Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 952069, 14 pages
http://dx.doi.org/10.1155/2015/952069

Research Article

Maintaining Connectivity of MANETs through Multiple Unmanned Aerial Vehicles

Ming Zhu, Fei Liu, Zhiping Cai, and Ming Xu

College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Zhiping Cai; zpcai@nudt.edu.cn

Received 8 June 2015; Revised 14 October 2015; Accepted 22 October 2015

Academic Editor: Gregory Chagnon

Recently, Unmanned Aerial Vehicles (UAVs) have emerged as relay platforms to maintain the connectivity of ground mobile ad hoc networks (MANETs). However, when deploying UAVs, existing methods have not consider one situation that there are already some UAVs deployed in the field. In this paper, we study a problem jointing the motion control of existing UAVs and the deployment of new UAVs so that the number of new deployed UAVs to maintain the connectivity of ground MANETs can be minimized. We firstly formulate the problem as a Minimum Steiner Tree problem with Existing Mobile Steiner points under Edge Length Bound constraints (MST-EMSELB) and prove the NP completeness of this problem. Then we propose three Existing UAVs Aware (EUA) approximate algorithms for the MST-EMSELB problem: Deploy-Before-Movement (DBM), Move-Before-Deployment (MBD), and Deploy-Across-Movement (DAM) algorithms. Both DBM and MBD algorithm decouple the joint problem and solve the deployment and movement problem one after another, while DAM algorithm optimizes the deployment and motion control problem crosswise and solves these two problems simultaneously. Simulation results demonstrate that all EUA algorithms have better performance than non-EUA algorithm. The DAM algorithm has better performance in all scenarios than MBD and DBM ones. Compared with DBM algorithm, the DAM algorithm can reduce at most 70% of the new UAVs number.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as promising relay platforms to improve networking performance (such as connectivity and throughput) for ground mobile ad hoc networks (MANETs) [1]. UAVs have some unique characteristics suitable for the relaying task. Firstly, the motion flexibility of UAVs can expand the scope of ground-based networks especially in scenarios with obstacles. Secondly, UAVs can communicate with ground nodes in the line of sight, which can improve the link capacity between ground nodes. Last but not least, UAVs are often integrated with Communication, Computation, and Control (3C) system and various sensors, so that UAVs can be aware of the environment and control their motion adaptively. The adaptability of UAVs makes them suitable in providing relay service for MANETs that has dynamic network topology.

A variety of efforts have been made to explore the benefits of using UAVs as communication relays for ground MANETs. Some work wants to optimize the deployment of UAVs to improve the connectivity of ground nodes. Chandrashekar et al. presented a method on deploying minimum number of UAVs to connect a disconnected MANET [2]. Some works study the motion control of UAVs to improve the link capacity of ground nodes. Jiang and Swindlehurst proposed a UAV's heading control algorithm that can maximize the link capacity of ground-to-air uplink channel using a multiantenna UAV [3]. Dixon and Frew proposed a motion control method for using chains of UAVs to improve the link capacity between two isolated ground nodes [4]. Some work considers both the deployment and motion control of UAVs. For example, Han et al. considered both the deployment and motion control of a single UAV to improve the connectivity of ground MANETs [1].

However, existing works on the deployment of UAVs have not considered a situation that some UAVs have already been deployed in the field. With the movement of ground MANETs, existing UAVs may fail to connect all ground nodes. New UAVs need to be supplied to maintain the connectivity of ground MANETs. In order to minimize the

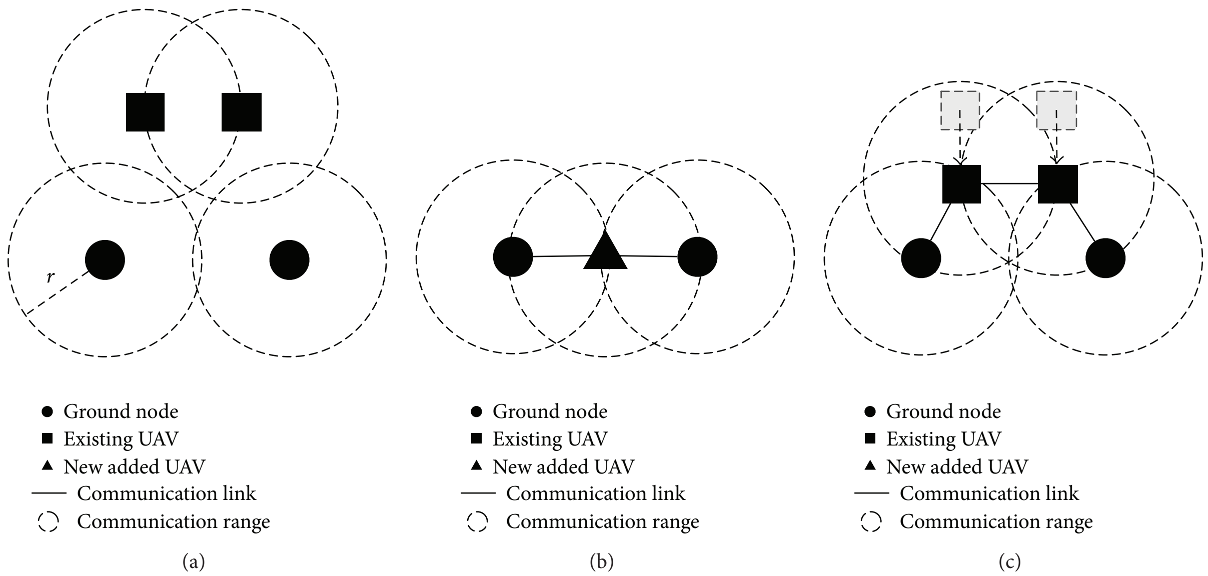


Figure 1: An example that illustrates the importance of existing UAVs in maintaining the connectivity of ground MANETs.

number of new added UAVs, both the movement of existing UAVs and the deployment of new UAVs need to be considered. This is a joint optimization problem that optimizes both the deployment and motion control of multiple UAVs.

