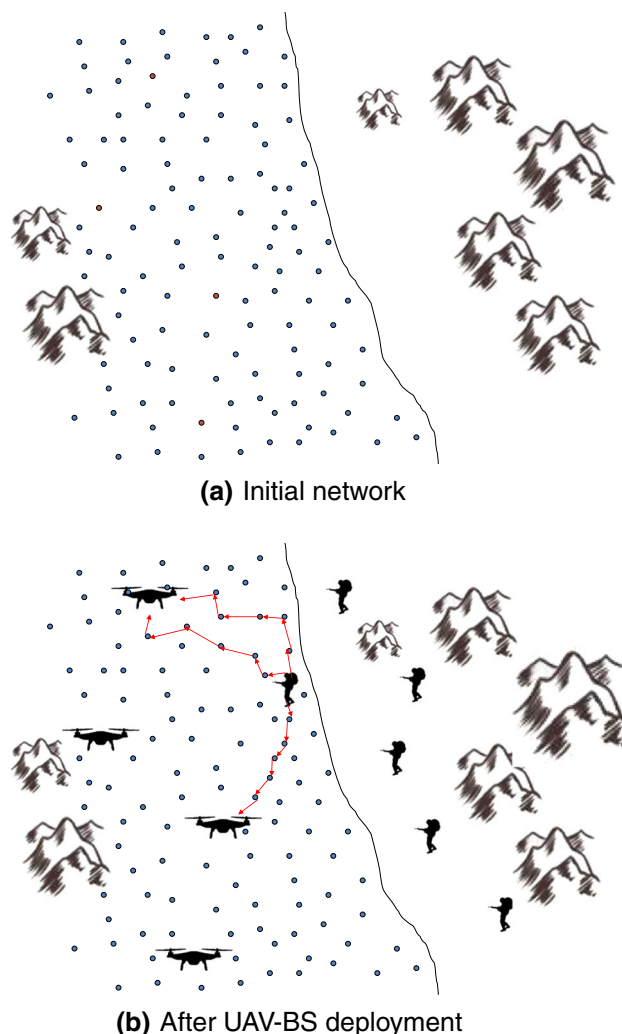


Fig. 1 Sample scenario

the set of UAV-BSs. In this scenario, fault-tolerance is achieved by forwarding the sensed data to the set of UAV-BSs using three disjoint paths as illustrated in Fig. 1(b).

This network model is first described in [9], and also used by [5, 14]. In this model, network consists of  $M$  supernodes and  $N$  sensor nodes that are randomly distributed in the 2D plane so that  $M \ll N$ . In this network model, a WSN is said to be  $k$ -vertex supernode-connected if removal of any  $k - 1$  sensor nodes does not disconnect any sensor node from all the supernodes, that is, each sensor node is still connected to some supernodes.

In the previous studies, number of supernodes was a predefined constant and they were deployed at known locations. In this study, our objective is to determine minimum number of supernode mounted UAVs ( $M$ ) and their possible locations to maintain a  $k$ -vertex UAV connected network topology.

More formally, we represent initial network topology with an undirected weighted graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E = \{v_i, v_j \mid \text{dist}(v_i, v_j) < R_{max}\}$  is the set of edges, where  $\text{dist}(v_i, v_j)$  defines the distance between nodes  $v_i$  and  $v_j$  and  $R_{max}$  is the maximum transmission range. Given that  $G = (V, E)$  is  $k$ -vertex supernode-connected with a set  $C \subset V$  of candidate supernode vertices and a set  $N \subset V$  of sensor node vertices, such that  $C \cap N = \emptyset$  and  $C \cup N = E$ . Our aim in this study is to find a set of vertices  $M \subset C$  such that  $G(V, M \cup N)$  is still  $k$ -vertex supernode-connected and number of elements in  $M$ ,  $|M|$ , is minimum.

### 3.2 Minimum drone placement (MDP) algorithm

The MDP algorithm involves two steps. In the first step, the algorithm assumes all sensor nodes has the capabilities of supernodes and treats them as candidate UAV-BSs and computes paths from sensor nodes to the set of candidate UAV-BSs. In this way, sensor nodes choose a set of other sensor nodes that they prefer to be a UAV-BS. Later, the second step chooses minimum number of sensor nodes to be UAV-BSs, so that network gets  $k$ -vertex UAV connected.

MDP is a distributed algorithm that determines possible locations for the UAV-BSs by message transmissions among sensor nodes and requires only one-hop neighborhood information. MDP is also an energy-aware solution that considers residual battery power levels of the sensor nodes. The MDP algorithm consists of two main steps:

1. Collecting path information and determining preferred paths,
2. Determining UAV-BS locations.

During the first step, MDP uses a modified version of ADPV algorithm. Unlike ADPV, MDP's aim is not to

determine transmission powers of the sensor nodes, but to determine possible locations for the UAV-BSs. For this aim, MDP has a simple, yet effective modification over ADPV algorithm. During the first phase, MDP determines preferred paths for each sensor node, which are used as alternative paths for the aim of restoring connectivity in the ADPV algorithm [14].

WSNs are characterized by their dynamic nature and this requires them to be adaptive to the environmental and topological changes. ADPV [14] is an adaptive approach that adapts the network topology to the node failures by dynamically adjusting the transmission powers of the sensor nodes. The main objective of ADPV is to prolong supernode-connected lifetime of the network.

ADPV consists of two phases: initialization and connectivity restoration. During the initialization phase ADPV determines alternative paths to be used during connectivity restoration phases. Whenever a node failure breaks  $k$ -vertex supernode connectivity, ADPV restores connectivity using these alternative paths. Therefore, computing high-quality alternative paths lies at the core of the ADPV algorithm. For determining alternative paths efficiently, ADPV proposes a novel optimization based on the well-known set-packing problem. The set-packing problem asks for the maximum number of pairwise disjoint sets among a family of sets. ADPV first determines a large set of alternative paths with minimum cost and then chooses a diverse subset, as being disjoint with others, using the maximum set-packing procedure. ADPV utilizes alternative paths to be pairwise disjoint, so that, a failed node can remove at most one path from the alternative paths. Experimental results indicate that using these alternative paths ADPV can restore connectivity several times and significantly prolong network lifetime.

