Placement of UAVs to Reconnect Lost Subnetworks in Wireless Sensor Networks

Lu Lin*, Xiaojun Zhu*†§, Ji'ao Tang*, Chao Dong‡
*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China.
†State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China.
‡College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, China.

***Abstract*—When a sensor network becomes disconnected, there are still many nodes that function correctly. To reconnect these lost nodes with the sink, we can deploy UAVs in-between to relay data. In contrast to previous works treating disconnected nodes as independent data sources, we consider a cooperative approach where sensors form subnets to facilitate data transmission. We formulate the problem to maximize the number of nodes that can find a route to the sink, given a limited number of relay UAVs. We prove that the problem is NP-hard. We then propose a two-step heuristic algorithm for the relay placement problem, show that the algorithm runs in polynomial time and analyze its approximation ratio. Simulations verify the effectiveness of the proposed approach.**

I. Introduction

The network lifetime can be formulated as the total amount of time that the network can maintain its full functionality [1]. The lifetime of a wireless sensor network is limited because it consists of low-cost and low-power sensor nodes. When a certain percentage of nodes die, the lifetime of this network is considered to be at the end [2]. But in fact, bottleneck nodes that cause the network to disconnect are only a minority, most nodes still can work normally but cannot forward data to the sink; thus, they are considered lost nodes. In this case, we can restore routing for disconnected regions to prolong the network lifetime by deploying a number of relay nodes, e.g. UAVs, thus the lost nodes can re-find a path to forward data.

Due to their high maneuverability, flexible deployment, and low cost, UAVs have recently attracted great interest in aiding wireless communication [3–10]. Compared with traditional ground communication, the ground-to-air channel between the UAV flying at high altitude and the ground node is mainly a line-of-sight channel, so the data transmission rate is more stable and more suitable for data forwarding. Compared to static nodes as relays, UAV relays have the advantage of flexible deployment. When the network is not connected again, the UAV is able to fly to the nearest node that is connected to the sink, be dispatched again by the sink and return before the power runs out.

However, in most theoretical studies on the deployment of relay UAVs, the communication distance and the communication radius are used as the basis for whether the UAV can communicate with the ground nodes [11–13]. In practice, the quality of communication between the UAV and the ground

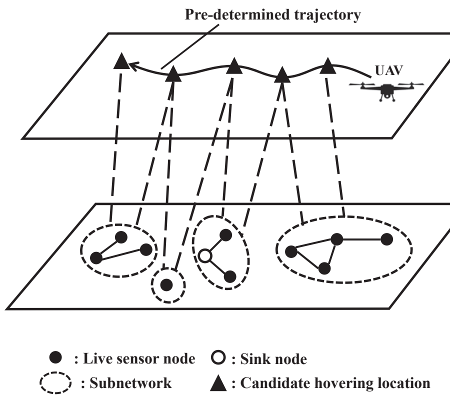


Fig. 1. A UAV follows a predetermined trajectory to survey the connection status of the sensor network. According to survey result, we can form subnets and get the connectivity between subnets and UAVs at each candidate hovering location.

node can be affected by other factors (e.g., buildings), resulting in poor signal reception at some hovering locations. This model is therefore too ideal. In addition, most of the current research focuses on the full connectivity of the network and does not take into account the limited number of UAVs.

In this paper, we consider how to maximize the recovery of network connectivity by deploying multi-UAV relays when the number of UAVs is limited, and for a better practical application, we deploy relays based on the topology maps actually detected by the UAV. We first let the lost nodes that can communicate with each other spontaneously form a subnet, then dispatch one or more UAVs to the lost area to collect information on subnets. As shown in Fig. 1, the UAV follows a predetermined trajectory and hovers over different nodes to obtain the affiliation of the nodes and subnets and record the connectivity to the subnets when hovering at different locations. We use the locations where the UAV hovers while collecting information as candidate hovering locations where the UAV is deployed as a relay. With the information UAV get, we select the deployment locations for a limited number of UAV relays from candidate hovering locations.

As the UAV may connect to the same subnet when hovering at different locations, the selection of one hovering location will affect other locations. Additionally, when designing the deployment scheme, we need to ensure that the subnets even-

§ corresponding author. xzhu@nuaa.edu.cn

ISBN 978-3-903176-57-7© 2023 IFIP

tually communicate with the sink. To tackle these problems, we propose a heuristic approach to select deployment locations for multi-UAV relays to enable as many lost nodes as possible to be reconnected with a limited number of UAVs.

Our contributions can be enumerated as follows.

1) We formulate a cooperative data forwarding scheme for UAV-based network restoration, where sensor nodes form subnets to forward data in cooperation with the UAV. We prove that the problem is NP-hard.
2) We propose a heuristic approach to select deployment locations for UAV relays which enable as many lost nodes as possible to be reconnected with the limited number of UAVs, and analyze the algorithm complexity.
3) We evaluated our algorithm exploring the impact of network size, UAV limitation number, and candidate hovering location distribution on the number of lost nodes reconnected. Simulations show the effectiveness of the proposed approach.

The rest of this paper is organized as follows. Section II reviews related works. Section III gives the system model, formulates the problem and proves it is NP-hard. Section IV analyzes the problem, proposes a heuristic algorithm and analyzes the complexity and approximate ratio. Section V conducts simulations to verify the effectiveness of the proposed solution, and Section VI concludes the paper.