We consider the usage of existing UAVs by moving them to proper positions so that the number of new UAVs that is needed can be reduced. The existing UAVs have a limited motion range that depends on the speed of UAVs. In order to support bidirectional communication between UAVs and ground nodes, we assume that UAVs have the same communication range as ground nodes. Figure 1 shows an example of how motion control of existing UAVs may reduce the number of needed new UAVs. Suppose that there are two ground nodes and two existing UAVs deployed in the field. Since the distance between two ground nodes is larger than their communication range r, the ground MANET is separated into two parts which is shown in Figure 1(a).

In order to maintain the connectivity of ground MANETs, methods that do not consider existing UAVs such as [2] will add new UAVs to connect partitioned parts as shown in Figure 1(b). Here a new UAV is added and deployed in the middle of two ground nodes. So these two ground nodes can now communicate with each other with the help of the new UAV. If we do not consider existing UAVs, at least one additional UAV needs to be deployed to maintain the connectivity of ground MANETs.

To reduce capital expenditure, users would try their best to reduce the number of newly deployed UAVs. In other words, they will exploit the existing UAVs instead of ignoring them. By moving existing UAVs to proper positions and using existing UAVs as relays, the connectivity of ground MANETs can be improved. Figure 1(c) shows one proper movement. Move these two existing UAVs directly towards the line that consisted of two ground nodes until the distance between existing UAVs and one ground node is less than the communication range r. Then a communication link is set up between two ground nodes. Thus the connectivity of ground MANETs is maintained and no new UAVs are needed to be deployed.

In this paper, we study the joint optimization problem of deployment and motion control of multiple UAVs so that the number of new added UAVs can be minimized. We firstly formulate this problem as a Minimum Steiner Tree problem with Existing Mobile Steiner nodes under Edge Length Bound constraints (MST-EMSELB) and prove the NP completeness of the problem. Then we present non-Existing UAVs Aware (non-EUA) algorithm and propose three Existing UAVs Aware (EUA) polynomial time approximation algorithms: Deploy-Before-Movement (DBM), Move-Before-Deployment (MBD), and Deploy-Across-Movement (DAM). The first two algorithms decouple the joint problem into deployment problem of new UAVs and motion control problem of existing UAVs. DBM algorithm optimizes the deployment of new UAVs before movement of existing UAVs and the MBD algorithm solves the problem contrarily. DAM algorithm is a mixed algorithm that solves the movement and deployment problem crossly. Simulation experiments show that all EUA algorithms have better performance in terms of new UAVs' number than non-EUA algorithm. DAM algorithm is always better than DBM and MBD algorithms and can improve the performance to 70% at most comparing with DBM algorithm.

The main contributions of this paper are as follows:

(1) We investigate a new problem in maintaining the connectivity of ground MANETs using multiple UAVs, which joints the deployment problem of new added UAVs and motion control problem of existing UAVs. We demonstrate the significance of considering existing UAVs to reduce the number of new added UAVs.

(2) We formulate the problem as a Minimum Steiner Tree problem with Existing Mobile Steiner nodes under Edge Length Bound constraints (MST-EMSELB) and

prove the NP completeness of the problem. Then, we propose three polynomial time approximation algorithms.

(3) We compare the performance of proposed algorithms with non-EUA algorithms in simulation environment and demonstrate the effectiveness of proposed algorithms.

The rest of this paper is organized as follows. Section 2 presents some related work. We present the system model in Section 3 and formulate the problem in Section 4. In Section 5, we present three polynomial time approximation algorithms for the problem. We demonstrate the performance of proposed algorithm through simulation in Section 6. Section 7 concludes this work.

2. Related Work

Related works lie in two research fields: static relay deployment problem and mobile relay motion control problem.

2.1. Static Relay Deployment Problem. Static relay deployment problems have been widely studied in wireless sensor networks (WSN). In WSN, it is hard to recharge the sensors after deployment. Due to the unbalanced load of message routing, some sensors will run out of energy before others. Then network partition happens and the whole network may be not available even though some sensors are still alive.

To extend the lifetime of WSN, some researches propose to deploy some static relay nodes in the field. Lloyd and Xue studied the problem of deploying minimum number of relay nodes so that, for each pair of sensor nodes, there is a connecting path consisting of relay and/or sensor nodes [5]. They proved that the problem is NP complete and propose a polynomial time 7-approximation algorithm for the problem. Zhang et al. studied four related fault-tolerant relay node placement problems, discussed their computational complexity, and present a polynomial time O(1)-approximation algorithm with a small approximation ratio [6]. Based on previous research, Misra et al. studied the constrained versions of the relay node placement problem, where relay nodes can only be placed at a set of candidate locations [7]. They also proposed a polynomial time O(1)-approximation algorithm for the problem. Lee and Younis extend the usage of relay node to federating disjoint segments of WSN and proposed a distributed cell-based optimized relay node placement (CORP) algorithm [8]. Marinho et al. studied the use of UAVs and cooperative multiple input multiple output (MIMO) techniques to keep the WSN connected [9].