For the first step of MDP, the method of initiating MDP differs from the ADPV algorithm. The ADPV algorithm starts with the 'Init' messages initiated by the supernodes, and in the case of MDP the goal is to determine UAV-BS locations and initially there is no UAV-BS. The MDP algorithm is initiated by the 'Init' messages transmitted by all ordinary sensor nodes. In this way, each sensor node behaves like a UAV-BS and lets sensor nodes compute disjoint paths intended for itself. Since ADPV determines minimum cost disjoint paths to the set of supernodes, MDP determines minimum cost disjoint paths to a subset of sensor nodes. In this way, by ignoring the intermediate sensor nodes and simply considering the destination nodes, each sensor node chooses a subset of sensor nodes that it prefers to be a UAV-BS. Algorithm 3.1 shows the details of collecting path information and determining preferred paths. The variables used in the pseudo code are defined in Table 1.

**Table 1** MDP notations

$N$	Set of sensor nodes
$P$	Preferred paths of a sensor nodes
$t$	Destination node, source node pairs of preferred paths
$T$	Union of destination node, source node pairs of preferred paths
$k$	Disjoint connectivity degree
$M$	Set of chosen UAV-BSs
$U$	List of uncovered sensor nodes
$s$	Chosen sensor node
$m, n, x, y$	Variables referencing sensor nodes

**Algorithm 3.1** Path Information Collection in MDP**Input:**  $I, L$ **Output:**  $t$ 

```

1:  $t \leftarrow \emptyset$ ;
2:  $P \leftarrow \emptyset$ ;
3: for all received PathInfo message  $I$  do
4:    $P' \leftarrow \text{MAXSETPACKING}(P \cup I.P)$ ;
5:   Sort( $P'$ );
6:    $P' \leftarrow \{p_i \in P' \mid i \leq L\}$ ;
7:   if  $\text{Cost}(P') < \text{Cost}(P)$  then
8:      $P \leftarrow P'$ ;
9:     Transmit PathInfo( $P$ );
10:  end if
11: end for
12:  $t \leftarrow \{m, n \mid \exists p \in P \text{ where } m = p.\text{Dst}, n = p.\text{Src}\}$ ;

```

When a sensor node receives a ‘PathInfo’ message containing a set of preferred paths, which are logical paths that are used for informing neighbor nodes about which paths they can use over the sender node, from one of its neighbors, it first calculates the union of the current preferred path sets of the sender and receiver. It then executes the maximum set packing procedure defined in [14] on this union to eliminate paths that have too many sensor nodes in common. This procedure is the optimization version of the well-known set packing (SP) problem and requests the maximum number of pairwise disjoint paths in a family of paths. As a result of this procedure, we maintain a very small set of preferred paths, but at the same time the quantity of those paths is large and diverse enough to find an adequate number of candidate UAV-BS locations for each sensor node.

The procedure then determines a set of candidate preferred paths  $P'$  as the first  $L$  minimum-cost paths. The cost

of a path is determined by its expected lifetime, which is equal to the expected lifetime of the first node to die on the path, and the altitude constraint of the destination node. This is why the estimation of the lifetime of each sensor node on a path is at the heart of this approach and the lifetime of a path made up of nodes  $n_0, n_1, \dots, n_l$ , in which  $n_0$  is the starting sensor node and  $n_l$  is the destination node is calculated as  $\min_{0 \leq i < l} \{b_i / P_f(d_i)\}$ , where  $b_i$  denote the residual energy level of sensor node  $n_i$ ,  $d_i$  denotes the distance between  $n_i$  and  $n_{i+1}$  for each  $0 \leq i < l$  and the power consumption function  $P_f$  is defined in Sect. 4.2. In order to apply the altitude constraint, we use the Pythagorean theorem to calculate hypotenuse, using the distance between  $n_{l-1}$  and  $n_l$ , and the altitude constraint of  $n_l$ .

The MDP algorithm chooses a set of preferred paths such that the minimum lifetime of those paths is maximized. When a sensor node chooses a path, it means that it prefers that a UAV-BS exists at the end of that path and it prefers to send its collected data along that path. If there is

an update in the preferred paths of a sensor node, it also notifies its neighbors and the algorithm ends when there is no more update and all sensor nodes determine their preferred path sets.

### 3.3 Determining UAV-BS locations

The second phase makes use of the preferred paths calculated by each sensor node during the first phase and determine a subset of sensor nodes that will cover all of the sensor nodes. Besides the preferred paths, this phase requires two additional entries, which are the list of sensor nodes and the expected level of fault tolerance ( $k$ ). It has a single output, which is the locations of UAV-BSs. Since only the source and destination nodes used in the second phase, as the first step, MDP ignores all intermediate nodes in the preferred paths. Considering all the paths in the preferred list are pairwise disjoint, selecting any  $k$  elements from this set and choosing the destination nodes as UAV-BSs will enable a  $k$ -vertex UAV connectivity.

sensor nodes, MDP uses an optimization based on the well-known set-cover problem. The set-cover problem is one of the Karp's 21 NP-complete problems [22] and the optimization version of this problem, which we use in our case, is an NP-hard problem [27]. Therefore, we use a well-known greedy heuristic to determine the smallest subset that covers all the sensor nodes. This heuristic works in iterations and in each iteration it asks for the node that contains the greatest number of uncovered elements. It adds the chosen node to the result and removes all the nodes it covers from the uncovered nodes list. Iterations terminate when no item remains in the uncovered list. This greedy approach guarantees coverage of all nodes and also approximately guarantees the minimum number of UAV-BSs [44]. In the case of set-packing heuristic, which is used for constructing preferred paths, there was no guarantee for the optimality of the result, but for the set-cover heuristic there is.