II. Related Work

UAVs can reconnect the network by acting as base stations or relay nodes. In [14], UAVs act as aerial flying base stations to serve ground users during temporary events. To achieve cost-effective UAV deployment in an autonomous and dynamic manner, reference [15] proposed a machine learning-based intelligent deployment scheme for UAV base station, and a novel framework is proposed to allow predictive deployment of UAVs as temporary base stations in [16].To achieve fair performance among users, reference [17] maximizes minimum throughput for all ground users in downlink communication by optimizing multiuser communication scheduling and association jointly with UAV trajectory and power control.

When UAVs act as relay nodes, optimizing the deployment of relays for different objectives is an important issue. In [18], multiple UAVs are used as UAV relays between IoT devices and BS to enhance the strength of the signal received in BS. Reference [19] studied the use of multiple UAVs in relaying, they first optimize the placement of the UAVs by maximizing the end-to-end signal-to-noise ratio, then compared the relay setups of multi-hop single link and multiple dual-hop links in terms of outage and bit error rate. To jointly optimize relay deployment, channel allocation, and relay assignment in a self-organized network, reference [12] divided the original problem into two sub-problems, by solving the two sub-problems alternately and iteratively, the original problem is finally solved. In order to tackle the coupled relationship among UAV relays, reference [13] modeled the problem of multi-relay deployment as a local interaction game. Reference [20] uses the moth flame optimizer (MFO) algorithm, interior search algorithm (ISA), and bat algorithm (BA) to identify the optimal positions for the placement of RNs to achieve full network connectivity. Unfortunately, these references do not take into account the number of UAVs when reconnecting the network.

To minimize the number of deployed relay nodes, the traditional approach is to use a minimum Steiner tree to find deployment locations [21–23]. However, since in our model, candidate hovering locations for UAV relay are predetermined, and there are no weights on the edges between subnets and candidate hovering locations, the Steiner tree approach cannot be used directly to solve our problem.

Similarly to our scenario, reference [24] obtains the network topology by first sending UAVs to probe the status of the network, and then uses the collected topology information to select the location of the deployment that can improve data delivery across the network. Reference [25] improves [24], which proposes a route reconstruction algorithm based on distributed network diagnosis and progressive relay node placement using UAVs. In this work, there is no sink in the network; thus, the objective is different from ours, which makes their solutions not applicable to our situation.

III. Problem Formulation

A UAV is dispatched to the disconnected area to check the status of the sensor network connection. It follows a predetermined trajectory, hovers at regular intervals, and collects network information at each hovering location. From the UAV we can get the information needed by our problem. Specifically, there are m subnets $S = \{s_1, s_2, \ldots, s_m\}$, where s_1 represents the subnet containing the sink node, and the rest subnets are composed of lost but live nodes. The number of nodes contained in each subnet s_i is n_i. There are also t candidate hovering locations $P = \{p_1, p_2, \ldots, p_t\}$, in which the UAV hovers while detecting connectivity to the subnets. The disconnected network can be represented by a bipartite graph $G = (S \cup P, E)$, as shown in Fig. 2, where $(i, j) \in E$ if and only if s_i can communicate with the UAV hovering at p_j. Also, at most C UAVs can serve as relays, i.e., we can only select C locations from the t ($t \geq C$) candidate hovering locations to deploy the UAV relays. Table I summarizes the notations.

TABLE I
Explanation of notations

Notation	Explanation
S	Collection of subnets
m	Total number of subnets
s_1	Subnet containing sink node
P	Collection of candidate hovering locations
t	Total number of candidate hovering locations
C	Limited number of UAVs

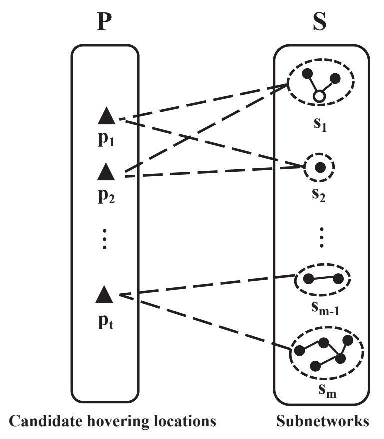


Fig. 2. A bipartite graph $G = (S \cup P, E)$, where S is the set of subnets, and P is the set of candidate hovering locations.

Since our objective is to maximize the number of nodes that can find a routing path to forward data to the sink, our optimization problem can be expressed as follows:

$$\max_{\mathbf{x}} \sum_{i=0}^{M} n_i f_i$$

subject to,

$$\begin{cases} \sum_{j} x_j \leq C \\ \\ f_i \leq \min \left\{ 1, \sum_{\substack{path\ q \\ from\ s_i\ to\ s_1}} \prod_{p_j \in q} x_j \right\} \end{cases} \tag{1}$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_t\}$ with x_j indicating whether p_j is selected and f_i indicates whether s_i can be connected to s_1.

Our model makes practical sense, as our selection of deployment locations is based on the network topology, which is obtained after sending a UAV to the candidate hovering locations to survey the connection status of the network. Before dispatching the UAV to the lost area, we set the flight trajectory and hovering locations of the UAV (e.g., fly in a zigzag trajectory, hovering every 10 meters), and use these hovering locations as follow-up candidate hovering locations in the algorithm. During the flight, the UAV receives the number of nodes from the subnet, and records the connectivity to the subnets when hovering at each location. In addition, since our main research goal is how to choose the deployment locations so that as many nodes as possible can restore the connection, the performance/resource consumption of UAVs and nodes is not considered in the model.