2.2. Mobile Relay Motion Control Problem. Similar to static relays, mobile relays were also proposed to extend the lifetime of WSN. But mobile relays often have rich resources and are able to move in the field. Thus mobile relays are more flexible than static relays and can be reused in different positions.

For WSN, the main purpose of using mobile relays is to save energy consumption of sensors and to extend the lifetime of the network. Wang et al. first studied the performance of a large dense network with one mobile relay and proposed a joint mobility and routing algorithm which can yield a network lifetime close to the upper bound [10]. Venkateswaran et al. proposed a novel relay deployment framework that utilizes mobility prediction of MANET nodes to optimally define the movement of the relay nodes [11]. Their simulation results indicate significant energy savings. El-Moukaddem et al. studied the problem of using mobile relays in data-intensive WSN to save the energy consumption of the whole network [12]. They consider the energy consumption of both mobility and wireless transmissions.

As airborne platform of mobile relay, UAVs have been introduced to WSN, MANET, and other kinds of ground networks to improve the connectivity or link capacity between ground nodes. Since UAVs have relatively high mobility, sensibility, and self-controllability, they are especially suitable for MANET that has dynamic topology.

Chandrashekar et al. considered the problem of providing full connectivity to disconnected ground MANET nodes by dynamically placing Unmanned Aerial Vehicles (UAVs) to act as relay nodes [2]. But they did not consider the mobility control problem of deployed UAVs. Jiang and Swindlehurst considered using a multiantenna UAV to connect a collection of single-antenna ground nodes [3]. By dynamically adjusting the UAV heading they can maximize the approximate sum rate of the ground-to-air uplink channel. Han et al. considered using one UAV to improve the connection of ground-based wireless ad hoc networks [1]. The location and movement of the UAV are optimized to improve four types of network connectivity including global message connectivity, worst case connectivity, network bisection connectivity, and k-connectivity. Both of these two works just consider using a single UAV.

With the development of UAV's manufacturing technology, the size of UAV becomes smaller and the price also becomes cheaper. Thus it is possible to use a team of UAVs to provide network connection for ground nodes or improve their link capacity. Zhan et al. investigated a communication system in which UAVs are used as relays between ground-based terminals and a network base station [13]. They developed an algorithm for optimizing the performance of the ground-to-relay links through control of the UAV heading angle. Cetin and Zagli studied UAVs' motion control to achieve continuous long-range communication relay infrastructure [14]. They have proposed a novel dynamic approach to maintain the communication between vehicles. Besides dynamically keeping vehicles in range and appropriate position to maintain communication relay, artificial potential field based path planning also provides collision avoidance system. Ponda et al. presented a cooperative distributed planning algorithm that ensures network connectivity for a team of heterogeneous agents operating in dynamic and communication limited environments [15]. The algorithm predicts the network topology and proposes relay tasks to repair connectivity violations. Dixon and Frew considered using chains of UAVs to improve the connectivity for two isolated ground nodes [4]. The mobility of UAVs is controlled to maximize the communication link capacity for the end-to-end connection. But they just assume that there are two nodes in ground MANETs.

3. System Model

We assume a scenario that multiple UAVs are used to maintain the connectivity of ground MANETs and some UAVs have already been deployed in the field. However, due to the movement of ground nodes and limited communication range, existing UAVs are not able to connect all ground nodes. Thus we need to add new UAVs to maintain the full connectivity of MANETs. The system model is shown in Figure 2. In the figure, vehicles represent ground nodes. As we can see, there are already two UAVs deployed in the field, but, due to the long distance between vehicles, the ground network is partitioned into two parts. In order to keep the full connectivity, a new UAV is added as relay that connects these two partitioned subnetworks.

3.1. Mobility Model of UAVs. We assume the UAVs used in this paper are small four-rotor UAVs. The four-rotor UAVs can stay in a constant position and fly directly up and down. It can also spin 360 degrees around itself with a zero radius. To simplify the system, we assume all UAVs fly in different altitude so that collision avoidance of UAVs needs not to be considered in this paper.

Since the four-rotor UAV is small and always uses battery as energy, the velocity of the UAVs is limited. Because ground nodes are continuously moving, the task of motion control of existing UAVs and deployment of new UAVs must be finished within a given deadline. This requirement is especially important for some military scenarios. So we assume in this paper the mobility of existing UAVs is constrained. They can move towards any direction. However, the distance between new positions and current positions must not be more than a constant length. And we call this constant length motion range in this paper.

3.2. Communication Model. Connectivity represents the communication capability between nodes in a network. Here, the link capacity is used to represent the connectivity. The meaning of maintaining the connectivity of ground MANETs is to keep the link capacity higher than a given threshold. When the link capacity between two nodes is higher than the threshold, these two nodes are connected. Otherwise, they are disconnected.