For example, consider the universe  $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  and a family  $\mathcal{S} = \{S_a, S_b, S_c, S_d\}$ , consist-

---

#### Algorithm 3.2 Determining UAV-BS Locations (DSL)

---

**Input:**  $N$  and  $T$  and  $k$

**Output:**  $M$

```

1:  $M \leftarrow \emptyset$ ;
2:  $U \leftarrow \emptyset$ ;
3:  $i \leftarrow 0$ ;
4: while  $i \leq k$  do
5:    $U \leftarrow U \cup N$ ;
6:    $i \leftarrow i + 1$ ;
7: end while
8: while  $U \neq \emptyset$  do
9:    $s \leftarrow \text{MAXINTERSECTION}(U, T)$ ;
10:   $M \leftarrow M \cup \{s\}$ ;
11:   $U \leftarrow U - \{x \mid x \in N, (s, x) \in T\}$ ;
12:   $T \leftarrow T - \{(s, y) \mid y \in N, (s, y) \in T\}$ ;
13: end while

```

---

Algorithm 3.2 shows the details of the second phase. The variables used in the pseudo code are defined in Table 1. Since MDP aims to determine a set of destination nodes that will cover all the sensor nodes at least for  $k$  times, MDP uses a list of uncovered elements that initially contains every node for  $k$  times. To determine the smallest subset of the sensor nodes that will cover all

ing of the subsets:  $S_a = \{1, 2, 3, 4\}$ ,  $S_b = \{4, 5, 6\}$ ,  $S_c = \{5, 6, 7, 8\}$ ,  $S_d = \{9, 10\}$ . When all the elements are needed to be covered once, first iteration of maximum set cover heuristic chooses one of the subsets  $S_a$  or  $S_c$ , because they have the most elements. Since  $S_a$  and  $S_c$  has the same number of elements, maximum set cover heuristic randomly chooses one of them. Let the chosen subset be  $S_a$ .

After choosing  $S_a$ , the algorithm removes the elements the chosen subset covers from the other subsets. Since  $S_a$  intersects with only  $S_b$ , the intersection is removed from  $S_b$  and after the first iteration  $S_b$  has two remaining elements, which are  $\{5, 6\}$ . During the second iteration, since  $S_c$  has the most uncovered elements, the algorithm chooses  $S_c$  and removes the elements it covers from other subsets. As a result of the second iteration, subset  $S_b$  has no elements. During the third iteration, the algorithm chooses  $S_d$  and as a result, no elements remain uncovered and the algorithm terminates with the chosen subsets  $S_a$ ,  $S_c$  and  $S_d$ .

For our case, consider the sample topology given in Fig. 2(a) with the desired connectivity degree ( $k$ ) value of two. In this topology there are five sensor nodes, so that all of them are initially pairwise connected. During the first step all the sensor nodes behave like they also have UAV-BS capabilities and asks sensor nodes to calculate disjoint paths destined to themselves. Sensor nodes compute their preferred paths to other sensors using the initialization phase of MDP algorithm, described in Sect. 3.2. According to these paths, ignoring the intermediate nodes and denoting number of occurrences with exponents, destination nodes of node  $a$  are  $c^2$ ,  $e$  and  $d$ ; destination nodes of node  $b$  are  $c^2$  and  $d^2$ ; destination nodes of node  $c$  are  $a, b, d^3$  and  $e$ ; destination nodes of node  $d$  are  $b, c^3$  and  $e$ ; finally destination nodes of node  $e$  are  $c^2$  and  $d^2$ . During the second step, using the set-cover algorithm that chooses the node that covers most elements in each iteration, MDP algorithm chooses node  $c$  to be a UAV. In this way, all the sensor nodes in the uncovered node list are covered twice and uncovered node list becomes empty. Therefore one UAV-BS is adequate for this sample topology when  $k = 2$ . The final topology, where a UAV-BS is located at the location of node  $c$ , is shown in Fig. 2(b).

With the increasing  $k$  value, the additional load on the MDP algorithm does not significantly increase and the MDP algorithm works very efficiently. Continuing from the example above, when  $k = 3$ , there will be no change in the first step, but the uncovered node list, in the second step, will initially have three copies of each node. During the first iteration, node  $c$  will be chosen and there will remain one copies of node  $a$ ,  $b$  and  $e$  in the uncovered node list. In the second iteration, node  $d$  will be chosen and the algorithm will terminate with the decision of placing two UAV-BSs to the locations of node  $c$  and  $d$ .

## 4 Experimental results

In this section we report our measurements regarding the minimum number of UAV-BSs required to maintain  $k$ -vertex UAV connectivity. We also discuss the contribution of placement of UAV-BSs at known locations to the

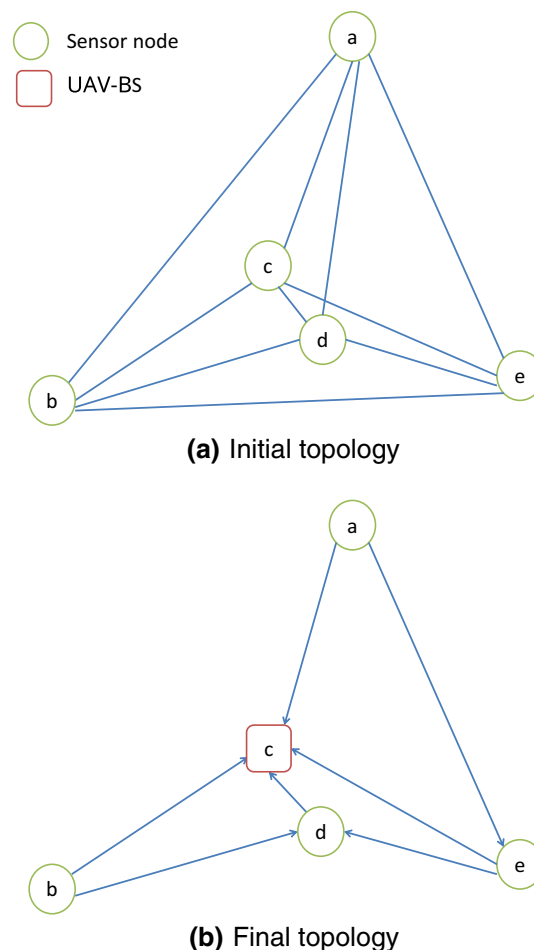


Fig. 2 Example UAV-BS location determination for  $k = 2$

network lifetime compared to random distribution of UAV-BSs. We also consider the number of message transmissions required during execution of our algorithm to measure its efficiency. In more detail, the metrics we mainly focus on to evaluate our algorithms are:

- *Number of required UAV-BSs* For each randomly generated network topology our first aim is to determine the minimum number of UAV-BSs required to maintain  $k$ -vertex UAV connectivity for different  $k$  values and network topologies. Since the aim of this study is to minimize the number of required UAV-BSs, this is an important metric for the evaluation of our algorithm.
- *Lifetime* We compare UAV-connected and  $k$ -vertex UAV connected lifetimes of the randomly generated topologies. For this evaluation we consider the first time UAV-connectivity and  $k$ -vertex UAV connectivity is broken. This metric shows the effect of how well our algorithm places the UAV-BSs and its effect to the overall lifetime of the network; hence this is also an important performance metric.



- *Number of message transmissions* This is the sum of the number of messages transmitted during the execution of our algorithm. This metric depicts the transmission power cost of generating the  $k$ -vertex UAV-BS connected topology. Since it directly affects the remaining energy levels of the sensor nodes in the network, we also monitored this metric.

We have implemented MDP using an extended version of a custom evaluation framework, which can generate random network topologies, execute the algorithms on the generated topologies, calculate the desired metrics, and visualize the outputs of the experiments. This framework has also been used for evaluating DPV and ADPV algorithms. This framework has the ability to measure UAV-connected and  $k$ -vertex UAV-connected lifetimes of the given heterogeneous WSNs. In order to measure the lifetime with random and uniform distribution of UAV-BSs, we use the ADPV algorithm with random UAV-BS locations and represent it as ADPV with random UAV-BS locations (ADPVR) throughout this section. The simulator and simulation models were developed using C# programming language and all experiments have been conducted on an eight-core Intel Xeon processor server running the Windows Server 2012 operating system with 16 GB of RAM.

#### 4.1 Experimental setup

In our experiments, we vary the number of sensor nodes in the network between 100 and 500, and assume that they are deployed uniformly and randomly in a  $600\text{ m} \times 600\text{ m}$  area. They can be either evenly or unevenly distributed on the ground due to operational requirements or terrain limitations. We assumed that the maximum communication range of the sensor nodes is 100 m. For the degree of disjoint connectivity, we executed the simulations for both  $k = 2$  and  $k = 3$ , which are typical values seen in  $k$ -connectivity studies [2, 5, 9]. For each message transmission, we assumed a packet loss rate of 10% and path loss exponent of 2. We assumed that the path length was bounded by a constant, say  $l = 6$ , following previous studies [5, 9, 14]. We also assumed initial power levels of the sensor nodes randomly vary from 80 to 100%. We also assumed that nodes can predict link length using the strength of the received signal. We did not target a specific sensor node technology in the simulations, but it can be considered as any sensor node platform capable of adjusting transmit power with a maximum attainable range  $R_{max}$  of 100 m. Finally, we repeated our experiments for at least 20 times to obtain more reliable and stable results and to report the average values at confidence intervals of 95%. Confidence intervals for reported values are shown on the corresponding charts.

Our simulation parameters are summarized in Table 2. For the purpose of comparing the algorithms, the power consumption model we used for sensor nodes is described in Sect. 4.2. For each set of scenarios, we first determine the minimum number of UAV-BSs and their locations using the MDP algorithm. Then, by randomly relocating the UAV-BSs, we generate another network topology and measure lifetimes of both topologies. We assume that the supernodes are mounted on rotary-wing UAVs, such as quadrotor drones, which can hover and remain stationary over a given area at a specified altitude. We assume each point has an altitude constraint that shows the minimum altitude at which UAVs can hover and we assume UAV-BSs fly at a fixed altitude, say 10 m, and assume that there is a line-of-sight communication link with sensor nodes in the range.

#### 4.2 Power consumption model

For the purpose of comparing the algorithms we use a well-known power consumption model proposed by Heinzelman et al. [18, 19]. This approach is based on the observation that the main factor in WSN power consumption is data communication, which consists of two factors: data transmission and data reception. In this model, the energy consumption for data reception is a constant value per bit. We represent this constant with  $\beta$  and assume that it is equal to 50 nJ/bit. For data transmission, the power to transmit a bit to a distance of  $d$  is  $\alpha_1 + \alpha_2 \times d^n$ , where  $\alpha_1$  and  $\alpha_2$  are parameters that depend on the transmitter circuitry, and  $n$  is the path loss exponent for the environment, which often has a value between 2 and 4. In our power consumption model, maximum initial energy,  $\alpha_1$ ,  $\alpha_2$ , and  $n$  are assumed to be 5J, 50 nJ/bit, 100 pJ/bit/m<sup>2</sup> and 2, respectively.

For our experiments, we assume all sensor nodes sense the environment and generate traffic at a fixed data rate of 250 kbps. We also assume that data aggregation is applied and that all nodes on a path carry the same load. Therefore, the total power consumption for receiving a bit and

**Table 2** Simulation parameters

Number of sensor nodes: $N$	[100...500]
Deployment area	600 m $\times$ 600 m
Initial transmission range of sensor nodes: $R_{max}$	100 m
Degree of disjoint connectivity: $k$	2 and 3
Packet loss rate	10%
Maximum path length	6
Power attenuation exponent	2
Confidence level	95%

transferring it to the next hop is equal to  $P_f(d) = \beta + \alpha_1 + \alpha_2 \times d^m$ .

### 4.3 Results

In Fig. 3, we present the results of MDP algorithm regarding the minimum number of UAV-BSs required to maintain a two- and three-vertex UAV connected network topology, respectively. According to the results, our first observation is that as  $k$  value increases, the number of required UAV-BSs also increases. Since higher  $k$  values require more disjoint paths between the sensor nodes and the set of UAV-BSs, this is the expected behavior.