Problem 1. *We are given a disconnected network $G = (S \cup P, E)$ and the UAV relays limitation number C, where $S = \{s_1, s_2, \ldots, s_m\}$ is the set of m maximal subnets with $n_1, n_2, \ldots, n_m$ nodes, $P = \{p_1, p_2, \ldots, p_t\}$ is the set of t candidate hovering locations, and E is the set of edges with $(i, j) \in E$ when the UAV can communicate with s_i while hovering at p_j. The problem is to select C deployment locations from the candidate hovering locations to maximize the number of nodes that can re-find a routing path to the sink.*

To the best of our knowledge, there is currently no readily available way to solve our problem. In our problem, there are two main difficulties: one is that the local optimum cannot guarantee the global optimum, i.e., when there exists a subnet that is far from the sink but contains a large number of lost nodes, how can we centrally deploy UAV relays on the path from this subnet to the sink; another is how to ensure that subnets can find a path to the sink, that is, to avoid the situation where subnets interconnect but are never able to forward information to the sink. Since the problem is more complicated if the data can be forwarded through multiple relays between the two subnets, in this paper, we only consider the case of single-hop, i.e., data is not forwarded between relays.

We will prove that Problem 1 is NP-hard by reducing from the classic NP-hard problem, set cover.

Problem 2. *(Set Cover Problem, SCP). Given a universe $\mathcal{U}$, a family $\mathcal{S}$ of subsets of $\mathcal{U}$, and an integer k, is there a subfamily $\mathcal{C} \subseteq \mathcal{S}$ of sets whose union is $\mathcal{U}$ and $|\mathcal{C}| \leq k$?*

It has been proved that SCP is NP-hard [26].

Theorem 1. *There is no polynomial time algorithm for Problem 1 unless P = NP.*

Proof. We prove by contradiction. Suppose otherwise, i.e., there exists a polynomial time algorithm $\mathcal{A}$ for Problem 1. We will show that this algorithm can be used to solve SCP.

Given an instance of SCP with parameters $\mathcal{U}$, $\mathcal{S}$ and k, construct an instance of Problem 1 as Fig. 3. There are $|\mathcal{U}| + 1$ subnets and $|\mathcal{S}|$ candidate hovering locations. Each subnet contains one sensor node, where s_1 is the subnet containing the sink and the remaining $|\mathcal{U}|$ subnets correspond to the elements in $\mathcal{U}$ in SCP. The candidate hovering location corresponds to the subset of $\mathcal{U}$ in $\mathcal{S}$, p_j is concatenated with s_i ($i \geq 2$) if the subset corresponding to p_j contains the element corresponding to s_i. We let the sink be connected to all candidate hovering locations and limit the number of UAVs to k. The construction can clearly be done in polynomial time.

We will prove that, SCP has a positive answer if and only if the optimal solution to the constructed Problem 1 instance has an optimal solution with objective value $|\mathcal{U}|$. To see this, observe that if with algorithm $\mathcal{A}$ it is possible to find k hovering locations such that all subnets are connected to the sink in polynomial time, then there must also exist k corresponding subsets of $\mathcal{S}$ in SCP that can cover all elements in $\mathcal{U}$. Conversely, if SCP has a desired set cover, we could select the hovering locations and get a solution to Problem 1 connecting all subsets.

Thus, if there is a polynomial time algorithm for Problem

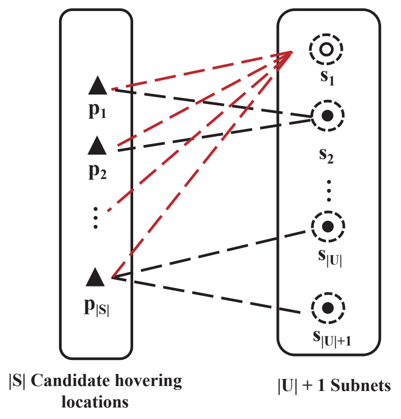


Fig. 3. An instance of Problem 1 constructed for a set covering problem instance

1, we can solve SCP in polynomial time, which is impossible unless P=NP. □

This theorem shows that Problem 1 cannot be solved exactly efficiently. We thus consider heuristic algorithms.

IV. Proposed Location Selection Algorithm

We propose a deployment location selection algorithm. The algorithm has two steps. In the first step, we greedily select relay locations, ignoring the number limitation, until all subnets are connected to the sink. In the second step, we respect the number limitation by removing excess relays. We find that this two-step method works much better than a single run of greedy selection, which will be compared in the evaluation. Meanwhile, we also record the best feasible solution during the first step, in case it is better than the two-step method.

A. Step A: Reconnect All Subnets by Ignoring Number Limitation

In this step, from the candidate hovering locations that can find a path to the sink (that is, s_1), we always select the one that allows more lost nodes to be reconnected. With the selection of candidate locations, some subnets will restore connectivity; we denote the subnets that have been reconnected as CS, and those that remain disconnected as DS. Furthermore, to measure how many lost nodes can be reconnected by selecting a candidate hovering location p_j, we use the total number of nodes contained in the subnets that the UAV is able to communicate with while hovering at p_j and has not regained connectivity, as the weight of that location w_j:

$$w_j = \sum_{s_i \in DS \wedge (i,j) \in E} n_i \tag{2}$$

Initially, only the nodes contained in s_1 can communicate with the sink, i.e., $CS = \{s_1\}$, $DS = \{s_2, s_3, \ldots, s_m\}$. In the first iteration, we calculate the weight of candidate hovering locations connected to s_1, then select the one with the maximal weight. After selection, we move the subnets in DS that are connected to the selected location to CS. In the second iteration, we recalculate the weights of all unselected hovering locations that are connected to the subnets in CS, and select the one with the maximal weight, then update CS and DS. Repeat the above steps, record the selected locations and the number of reconnected nodes in the Cth iteration, and then continue until no more subnets can be reconnected.