Link capacity is the upper bound of data rate when transmitting data. According to Shannon equation the link capacity can be computed using (1). Here, C is the link capacity in b/s; B is the bandwidth and SNR is the signal noise ratio

$$C = B\log_2(1 + \mathrm{SNR}). \tag{1}$$

Signal noise ratio (SNR) is the ratio of receiving power of signal with power of noise signal. The SNR of node i from node j is defined as (2) [16]. Here, P_j is the sending power, $d_{i,j}$ is the distance between node i and node j, and α is the path loss. N_0 is the power density of noise and here we consider the Gaussian white noise, so $N_0 = 4 * 10^{-21}$ Hz. B is the bandwidth

$$\mathrm{SNR}_{i,j} = \frac{S_{i,j}}{N_i} = \frac{P_j/d_{i,j}^{\alpha}}{N_0 * B}. \tag{2}$$

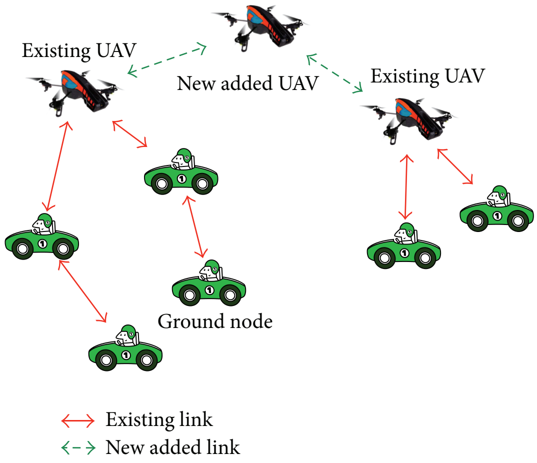


Figure 2: System model.

According to (1) and (2), we can find that there are two factors that affect the connectivity. One factor is the distance between two nodes. The larger the distance is, the weaker the link capacity will be. When distance is large enough, the link capacity will be smaller than the given threshold and these two nodes will be disconnected. The other factor is the path loss, which reflects the signal interference of the environment. Different environment has different path loss. Signal loses more power when it is transmitted near the ground than that in the air. So when two nodes have the same distance, the link capacity between these two nodes is higher when these two nodes are in the air than that when they are on the ground.

In this paper, we define the connectivity between any two nodes as a binary variable. If the distance between two nodes is not more than a constant length, we assume these two nodes are connected. Otherwise, these two nodes are disconnected. Here we name the constant length communication range. The communication range of ground nodes is smaller than the communication range between two UAVs or between one UAV and one ground node.

4. Problem Formulation

In our definitions, we assume that all current positions of ground nodes and existing UAVs are known. We also assume there are no obstacles that affect the mobility of UAVs or transmissions. Our problem can be described as follows: given a set of ground nodes and a set of existing UAVs, we want to find new positions for existing UAVs and positions for new added UAVs to form tree spanning all ground nodes so that the number of new added UAVs is minimized. There are two constraints in this problem. One is the distance between new position and current position of each existing UAV that is not more than a given motion range. The other is that length of each edge in the tree is no more than a given communication range.

4.1. Formulated Problem of MST-EMSELB. Since this problem is similar to Steiner Tree Problem with minimum number of Steiner points, we formulate this problem as a Minimum Steiner Tree problem with Existing Mobile Steiner points under Edge Length Bound constraints (MST-EMSELB). The Steiner points here stand for UAVs and the Edge Length Bound is the communication range. The formal definition of MST-EMSELB problem is shown as follows.

Given. There are a set of ground nodes P, a set of existing UAVs Q, a motion range l, ground node communication range r, and ground-air communication range R. Here, $r < R$,

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_n\}, \\ Q &= \{q_1, q_2, \dots, q_m\}. \end{aligned} \tag{3}$$

Output. There are new positions of exiting UAVs R, positions of new added UAVs S, and a tree T spanning P:

$$\begin{aligned} R &= \{r_1, r_2, \dots, r_m\}, \\ S &= \{s_1, s_2, \dots, s_k\}, \\ T &= \{P \cup R \cup S, E\}. \end{aligned} \tag{4}$$

Objective. Consider

$$\min(k). \tag{5}$$

Constraints. Consider

$$\begin{aligned} &\left|e_{i,j}\right| \le r, \quad \left(e_{i,j} \in E,\ i, j \in P\right), \\ &\left|e_{i,j}\right| \le R, \quad \left(e_{i,j} \in E,\ i \in P,\ j \in R \cup S\right), \\ &\left|r_i - q_i\right| \le l, \quad 1 \le i \le m. \end{aligned} \tag{6}$$

4.2. NP Completeness of MST-EMSELB. In this section, we will prove that the decision version of the MST-EMSELB problem is NP complete. The NP hardness of the problem is proved by a polynomial time reduction from the Minimum Steiner Tree problem with Edge Length Bound constraints (MST-ELB) problem which is proved to be NP complete [17]. We firstly introduce the decision version of both MST-ELB problem and MST-EMSELB problem. Then we present the reduction from MST-ELB to STP-EMSELB.

4.2.1. Decision Version of MST-ELB Problem. Given a set P of n terminal points in the two-dimensional Euclidean plane $\mathbb{R}^2$, a positive constant L, and a nonnegative integer V, the MST-ELB problem asks whether there exists a tree spanning a point set $Q \supseteq P$ such that each edge in the tree has a length not greater than L and the number of Steiner points (points in $Q \cap \neg P$) is less than or equal to V. Lin and Xue [17] have proved that the MST-ELB problem is NP complete and presented a polynomial time approximation algorithm whose worst case performance ratio is 4 [18].

4.2.2. Decision Version of MST-EMSELB Problem. Given a set A of x terminal points and a set B of y mobile Steiner points in the two-dimensional Euclidean plane $\mathbb{R}^2$, a nonnegative constant r, a positive constant R, and a nonnegative integer W, the MST-EMSELB problem firstly asks whether there exists a motion of mobile Steiner points in set B such that distance between new position and old position of each mobile Steiner point is no more than r. Then, the MST-EMSELB problem will check if there exists a tree spanning a point set $C \supseteq A$ such that the length of each edge in the tree is no more than R and the number of new added Steiner points (points in $C \cap \neg(A \cup B)$) is no more than W.