Our second observation is that number of sensor nodes is inversely proportional to the required number of UAV-BSs. With the increasing number of sensor nodes, the number of UAV-BSs required to maintain  $k$ -vertex UAV connectivity decreases for both  $k = 2$  and  $k = 3$ . With the increasing number of sensor nodes, it becomes easier to find alternative routes, and so this is also an expected behavior and the MDP results align with the expected behavior. For dense networks, we can observe that the number of required UAV-BSs is quite low. For instance, in one extreme, when we look at the results of a 500-node network, for  $k = 2$ , MDP determines the number of required UAV-BSs to be around 5. Even though this number may not be the minimum, it shows MDP is quite successful in determining almost the minimum number of UAV-BSs and their locations.

In previous studies [5, 9, 14], with the increasing number of sensor nodes, they also increased the number of supernodes. For the evaluation, they used a ratio of 5% and 10% supernodes, and they initially deployed supernodes randomly, regardless of where the sensor nodes were located. For example, for a network topology of 500 sensor nodes and for  $k = 2$ , they executed their simulations for 25

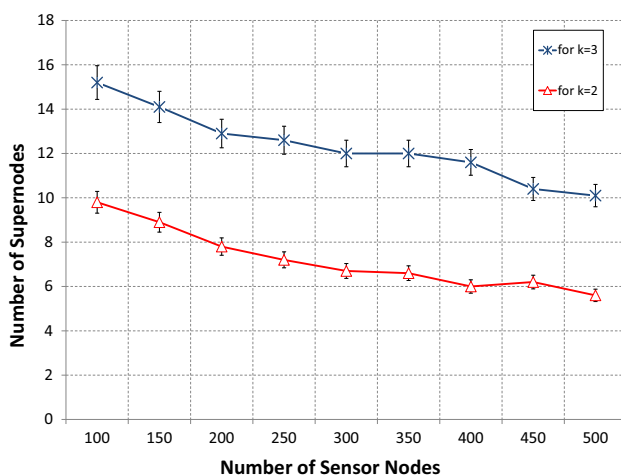


Fig. 3 Number of required UAV-BSs

and 50 supernodes. Our study, on the other hand, reveals that far fewer supernodes might be sufficient (mean 5.6 UAV-BSs within the 95% confidence interval [5.29,5.91]) to maintain the same level of fault-tolerance for the same network topologies. For  $k = 3$ , they do not change the number of supernodes for their experiments, but our analysis shows that on average 1.73 (within the 95% confidence interval [1.68,1.80]) times more supernodes are needed compared to  $k = 2$ .

In Fig. 4, we compare the UAV-connected and  $k$ -vertex UAV connected lifetimes of the topologies generated by the MDP and ADPVR algorithms and observe the most striking result. Note that, for this evaluation, we consider the first time UAV-connectivity and  $k$ -vertex UAV connectivity has been discontinued. We observe that, simply by changing the locations of the UAV-BSs, MDP can improve both UAV-connected and  $k$ -vertex UAV connected lifetimes of the networks. For  $k = 2$ , MDP shows 29% improvement and for  $k = 3$  and 22% improvement, on average, in UAV-connected lifetimes of the network compared to random deployment of UAV-BSs. Moreover, for the same experimental scenarios, the MDP provides respectively 35% and 19% longer two-vertex and three-vertex UAV-connected lifetimes, on average, than the random deployment of UAV-BSs. This result can be attributed to the fact that the MDP becomes more efficient in finding better positions for UAV-BSs.

In Fig. 5, we compare network UAV-connected and 2-vertex UAV connected lifetimes for different UAV-BS counts and also observe the effect of UAV-BS counts on the lifetime of the network. For example, for this scenario, for 200 sensor nodes and for a UAV ratio of 10%, 20 UAV-BSs and for 200 sensor nodes and for a UAV ratio of 5%, 10 UAV-BSs are randomly deployed on the network and lifetime measurements are compared with the MDP algorithm, which uses a minimum number of UAV-BSs. For these settings, ADPVR shows a 75% and 25% improvement in the UAV-connected lifetimes compared to MDP using four-fold and two-fold more UAV-BSs on average, respectively. According to the results, as discussed in [14], with the increasing number of UAV-BSs, lifetime also increases. However, the relationship between the increase in the number of UAV-BSs and the increase in lifetime is sublinear, and therefore, the increase in lifetime becomes insignificant as the number of UAV-BSs increases.

Since sensor nodes consume most of their battery power for the transmission of messages, another important metric we need to take into account is the number of required message transmissions. This metric represents the cost of the transmission power to generate the  $k$ -vertex UAV-BS connected topology. If the power consumption to determine UAV locations is too high, the efficiency of our algorithm no longer makes sense.