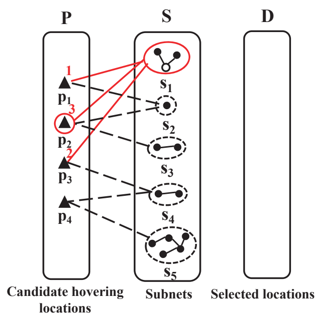
Fig. 4 gives an example with five disconnected subnets and four candidate hovering locations in the area. The UAV limitation number C is 2, and use D to record the hovering locations selected during the process. As shown in Fig. 4(a), initially, $CS = \{s_1\}$, so p_1, p_2 and p_3 are candidates in this iteration. Calculate the weight of these three locations $w_1 = n_1 = 1$, $w_2 = n_2 + n_3 = 3$, $w_3 = n_4 = 2$, so p_2 is selected, that is, $D = \{p_2\}$. In Fig. 4(b), with selection of p_2, $CS = \{s_1, s_2, s_3\}$, the candidates become p_1 and p_3. Recalculate the weight of p_1 and p_3, then select p_3 with $w_3 = 2$. Similarly to previous iterations, in Fig. 4(c) s_4 is added to CS, and the candidates in this iteration are p_1 and p_4. So far, we have selected two deployment locations, which is equal to C. Record the result at this time, including the total number of reconnected nodes 5 and the selected hovering locations p_2 and p_3. Continue to calculate and compare the weights of p_1 and p_4, finally selecting p_4 as the last deployment location, as shown in Fig. 4(d), so that all subnets have been reconnected.

B. Step B: Remove Excess Relays to Satisfy Number Limitation

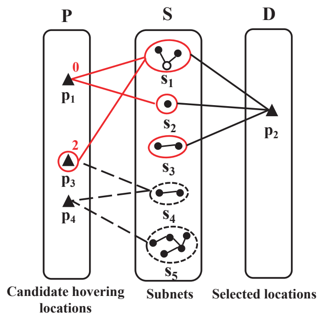
Step A follows the idea of local optimality; therefore, it cannot take a global view to deploy UAVs if there exists a large subnet that needs to be forwarded through multiple smaller subnets to communicate with the sink. Therefore, according to the result of the first step, if the number of UAVs required to connect all subnets is greater than C, we remove excess hovering locations.

For each selected location d_j, we try to remove it and calculate the total number of nodes that can be reconnected through the remaining selected locations, denoted by r_j. We remove the one with maximal r_j. Repeat this step until the number of remaining selected locations equal to C. Compare the number of nodes connected when the hovering locations selected in the first step reaches C with the number of nodes connected after the deletion of the hovering locations exceeding the upper limit in the second step, then take the solution that has the better result.

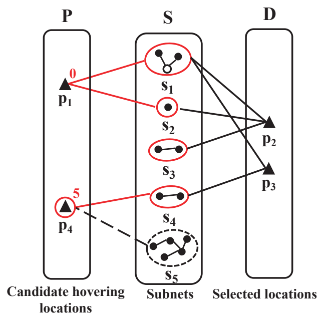
Fig. 5(a) shows the final result of the example in Step A. Since the number of locations in D is greater than C, we need to remove excess locations. When p_4 is removed from D, the number of nodes remaining that can reconnect is 5 (Fig. 5(b)). Similarly, when p_3 and p_2 are removed, the number of remaining nodes that can be reconnected is 3 and 7 (Fig. 5(c) and Fig. 5(d)), so we remove p_2 from D. At this time, the number of remaining deployment locations reaches C, therefore, the final number of nodes that can reconnect in step B is 7. Since the solution of step B is better than the



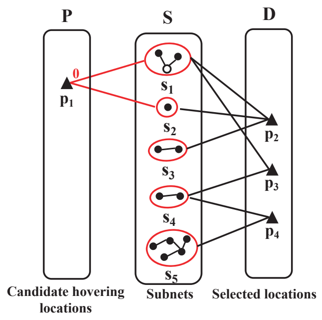
(a) Only s_1 is in CS, so p_2 is selected.



(b) s_1, s_2, s_3 are in CS, so p_3 is selected.

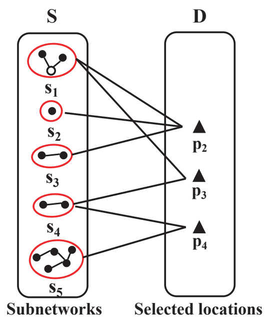


(c) s_1, s_2, s_3, s_4 are in CS, so p_4 is selected.

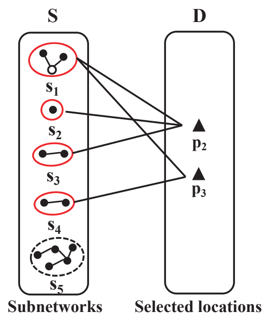


(d) All subnets are reconnected.

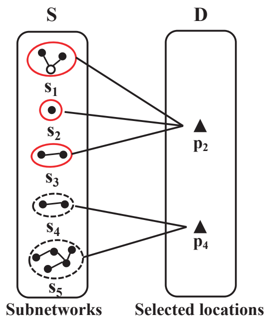
Fig. 4. Illustration of step A: reconnect all subnets.