Theorem 1. *There is a polynomial time reduction from MST-ELB problem to MST-EMSELB problem.*

Theorem 2. *MST-EMSELB problem is NP complete.*

Given the number of new added Steiner points and a topology which specifies the edges in the final tree, a bottleneck tree under this given topology and edge bound constraint can be computed in polynomial time [19] and the mobility constraint of mobile Steiner points can be checked in polynomial time. Therefore MST-EMSELB problem belongs to the class NP. It follows from this remark and Theorem 1 that we have proved Theorem 2. Detailed proofing of Theorem 1 can be found in our previously published paper in [20].

5. Heuristic Solution

As previously mentioned, the MST-EMSELB problem belongs to NP complete problem; thus we try to find polynomial time approximation algorithms for this problem. In this section, we firstly present existing methods and then propose three heuristic algorithms for MST-EMSELB problem.

5.1. Non-EUA Algorithm. Currently, there are no particular algorithms designed for MST-EMSELB problem. The most related problem of MST-EMSELB is MST-EMS problem. When we crystallize the MST-EMS problem in the scenarios of using UAVs to maintain the connectivity of MANETs, a specific problem of MST-EMS is using new UAVs to maintain the connectivity of MANETs without considering existing UAVs.

Lin and Xue presented a minimum spanning tree (MST) based heuristic algorithm for MST-EMS problem whose worst case approximation ratio is 4 [18]. The MST heuristic algorithm firstly generates a minimum spanning tree over P. It then divides each edge e in the tree into small pieces of length at most R by inserting $\lceil l(e)/R \rceil - 1$ degree-2 Steiner points so that all pieces in edge e have equal length. Here $l(e)$ is the Euclidean length of edge e.

Since no mobile Steiner points are considered, the MST heuristic algorithm cannot be directly used for MST-EMSELB problem. Here we just take the Lin and Xue methods as a comparative method. Since this method has not considered the existing UAVs, we call this method non-Existing UAVs Aware (non-EUA) algorithm. Non-EUA algorithm just

```
Input: $P, r, R$
    // $P$ stands for a set of $n$ terminals, $r$ is the edge bound between terminals
    and $R$ is the edge bound between terminals and the Steiner points.
Output: $S$ //the set of Steiner points
Initially: $S \leftarrow \Phi$;
    // The set of Steiner points is initially empty.
generate a complete graph $G(V, E)$ over $P$;
compute a minimum spanning tree $T$ based on $G$;
for each edge $e_i$ in $T$ do
    if $|e_i| > r$ then
        if $|e_i| \leq R$ then
            put 1 Steiner points into $s_i$;
        else
            put $\lceil l(e_i)/R \rceil - 1$ Steiner points into $s_i$;
        end if
        $S \leftarrow s_i$;
    end if
end for
return $S$;
```

Algorithm 1: Non-EUA algorithm.

computes minimum number of new UAVs that is needed to connect all ground nodes. None of the existing UAVs will be reused for connecting ground MANETs. So the number of needed new UAVs computed by non-EUA should be an upper bound of other Existing UAVs Aware algorithms. The non-EUA algorithm is shown in Algorithm 1.

5.2. DBM Algorithm. Since this is a joint optimization problem and there are two variables that need to be optimized, one variable is the new position of existing UAVs and the other variable is the position of new added UAVs. So we decouple the MST-EMSELB problem into two subproblems: the movement control problem of existing UAVs and the deployment problem of new added UAVs. In order to optimize the joint problem, we solve these two subproblems one by one. The first algorithm we proposed is Deploy-Before-Movement (DBM) algorithm that firstly optimizes the deployment of new UAVs and then optimizes the movement control of existing UAVs.

The main idea of DBM algorithm is shown as follows. Firstly we use the non-EUA algorithm to generate candidate positions of new added UAVs without considering existing UAVs. Then we match existing UAVs with candidate positions of new added UAVs. A match between an existing UAV and a candidate position of a new added UAV means that the new added UAV will be replaced by the existing UAV by moving this existing UAV to the candidate position. Since the motion range of existing UAVs is limited, the number of matches is also constrained. Here, we use Hungary algorithm to find maximum matches so that the number of needed new UAVs can be minimized. The DBM algorithm is shown in Algorithm 2.

Although DBM algorithm can utilize the mobility of existing UAVs, the candidate position for moved existing UAVs is limited which is the position computed by non-EUA algorithm. Due to the motion range limitation, existing UAVs may not be able to move to these candidate positions. Figure 3 shows one worst case for DBM algorithm. The worst case scenario is similar to the example showed in Figure 1. But the motion range of existing UAVs is constrained in this scenario. In Figure 3(a), the radius of motion range of existing UAVs is l. When using DBM algorithm, the candidate position of new UAVs is the middle point of two ground nodes as shown in Figure 3(b). The distance from one existing UAV to the candidate position is d and in this case d is larger than l. Thus, none of these two existing UAVs can be matched to the candidate position. So the minimum number of needed new UAVs computed by DBM algorithm is 1.