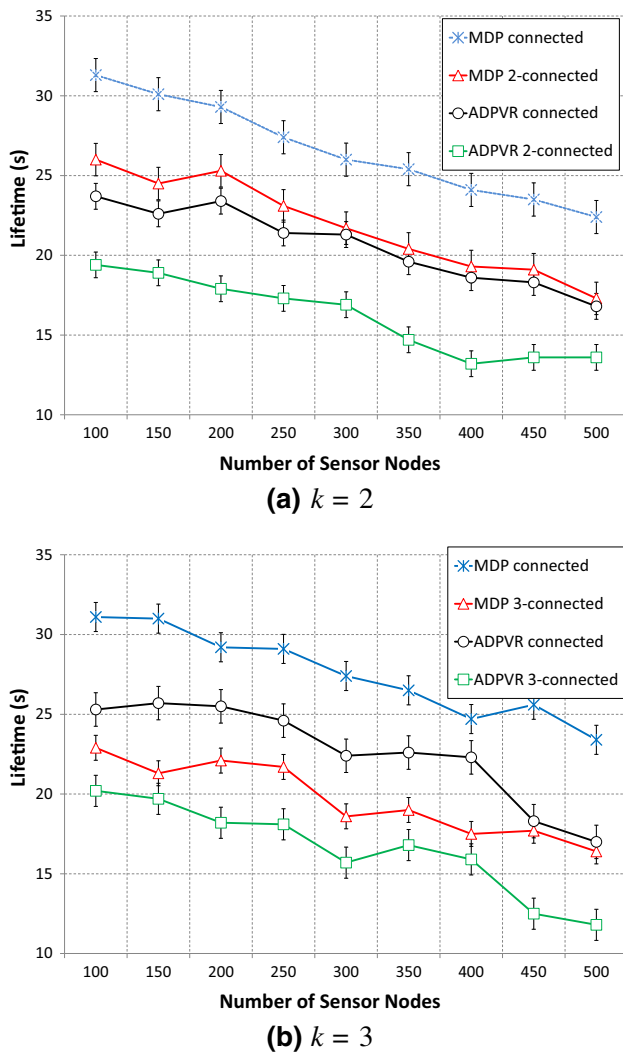


Fig. 4 Lifetime comparison of the MDP and ADPVR algorithms

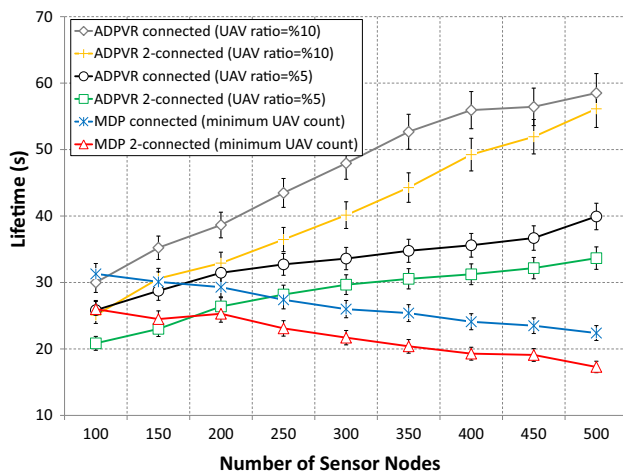


Fig. 5 Lifetime comparison of the MDP and ADPVR algorithms for different UAV-BS counts

In Fig. 6, we can observe the total number of required message transmissions to execute the MDP and ADPVR algorithms. Since ADPVR algorithm only uses message transmissions to determine transmission ranges of sensor nodes and randomly determines UAV-BS locations without making any message transmissions, as expected, the number of message transmissions of ADPVR algorithm is lower than that of MDP algorithm. As shown in Fig. 6(a) and (b), for  $k = 2$  and  $k = 3$ , MDP incurs a 100% and 50% increase on the required number of message transmissions compared to ADPVR, respectively. As seen from the figures, the number of message transmissions of both algorithms increases almost linearly with the number of sensor nodes, and the ratio of these message transmission counts does not change significantly. Considering that in the MDP algorithm, every sensor node behaves like a UAV-BS and lets sensor nodes compute disjoint paths destined for itself, this is also understandable.

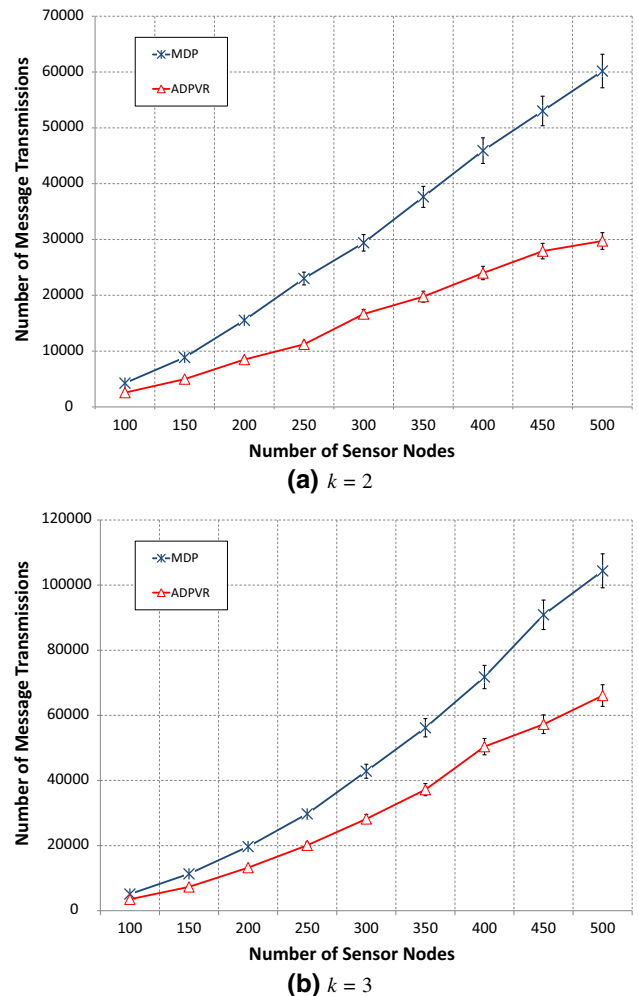


Fig. 6 Number of message transmissions in MDP and ADPVR algorithms

## 5 Conclusion

In this study, we present the MDP algorithm, a distributed and energy-efficient approach to locate minimum number of UAV-BSs to maintain  $k$ -vertex UAV connectivity. The motivation of this algorithm is to optimize the use of UAV-BS in heterogeneous WSNs and also to prolong UAV-connected network lifetime. MDP places UAV-BSs with respect to the positions of already-deployed sensor nodes using an optimization that is based on the set-cover problem. The experimental results for the MDP algorithm validate our theoretical expectations. The number of UAV-BSs required to make the topology  $k$ -vertex UAV connected decreases as the sensor node density increases. Also, with the increasing fault-tolerance degree, the number of required UAV-BSs also increases. With the extensive simulations, we demonstrate that MDP can significantly prolong UAV-connected lifetime of the network up to 40% compared to uniform distribution of UAV-BSs. The solution we propose in this paper is distributed and localized, so it is scalable to large networks and therefore suitable for use in real applications.

As part of future work, we plan to perform our simulations with a practical platform comprising real UAV-BSs and relevant devices. Additionally, in order to achieve a more robust network topology, we want to extend our approach by changing UAV-BS positions dynamically and relocating additional ones whenever necessary. In this way, instead of being stationary, the UAV-BSs can be mobile and could therefore be repositioned to further increase network lifetime.