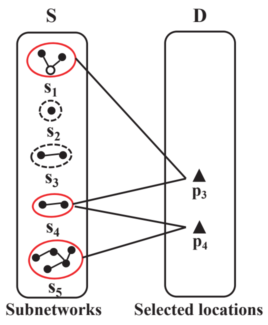
(a) Reconnecting all subnets.



(b) Removing p_4



(c) Removing p_3



(d) Removing p_2

Fig. 5. Illustration of step B: remove excess relays

solution of step A, p_3 and p_4 are finally selected as deployment locations. The result is shown in Fig. 5(d).

C. Putting It Altogether

Algorithm 1 shows the pseudocode of the deployment location selection algorithm. Step A is from line 1 to line 16. In lines 4 to 11, we traverse CS to find the candidate locations in this iteration, then use Eq. 2 to calculate their weights and select the location with the maximal weight. We iterate until no subnets can be connected and record the number of reconnected sensor nodes when the number of selected locations is equal to C. If the number of locations selected in step A is greater than C, we continue with step B in lines 17 to 22, that is, we remove the locations that have the least impact on reconnecting maximal nodes until the number of remaining locations is C. In line 23, we compare the result in step A and B, taking the better solution as the output.

Theorem 2. *Algorithm 1 runs in $O(|E|m^2)$ where m is the number of subnets and $|E|$ is the number of edges.*

Proof. In the worst case, selecting one location only can reconnect one subnet, i.e., the while loop in line 2 has at most $m-1$ iterations. In each iteration, the sizes of CS are 1, 2, $\ldots$, $m-1$. Therefore, the times lines 4 to 6 are executed is less than

$$|E|(1+2+\ldots+(m-1)) = \frac{m(m-1)|E|}{2}.$$

In each iteration, the times line 7 is executed is less than tm, where t is the number of candidate hovering locations, thus the total execution times of line 7 is less than $tm(m-1)$. Therefore, the running time of step A $R_A(m)$ from line 2 to line 16 is

$$R_A(m) \leq \frac{m(m-1)|E|}{2} + tm(m-1) = O((|E|+t)m^2).$$

When the number of UAV relays required to reconnect all subnets is more than C, we also need to execute step B. Since $m-1$ deployment locations were finally selected in step A in the worst case, the while loop in line 18 has at most $m-c$ iterations. As $C \geq 1$, $m-C \leq m-1$. According to Eq. 1, calculating $restReconNodes$ runs in $O(m^2)$, thus the running time of step B $R_B(m)$ is

$$R_B(m) \leq (m-1)m^2 = O(m^3).$$

Therefore the running time of Algorithm 1 $R(m)$ is

$$\begin{aligned} R(m) &= R_A(m) + R_B(m) \\ &\leq O((|E|+t)m^2) + O(m^3) \\ &= O((|E|+t)m^2 + m^3). \end{aligned}$$

Because both m and t are in $O(|E|)$, the theorem is proved. $\square$

Theorem 3. *The approximation ratio of Algorithm 1 is $\frac{\sum_{i\in Q} n_i}{\sum_{i=1}^{m} n_i}$, where Q is the set of the first C subnets containing*

Algorithm 1: A location selection algorithm

```
Input: Graph $G = (S \cup P, E)$, UAV number limitation $C$
Output: a set of deployment locations $D \subseteq P$ with $|D| \le C$
1  $CS \leftarrow \{s_1\}$, $DS \leftarrow S \setminus CS$;
2  while $DS \neq \emptyset$ do
3      $candidates \leftarrow \emptyset$;
4      for each $s_i \in CS$ do
5          if $(i, j) \in E \wedge p_j \notin D$ then
6              Add $p_j$ to the set $candidates$;
7      Use Eq. 2 to calculate the weights of all locations in $candidates$;
8      Find the location $p_r$ with the maximal weight;
9      if $w_r$ = 0 then
10         break;
11     Add $p_r$ to $D$;
12     for $s_i \in DS \wedge (i, r) \in E$ do
13         Move $s_i$ from $DS$ to $CS$;
14     if $|D|$ = $C$ then
15         Calculate the number of reconnected nodes, denoted by $A$;
16         $D' \leftarrow D$;
17 while $|D| > C$ do
18     for each $d_j \in D$ do
19         Calculate the number of reconnected sensor nodes if $D \setminus \{d_j\}$ is selected, denoted by $r_j$;
20     Remove $d_j$ with the maximal $r_j$ from $D$;
21     Update the graph;
22 Calculate the number of reconnected nodes, denoted by $B$;
23 if $B < A$ then
24     $D \leftarrow D'$
```

the least number of nodes, and C is the number limitation of relay UAVs.

Proof. The subgraph output in step A is the maximum connected component containing the sink. Denote the subgraph output in the first step as $G_1 = (S_1 \cup P_1, E_1)$, where $S_1 \subset S, P_1 \subset P, E_1 \subset E$. If there exists $s_i \notin S_1$ but it can be connected to a subnet in S_1 via p_i, then p_i must not belong to P_1. According to the selection rule in the first step, p_i can be selected as a hovering location, contradicting the termination condition of the algorithm. Therefore, s_i can always be reconnected after the first step.

The algorithm can finally connect at least C subnets. According to the deletion rule, each hovering location left is connected to at least one subnet of degree 1. So in the worst case, as the limited number of UAVs is C, the algorithm can connect at least C subnets, and these C subnets can communicate with the sink.