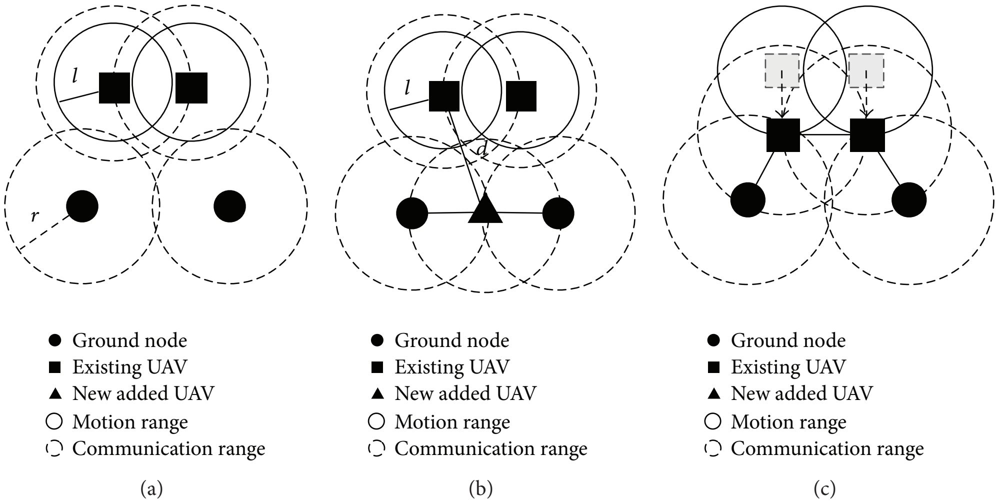
However, the best solution for this case is shown in Figure 3(c). The best movement of existing UAVs is that they all move directly towards the line that consisted of two ground nodes until meeting the motion range. A communication link will be set up for two ground nodes using two existing UAVs as relays and no new UAVs are needed in this solution. The worst case indicates that DBM algorithm cannot make best use of existing UAVs due to the less optimal motion control.

5.3. MBD Algorithm. As previously mentioned, we decouple the joint optimization problem into two subproblems: the deployment of new UAVs problem and motion control problem of existing UAVs. DBM algorithm firstly solves the deployment problem and then solves the motion control problem. However, due to the less optimal motion control, DBM algorithm encounters some worst cases that none of existing UAVs can be reused. So we reverse the solution and propose Move-Before-Deployment (MBD) algorithm that firstly solves the movement problem and then solves the deployment problem.

The main idea of MBD algorithm is as follows. Firstly, we use a heuristic function to generate new positions of existing UAVs Q'. Then, we merge the set of ground nodes P and the set of existing UAVs with new positions Q' into a big node set $P \cup Q'$. After that, we generate a minimum spanning tree

```
Input: $P, Q, l, r, R$
    // $P$ is the position of ground nodes, $Q$ is the position of existing UAVs, $l$ is the motion-range of existing UAVs,
    $r$ is the communication-range between ground nodes and $R$ is the communication-range between ground nodes and UAVs.
Output: $Q', S$ //$Q'$ is the new position of existing UAVs and $S$ is the position of new added UAVs
Initially: $Q' \leftarrow Q$; $S \leftarrow \Phi$;
generate candidate positions $T$ of new UAVs using non-EUA($P, r, R$) algorithm;
cost $\leftarrow \Phi$;
for each pair $q_i \in Q$ and $t_j \in T$ do
    if distance($q_i, t_j$) $\le l$ then
        cost[$i$][$j$] = distance($q_i, t_j$);
    else
        cost[$i$][$j$] = $+\infty$;
    end if
end for
compute maximum match $M$ using hungary($Q, T, Cost$);
for each $t_i \in T$ do
    if there is a UAV $q'_j \in Q'$ and $m_{i,j} \in M$ that $m_{i,j}$ == True then
        $q'_j \leftarrow t_i$;
    else
        $S \leftarrow t_i$;
    end if
end for
return $Q', S$;
```

Algorithm 2: Deploy-Before-Movement (DBM) algorithm.



(a)

(b)

(c)

Figure 3: A worst case for DBM algorithm.

T over set $P \cup Q'$ and then existing UAVs cut process will be used to cut all 1-degree existing UAVs in the tree T until all existing UAVs in the tree have at least 2 degrees. For the rest subtree T' of T, new UAVs will be added to edges of T' that has larger length than r.

The heuristic function mentioned in previous paragraph to generate new positions of existing UAVs might affect the performance of the whole MBD algorithm. We find that the DBM algorithm will be a perfect heuristic function to generate new positions for existing UAVs if the motion constraints of UAVs are released. This is because when the motion constraints of existing UAVs are released, they can move to any positions. So all of existing UAVs can match any candidate positions of new UAVs, since these candidate positions are computed by non-EUA algorithm that can minimize the number of new UAVs to maintain the connectivity of ground MANETs without considering existing UAVs. Thus, DBM algorithm can find the best positions for existing UAVs. The MBD algorithm is shown in Algorithm 3.