## References

1. Alemdar, H., & Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15), 2688–2710. <https://doi.org/10.1016/j.comnet.2010.05.003>.
2. Almasaeid, H.M., & Kamal, A.E. (2009). On the minimum  $k$ -connectivity repair in wireless sensor networks. In *2009 IEEE International Conference on Communications* (pp. 1–5).
3. Alzenad, M., El-Keyi, A., Lagum, F., & Yanikomeroglu, H. (2017). 3-d placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage. *IEEE Wireless Communications Letters*, 6(4), 434–437.
4. Azharuddin, M., & Jana, P.K. (2015). A ga-based approach for fault tolerant relay node placement in wireless sensor networks. In *Computer, Communication, Control and Information Technology (C3IT), 2015 3rd International Conference on* (pp. 1–6). <https://doi.org/10.1109/C3IT.2015.7060111>.
5. Bagci, H., Korpeoglu, I., & Yazici, A. (2014). A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(4), 914–923.
6. Bari, A., Jaekel, A., Jiang, J., & Xu, Y. (2012). Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements. *Computer Communications*, 35(3), 320–333. <https://doi.org/10.1016/j.comcom.2011.10.006>.
7. Bellazreg, R., Boudriga, N., & An, S. (2013). Border surveillance using sensor based thick-lines. In *Information Networking (ICOIN), 2013 International Conference on* (pp. 221–226). <https://doi.org/10.1109/ICOIN.2013.6496380>.
8. Bogdanov, A., Maneva, E., & Riesenfeld, S. (2004). Power-aware base station positioning for sensor networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 1, p. 585).
9. Cardei, M., Yang, S., & Wu, J. (2008). Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 19(4), 545–558. <https://doi.org/10.1109/TPDS.2007.70768>.
10. Cicek, C.T., Gultekin, H., Tavli, B., & Yanikomeroglu, H. (2019). Uav base station location optimization for next generation wireless networks: Overview and future research directions. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)* (pp. 1–6).
11. Dandekar, D.R., & Deshmukh, P.R. (2013). Energy balancing multiple sink optimal deployment in multi-hop wireless sensor networks. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International* (pp. 408–412).
12. Das, D., Rehena, Z., Roy, S., & Mukherjee, N. (2013). Multiple-sink placement strategies in wireless sensor networks. In *2013 5th International Conference on Communication Systems and Networks (COMSNETS)* (pp. 1–7).
13. Davis, A., & Chang, H. (2012). Underwater wireless sensor networks. *Oceans, 2012*, 1–5. <https://doi.org/10.1109/OCEANS.2012.6405141>.
14. Deniz, F., Bagci, H., Korpeoglu, I., & Yazici, A. (2016). An adaptive, energy-aware and distributed fault-tolerant topology-control algorithm for heterogeneous wireless sensor networks. *Ad Hoc Networks*, 44, 104–117.
15. Dhillon, S. S., & Chakrabarty, K. (2003). Sensor placement for effective coverage and surveillance in distributed sensor networks. *IEEE Wireless Communications and Networking*, 3, 1609–1614.
16. Haeyong, K., Taekyoung, K., & Pyeongsoo, M. (2008). Multiple sink positioning and routing to maximize the lifetime of sensor networks. *IEICE Transactions on Communications*, 91(11), 3499–3506.
17. Hanh, N.T., Le Nguyen, P., Tuyen, P.T., Binh, H.T.T., Kurniawan, E., & Ji, Y. (2018). Node placement for target coverage and network connectivity in wsns with multiple sinks. In *2018 15th IEEE Annual Consumer Communications and Networking Conference (CCNC)* (pp. 1–6).
18. Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on* (Vol. 2, p. 10). <https://doi.org/10.1109/HICSS.2000.926982>.
19. Heinzelman, W., Chandrakasan, A., & Smith, A. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1, 660–670.
20. Akyildiz, I. F., & Kasimoglu, I. H. (2004). Wireless sensor and actor networks: Research challenges. *Ad Hoc Networks*, 2(4), 351–367.
21. Kalantari, E., Yanikomeroglu, H., & Yongacoglu, A. (2016). On the number and 3d placement of drone base stations in wireless cellular networks. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)* (pp. 1–6).
22. Karp, R. (1972). Reducibility among combinatorial problems. In: R. Miller, J. Thatcher, & J. Bohlinger (Eds.), *Complexity of*