Let N be the total number of lost nodes connected by Algorithm 1, then

$$N \ge \sum_{i \in Q} n_i,$$

where Q is the C smallest subsets, and C is the number of relay UAVs. Let N^* be the total number of lost nodes connected by the optimal solution, then

$$N^* \le \sum_{i=1}^{m} n_i.$$

Therefore, the approximation ratio of Algorithm 1 is

$$\frac{N}{N^*} \ge \frac{\sum_{i \in Q} n_i}{\sum_{i=1}^{m} n_i}.$$

□

Theorem 4. *When the UAV can communicate with the sink when hovering at any candidate hovering location, the approximation ratio of Algorithm 1 is $1 - (\frac{C-1}{C})^C$, where C is the number of relay UAVs.*

Proof. Since the UAV hovering at any candidate hovering location can forward the data to the sink, any location can be selected without depending on the selection of other locations or not. In step A of the algorithm, assume that the difference between the total number of currently connected nodes and the optimal solution at the ith ($i \le C$) iteration is u_i, then there must be at least one hovering position among the C hovering locations selected by the optimal solution that is connected to at least $\frac{u_i}{C}$ nodes in the current network connectivity state. Therefore, according to the greedy rule, we have

$$u_{i+1} \le u_i - \frac{u_i}{C} = \frac{C-1}{C} u_i.$$

So, at the Cth iteration, it holds that

$$u_C \le \frac{C-1}{C} u_{C-1} \le (\frac{C-1}{C})^2 u_{C-2} \le \ldots \le (\frac{C-1}{C})^C u_0.$$

Let N_1 be the record solution in step A, N be the total number of lost nodes connected by Algorithm 1 and N^* be the total number of lost nodes connected by the optimal solution, then

$$u_C = N^* - N_1, u_0 = N^*, N \ge N_1.$$

Therefore,

$$N^* - N \le (\frac{C-1}{C})^C N^*,$$

i.e.,

$$\frac{N}{N*} \ge \frac{N_1}{N*} \ge 1 - (\frac{C-1}{C})^C.$$

□

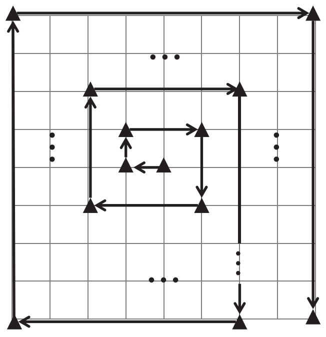


Fig. 6. UAV's surveying trajectory

TABLE II
THE DEFAULT SETTINGS OF SIMULATIONS

UAV flight altitude (H)	10 m
Communication distance (D)	30 m
Projected communication radius (R)	28 m
Square size	$300m \times 300m$
Number of lost nodes	300
Number of UAVs (C)	10
Survey step (L)	1 m

V. SIMULATIONS

In this section, we evaluate our algorithm by simulations. We implement the following methods in Python3:

1) LSA, the proposed algorithm in Algorithm 1;
2) Greedy, a greedy algorithm where the weights of all candidate hovering locations are calculated only once, then select the one with maximal weight from the locations connected to the sink in each iteration;
3) RRA, route reconstruction algorithm [25] adapted to our problem. It preferentially connects the subnet pair that is capable of being connected by a relay UAV and is most closely connected with themselves, then from the candidate locations that can connect the subnet pair, it first selects the one that can connect more subnets and then selects the one with the highest closeness centrality.

The default settings are as follows. The flight altitude of the UAV H is set to 10 m and the communication distance between the sensor node and the UAV D is set to 30 m. Therefore, the communication radius between the UAV and the sensor node R is 28 m. We randomly generate sensor nodes in a $300m \times 300m$ square and set UAV relay number C to 10. The default number of nodes generated is 300. For surveying possible hovering locations, we send a UAV on top of the sensors traveling along a zigzag pattern in Fig. 6. We set the step size L at 1 m. We conduct each experiment ten times and take the average as the final result. Table II summarizes the settings. The setting is consistent with [27].

A. Impact of network size on the number of reconnected nodes

We study the impact of network size. We vary the number of nodes to 100, 150, 200, 250, and 300, and set the number of UAV C to 10.

The results are shown in Fig. 7. As can be seen, LSA gives better results than the other two in all cases. For the Greedy algorithm, since it does not update the weights of the candidate hovering locations before the selection process, it may select an invalid hovering location, so its results are poor. For the RRA algorithm, the result is worse than LSA because it gives priority to connecting the most connected subnet of itself, so it cannot guarantee that the subnet can eventually communicate with the sink.

B. Impact of UAV limitation on the number of reconnected nodes

We study the impact of the UAV limitation number C. We conduct experiments with the number of UAVs at 4, 7, 10, 13, and 16. The results are shown in Fig. 8.

The results show that the deployment locations selected by LSA can reconnect more lost nodes, and the larger C is, the more obvious the difference between the other two algorithms and LSA. Furthermore, the reason why the number of reconnected nodes remains unchanged when the number of UAVs increases from 13 to 16 is that the number of UAVs reaches saturation and the newly increased number of UAVs is the number of redundant UAVs.

C. Impact of candidate locations distribution on the number of reconnected nodes

The surveying step length of the UAV affects the distribution of candidate hovering locations. In this section, we explore the impact of the distribution of candidate locations by changing the survey step length L, and the number of UAVs C is set to 10.

The results are shown in Fig. 9. As the length of the survey step increases, the number of nodes that can be reconnected by LSA gradually decreases, while the number of nodes reconnected by Greedy gradually increases and approaches the result of LSA due to the decrease in the number of candidate hovering locations.