In MBD algorithm, the existing UAVs cut process is an important and necessary process to minimize the number of new added UAVs, because when constructing the minimum

```
Input: $P, Q, l, r, R$
    // $P$ is the position of ground nodes, $Q$ is the position of existing UAVs, $l$ is the motion-range of existing UAVs,
    $r$ is the communication-range between ground nodes and $R$ is the communication-range between ground nodes and UAVs.
Output: $Q', S$ //$Q'$ is the new position of existing UAVs and $S$ is the position of new added UAVs
Initially: $Q' \leftarrow Q$; $S \leftarrow \Phi$;
compute new positions of existing UAVs $Q'$ using DBM$(P, Q, l, r, R)$ algorithm;
generate a complete graph $G(V, E)$ over $P \cup Q'$;
compute a minimum spanning tree $T(V, E')$ based on $G(V, E)$;
while true do
    $U \leftarrow \Phi, W \leftarrow \Phi$; //$U$ is the set of 1-degree UAVs and $W$ is the set of edge connecting vertexes in $U$.
    for each $v_i \in V$ do
        if degree$(v_i) == 1, v_i \in Q'$ then
            $U \leftarrow v_i, W \leftarrow e_{i,x}$;
        end if
    end for
    if sizeof$(U) > 0$ then
        $V \leftarrow (V - U), E' \leftarrow (E' - W)$;
    else
        break;
    end if
end while
for each edge $e_i' \in E'$ do
    if $|e_i'| > r$ then
        if $|e_i'| \leq R$ then
            put 1 Steiner points into $s_i$;
        else
            put $\lceil l(e_i')/R \rceil - 1$ Steiner points into $s_i$;
        end if
        $S \leftarrow s_i$;
    end if
end for
return $Q', S$;
```

Algorithm 3: Move-Before-Deployment (MBD) algorithm.

spanning tree T, all existing UAVs have been considered as terminals. However, existing UAV that has 1 degree in T is the leaf of the tree and should not be considered in adding new UAVs in next step. So we recursively cut 1-degree existing UAVs until all existing UAVs have at least 2 degrees and then the number of new added UAVs can be minimized.

The existing UAVs cut process is shown in Figure 4. There are three ground nodes and four existing UAVs in this scenario. The generated minimum spanning tree over all ground nodes and existing UAVs is shown in Figure 4(a). We can find that the degree of existing UAV A is 1. So the existing UAVs cut process will firstly delete UAV A and generate a subtree in Figure 4(b). Again, we find the degree of existing UAV B is 1. So UAV B is also cut and generates a subtree in Figure 4(c). Now, we find that there are not any 1-degree existing UAVs in the tree and all existing UAVs have at least 2 degrees. So we will add new UAVs to edges of the subtree that has larger length than r which is shown in Figure 4(d).

5.4. DAM Algorithm. Although MBD algorithm can minimize the number of new added UAVs using the existing UAVs cut process, the new positions of existing UAVs may not be optimal. This is because the new position of existing UAVs in MBD algorithm is generated using DBM algorithm which is based on the release of motion range constraints. So, when the motion range of existing UAVs is constrained, the performance of MBD algorithm will be depressed.

Since the deployment of new UAVs and the motion control of existing UAVs affect each other, we think that if we can solve the deployment and motion control problem crosswise, then the solution of the joint problem may be optimal. So here we propose a Deploy-Across-Movement (DAM) algorithm that solves these two problems simultaneously.

The main idea of DAM algorithm is as follows. Firstly, we generate a complete graph $G(V, E)$ on ground nodes and sort all edges of E in length increasing order. Then, we consider all edges $e_{i,j}$ in the set E that the length of the edge is no more than r and vertexes of the edge belong to different components. After this step, we will get several components that consist of connected ground nodes. Now we will recursively move existing UAVs and add new UAVs to connect partitioned components until all partitioned components are connected into one component. In each loop, we will try to connect all vertexes pairs v_i and v_j that belong to different components using two different methods. One method uses existing UAVs to set up a communication chain

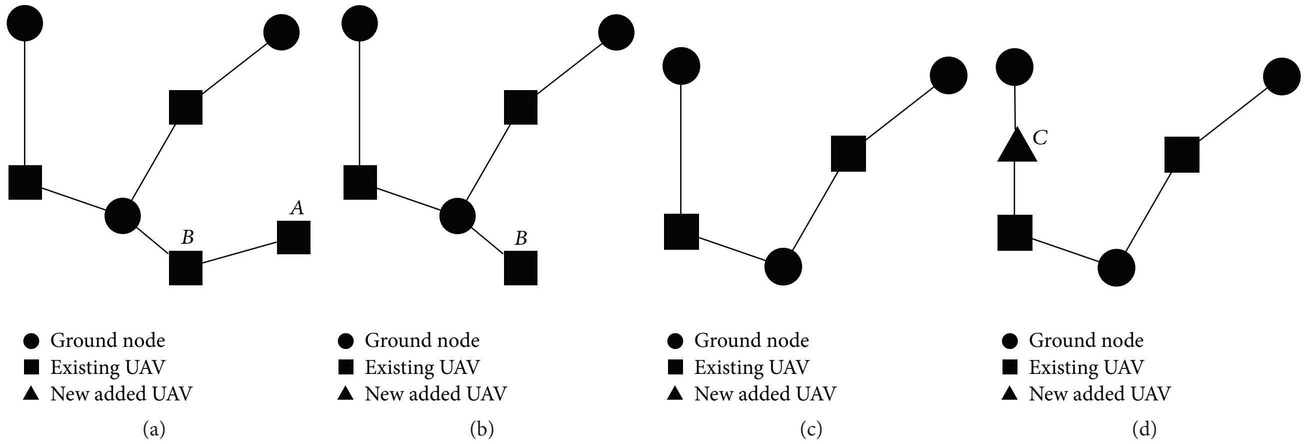


Figure 4: Existing UAVs cut process and new UAVs adding process of MBD algorithm.