- Computer Computations, The IBM Research Symposia Series* (pp. 85–103). US: Springer.
23. Kashyap, A., Khuller, S., & Shayman, M. (2006). Relay placement for higher order connectivity in wireless sensor networks. In *INFOCOM 2006. Twenty Fifth IEEE International Conference on Computer Communications. Proceedings* (pp. 1–12).
  24. Kim, H., Ben-Othman, J., & Bellavista, P. (2017). Collision-free reinforced barriers in UAV networks. *Journal of Computational Science*, 22, 289–300.
  25. Kim, H., & Cobb, J. A. (2015). Optimization algorithms for transmission range and actor movement in wireless sensor and actor networks. *Computer Networks*, 92, 116–133.
  26. Kim, S., Ko, J.G., Yoon, J., & Lee, H. (2007). Multiple-objective metric for placing multiple base stations in wireless sensor networks. In *2007 2nd International Symposium on Wireless Pervasive Computing* (pp. 627–631).
  27. Korte, B., & Vygen, J. (2012). *Combinatorial optimization: theory and algorithms* (5th ed.). Berlin: Springer.
  28. Lin, G. H., & Xue, G. (1999). Steiner tree problem with minimum number of Steiner points and bounded edge-length. *Information Processing Letters*, 69(2), 53–57.
  29. Liu, B., Zhu, Q., & Zhu, H. (2020). Trajectory optimization and resource allocation for UAV-assisted relaying communications. *Wireless Networks*, 26(1), 739–749.
  30. Liu, H., Nayak, A., & Stojmenović, I. (2009). Fault-tolerant algorithms/protocols in wireless sensor networks. In S. Misra, I. Woungang, S. Misra (Eds.), *Guide to wireless sensor networks. Computer communications and networks*. London: Springer. [https://doi.org/10.1007/978-1-84882-218-4\\_10](https://doi.org/10.1007/978-1-84882-218-4_10).
  31. Lloyd, E., & Xue, G. (2007). Relay node placement in wireless sensor networks. *IEEE Transactions Computers*, 56(1), 134–138.
  32. M. Yarvis, Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y., & Singh, S. (2005). Exploiting heterogeneity in sensor networks. In *Proceedings of the IEEE International Conference on Computer Communication* (Vol. 2, pp. 878–890). Miami, FL, USA. <https://doi.org/10.1109/INFCOM.2005.1498318>.
  33. Ma, C., Liang, W., Zheng, M., & Yang, B. (2019). Relay node placement in wireless sensor networks with respect to delay and reliability requirements. *IEEE Systems Journal*, 13(3), 2570–2581. <https://doi.org/10.1109/JSYST.2018.2838072>.
  34. Mozaffari, M., Saad, W., Bennis, M., Nam, Y., & Debbah, M. (2019). A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Communications Surveys Tutorials*, 21(3), 2334–2360. <https://doi.org/10.1109/COMST.2019.2902862>.
  35. Oyman, E.I., & Ersoy, C. (2004). Multiple sink network design problem in large scale wireless sensor networks. In *Communications, 2004 IEEE International Conference on* (Vol. 6, pp. 3663–3667).
  36. Pardesi, P., & Grover, J. (2015). Improved multiple sink placement strategy in wireless sensor networks. In *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on* (pp. 418–424).
  37. Poe, W. Y., & Schmitt, J. B. (2007). Minimizing the maximum delay in wireless sensor networks by intelligent sink placement. *Distributed Computer Systems Lab University of Kaiserslautern*, 67655, 1–20.
  38. Qiu, L., Chandra, R., Jain, K., & Mahdian, M. (2004). Optimizing the placement of integration points in multi-hop wireless networks. *Proceedings of ICNP*, 4, 271–282.
  39. Rehana, R.T., Maneesha, R.V., & Sangeeth, K. (2008). Fault tolerant clustering approaches in wireless sensor network for landslide area monitoring. In *The International Conference on Wireless Networks* (pp. 107–113).
  40. Safa, H., El-Hajj, W., & Zoubian, H. (2014). A robust topology control solution for the sink placement problem in wsns. *Journal of Network and Computer Applications*, 39, 70–82.
  41. Senel, F., & Younis, M. (2016). Novel relay node placement algorithms for establishing connected topologies. *Journal of Network and Computer Applications*, 70, 114–130.
  42. Sitanayah, L., Brown, K. N., & Sreenan, C. J. (2014). A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wireless sensor networks. *Ad Hoc Networks*, 23, 145–162.
  43. Sitanayah, L., Brown, K. N., & Sreenan, C. J. (2015). Planning the deployment of multiple sinks and relays in wireless sensor networks. *Journal of Heuristics*, 21(2), 197–232.
  44. Slavík, P. (1996). A tight analysis of the greedy algorithm for set cover. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, STOC '96* (pp. 435–441). ACM, New York, NY, USA (1996). <https://doi.org/10.1145/237814.237991>.
  45. Xu, X., Liang, W. (2011). Placing optimal number of sinks in sensor networks for network lifetime maximization. In *2011 IEEE International Conference on Communications (ICC)* (pp. 1–6).
  46. Youssef, W., & Younis, M. (2007). Intelligent gateways placement for reduced data latency in wireless sensor networks. In *2007 IEEE International Conference on Communications* (pp. 3805–3810).
  47. Zhao, H., Wang, H., Wu, W., & Wei, J. (2018). Deployment algorithms for UAV airborne networks toward on-demand coverage. *IEEE Journal on Selected Areas in Communications*, 36(9), 2015–2031.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Fatih Deniz** received his B.S. degree in Computer Science from Bilkent University in 2007. He received his MS and Ph.D. degrees from Middle East Technical University in 2010 and 2016, respectively, both in Computer Science. He has worked in the Central Bank of the Republic of Turkey as an IT specialist since 2012. His research interests include fault-tolerant topology-control algorithms in wireless sensor networks.



**Hakki Bagci** received his M.S. (2007) and Ph.D. (2014) degrees from Bilkent University and Middle East Technical University, respectively, both in Computer Science. His research interests include distributed-algorithm design for wireless sensor networks.



**Ibrahim Korpeoglu** is Chair and Professor at the Department of Computer Engineering, Bilkent University, Ankara, Turkey. He received his M.S. (1996) and Ph.D. (2000) degrees from University of Maryland at College Park, both in Computer Science. He received his B.S. degree (summa cum laude) in Computer Engineering from Bilkent University in 1994. Since 2002, he has been a faculty member in the Department of Computer Engineering,

Bilkent University. Previously, he worked at several research and

development companies in the USA including Ericsson, IBM T.J. Watson Research Center, Bell Laboratories and Bell Communications Research (Bellcore). He received Bilkent University Distinguished Teaching Award in 2006 and the IBM Faculty Award in 2009. He is a member of ACM and a senior member of IEEE.



**Adnan Yazici** is a full professor in the Department of Computer Science, Nazarbayev University, Nur-Sultan, Kazakhstan. He received his Ph.D. in Computer Science from the Department of EECS at Tulane University in the US, in 1991. His current research interests include intelligent database systems, spatio-temporal databases, multimedia and video databases and wireless multimedia sensor networks. Prof. Dr. Yazici has published more

than 180 international technical papers. He received the IBM Faculty Award in 2011 and the Parlar Foundation's Young Investigator Award in 2001. Prof. Dr. Yazici was a Conference Co-Chair of the 23rd IEEE International Conferences on Data Engineering in 2007, the 38th Very Large Data Bases Conference in 2012 and Program Co-Chair of the Flexible Query Answering Systems Conference in 2011. He is the director of the Multimedia Database Lab at METU. He is a member of ACM and a senior member of IEEE.