D. Running time with different number of nodes

To further explore the complexity of the algorithm, we conducted experiments on the running time of the program for different sizes of problems. We set the number of UAVs C to 10. The results are shown in Fig. 10.

For the LSA algorithm, the running time is faster in practice because it calculates only the weights of the candidate hovering locations that meet the requirements at each iteration. For the Greedy algorithm, it needs to calculate the weights of all candidate hovering locations, so it is less affected by the number of nodes. For the RRA algorithm, since it needs to calculate how closely a subnet is connected to itself, which

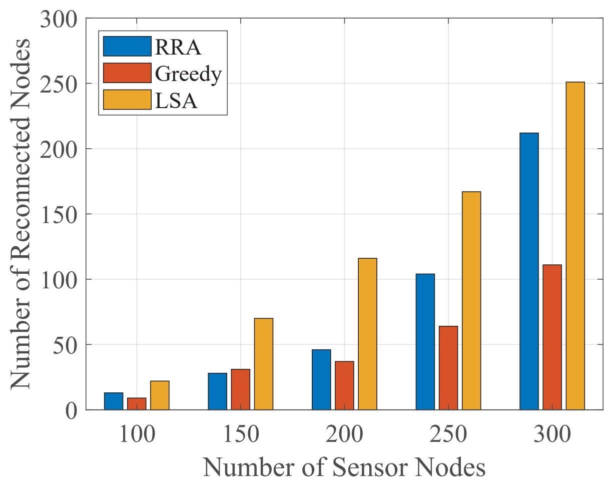


Fig. 7. Impact of network size on the number of reconnected nodes

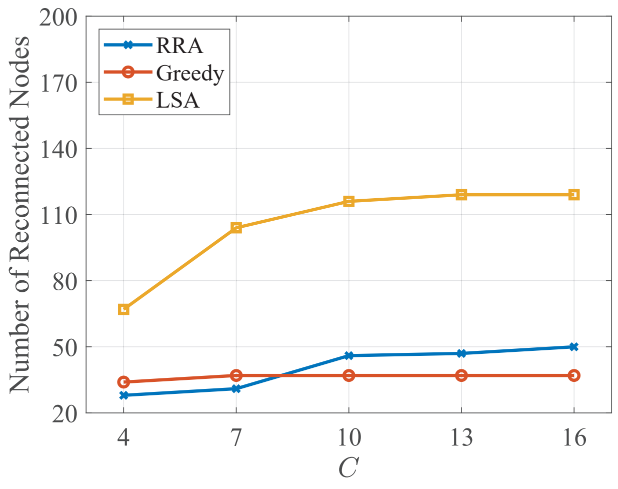


Fig. 8. Impact of UAV number limitation on the number of reconnected nodes

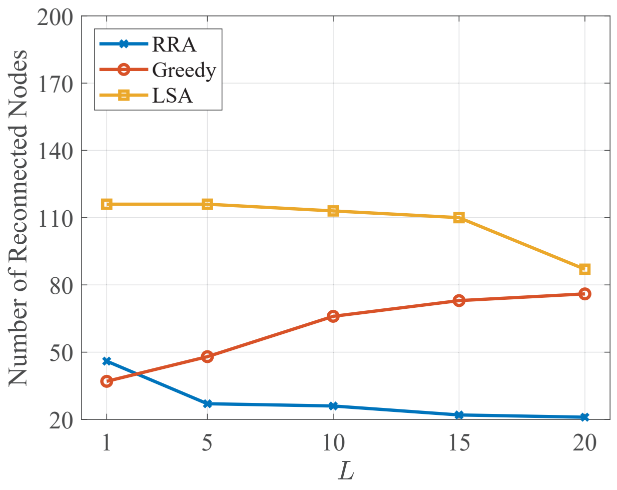


Fig. 9. Impact of candidate locations distribution on the number of reconnected nodes

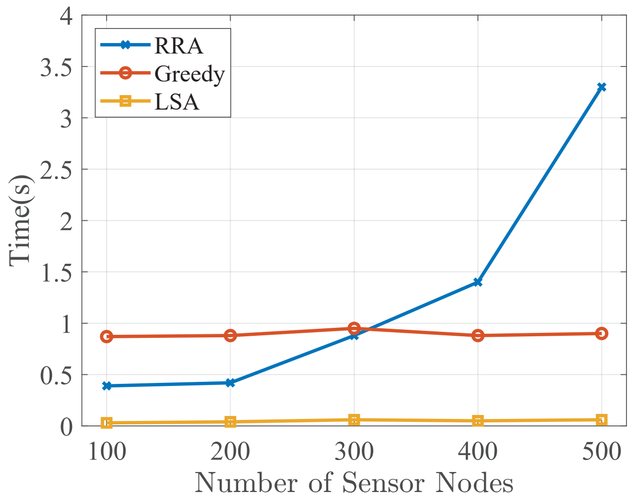


Fig. 10. Running time with different number of sensor nodes

has a large computational complexity, it is mainly influenced by the number of sensor nodes.

VI. Conclusion

This paper considers a cooperative approach in which sensors form subnets to facilitate data transmission using UAV relays. Under this model, we formulate the problem, prove it is NP-hard, and propose a heuristic algorithm to maximize the number of lost nodes that can re-find a routing path to the sink. For any disconnected area, we first form the nodes into subnets and then select the deployment locations for UAV relays based on the connectivity between the subnets and the UAV hovering at the candidate locations. The algorithm is proven to run in polynomial time. Its approximation ratio is analyzed. Simulations show that our algorithm is better at reconnecting lost nodes in different situations. In the future, we will further optimize the algorithm and find the optimal solution based on the branch-and-bound approach.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61972199, 61931011), the Fundamental Research Funds for the Central Universities (NP2022415), and the open research fund for a graduate student in the Nanjing University of Aeronautics and Astronautics (xcxjh20221610).

References

[1] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, “A survey of network lifetime maximization techniques in wireless sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

[2] F. Engmann, F. A. Katsriku, J.-D. Abdulai, K. S. Adu-Manu, and F. K. Banaseka, “Prolonging the lifetime of wireless sensor networks: a review of current techniques,” *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[3] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A tutorial on uavs for wireless networks: Applications, challenges, and open problems,” *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[4] Y. Zeng, J. Lyu, and R. Zhang, “Cellular-connected uav: Potential, challenges, and promising technologies,” *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2018.

[5] N. Zhao, F. Cheng, F. R. Yu, J. Tang, Y. Chen, G. Gui, and H. Sari, “Caching uav assisted secure transmission in hyper-dense networks based on interference alignment,” *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 2281–2294, 2018.

[6] J. Fei, X. Zhu, and R. Zhang, “Evaluation of tcp variants on dynamic uav networks,” in *Proceedings of CSCWD*, 2023.

[7] S. Liu, C. Dong, X. Zhu, J. Tang, and L. Zhang, “Performance evaluation of batman-adv protocol on convergecast communication in UAV networks,” in *Proceedings of GLOBECOM*. IEEE, 2022, pp. 5105–5110.

[8] X. Zhu and S. Tang, “A Branch-and-Bound Algorithm for Building Optimal Data Gathering Tree in Wireless Sensor Networks,” *INFORMS Journal on Computing*, vol. 33, no. 4, pp. 1446–1460, 2021.

[9] ——, “Exact Algorithms for the Minimum Load Spanning Tree Problem,” *INFORMS Journal on Computing*, vol. 33, no. 4, pp. 1431–1445, 2021.

[10] X. Zhu, G. Chen, S. Tang, X. Wu, and B. Chen, “Fast approximation algorithm for maximum lifetime aggregation trees in wireless sensor networks,” *INFORMS Journal on Computing*, vol. 28, no. 3, pp. 417–431, 2016.

[11] L. Wang, H. Zhang, S. Guo, and D. Yuan, “Deployment and association of multiple uavs in uav-assisted cellular networks with the knowledge of statistical user position,” *IEEE Transactions on Wireless Communications*, 2022.

[12] X. Zhong, Y. Guo, N. Li, and Y. Chen, “Joint optimization of relay deployment, channel allocation, and relay assignment for uavs-aided d2d networks,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 804–817, 2020.

[13] X. Zhong, Y. Guo, N. Li, and S. Li, “Deployment optimization of uav relays for collecting data from sensors: A potential game approach,” *IEEE Access*, vol. 7, pp. 182 962–182 973, 2019.

[14] T. Kimura and M. Ogura, “Distributed collaborative 3d-deployment of uav base stations for on-demand coverage,” in *Proceeding of IEEE INFOCOM*. IEEE, 2020.

[15] J. Hu, H. Zhang, Y. Liu, X. Li, and H. Ji, “An intelligent uav deployment scheme for load balance in small cell networks using machine learning,” in *Proceeding of IEEE WCNC*. IEEE, 2019.

[16] Q. Zhang, W. Saad, M. Bennis, X. Lu, M. Debbah, and W. Zuo, “Predictive deployment of uav base stations in wireless networks: Machine learning meets contract theory,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 637–652, 2020.

[17] Q. Wu, Y. Zeng, and R. Zhang, “Joint trajectory and communication design for multi-uav enabled wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[18] S. Fu, Y. Tang, N. Zhang, L. Zhao, S. Wu, and X. Jian, “Joint unmanned aerial vehicle (uav) deployment and power control for internet of things networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4367–4378, 2020.

[19] Y. Chen, N. Zhao, Z. Ding, and M.-S. Alouini, “Multiple uavs as relays: Multi-hop single link versus multiple dual-hop links,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6348–6359, 2018.

[20] S. Sapre and S. Mini, “Optimized relay nodes positioning to achieve full connectivity in wireless sensor networks,” *Wireless Personal Communications*, vol. 99, no. 4, pp. 1521–1540, 2018.

[21] H. Liu, P. Wan, and X. Jia, “On optimal placement of relay nodes for reliable connectivity in wireless sensor networks,” *Journal of Combinatorial Optimization*, vol. 11, no. 2, pp. 249–260, 2006.

[22] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, “Relay sensor placement in wireless sensor networks,” *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[23] S. Lee and M. Younis, “Optimized relay node placement for connecting disjoint wireless sensor networks,” *Computer Networks*, vol. 56, no. 12, pp. 2788–2804, 2012.

[24] S.-Y. Park, C. S. Shin, D. Jeong, and H. Lee, “Dronenetx: Network reconstruction through connectivity probing and relay deployment by multiple uavs in ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 192–11 207, 2018.

[25] C. S. Shin, S.-Y. Park, J. Yoon, and H. Lee, “Progressive ad-hoc route reconstruction using distributed uav relays after a large-scale failure,” in *Proceeding of IEEE WCNC*. IEEE, 2018.

[26] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of Np-Completeness*. W H Freeman and Co, 1979.

[27] X. Li, X. Zhu, and C. Dong, “Planning path for uavs to collect data from disconnected sensor networks,” in *Proceeding of ICPADS*. IEEE, 2021.