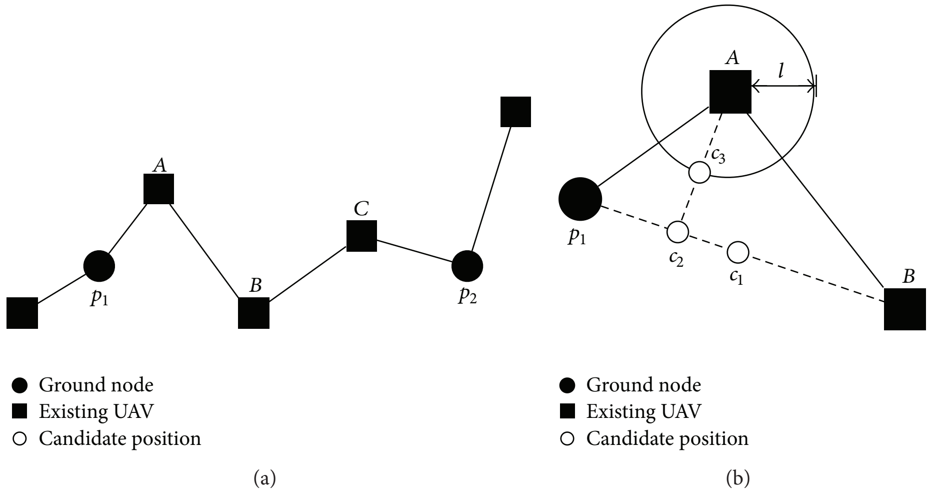


Figure 5: Three candidate positions in EUC method.

between v_i and v_j by moving UAVs to certain positions. New UAVs will be added to edges of the chain which has larger length than r. The other method does not consider existing UAVs and just try to set up a communication chain between v_i and v_j by adding new UAVs. The number of new added UAVs using these two methods will be compared and the less one will be recorded as the minimum number of new UAVs (MNN) for connecting v_i and v_j. The vertexes pair that has minimum MNN will be selected to connect two partitioned components in this loop. New positions of existing UAVs and positions of new added UAVs generated to connect this vertexes pair would also be recorded as part of the final result. The DAM algorithm is shown in Algorithm 4.

The New-UAV-Chain method in DAM algorithm is just the non-EUA algorithm in two-ground node case. The Existing-UAV-Chain (EUC) method is an important part of DAM algorithm since it controls the movement of existing UAVs. The details of Existing-UAV-Chain method are shown in Algorithm 5. Given two ground nodes p_1, p_2 and all existing UAVs Q, it will firstly generate a minimum spanning tree T over set $p_1 \cup Q \cup p_2$ and get the existing UAVs chain C from p_1 to p_2. For each UAV q_i in the chain C, it will compute a new position that depends on position of its left node n_l and right node n_r in the chain. For the first UAV, the left node is p_1 and for the last UAV the right node is p_2. There are three candidate new positions for q_i with different priority. The first candidate position with highest priority is the middle point of nodes p_1 and p_2. The second candidate position with less priority is the shadow point from q_i to the line that consists of nodes p_1 and p_2. The last candidate position with least priority is the point of all reachable points from q_i, nearest to the line that consists of nodes p_1 and p_2 when distance between q_i and the line is larger than the motion range l. The new position of UAV q_i will be set to one of these three candidate positions with highest priority under motion range constraint. Afterwards, new UAVs will be added to edges in chain C that has larger length than r.

Figure 5 shows the three candidate positions in EUC method. There are two ground nodes p_1, p_2 and five existing UAVs in this scenario. The minimum spanning tree generated is shown in Figure 5(a). The UAV chain from p_1 to p_2 is $A \rightarrow B \rightarrow C$. Then EUC method will compute candidate

```
Input: $P, Q, l, r, R$
    // $P$ is the position of ground nodes, $Q$ is the position of existing UAVs, $l$ is the motion-range of existing UAVs, $r$ is the communication-range between ground nodes and $R$ is the communication-range between ground nodes and UAVs.
Output: $Q', S$ //$Q'$ is the new position of existing UAVs and $S$ is the position of new added UAVs
Initially: $Q' \leftarrow Q$; $S \leftarrow \Phi$;
generate a complete graph $G(V, E)$ over $P$;
for each vertex $v_i \in V$ do
    $C_i \leftarrow i$; //$C$ stores the component number for each node.
end for
sort all edges $e_{i,j} \in E$ in length $l(e_{i,j})$ increasing order;
for each edge $e_{i,j} \in E$ that $l(e_{i,j}) \le r$ do
    if $C_i ! = C_j$ then
        all nodes in component $C_i$ changes its component to $C_j$;
    end if
end for
while true do
    $\mathrm{mnn}_{\min} \leftarrow -1$, pair $\leftarrow \Phi$;
    for each edge $e_{i,j} \in E$ that $l(e_{i,j}) > r$ do
        if $C_i ! = C_j$ then
            $\mathrm{mnn}_1 \leftarrow$ Exist-UAV-Chain$(i, j, Q', l, r, R)$;
            $\mathrm{mnn}_2 \leftarrow$ New-UAV-Chain$(i, j, l, r, R)$;
            $\mathrm{mnn} \leftarrow \min(\mathrm{mnn}_1, \mathrm{mnn}_2)$;
            if $\mathrm{mnn} < \mathrm{mnn}_{\min} || \mathrm{mnn}_{\min} < 0$ then
                $\mathrm{mnn}_{\min} \leftarrow \mathrm{mnn}$, pair $\leftarrow [i, j]$;
            end if
        end if
    end for
    if $\mathrm{mnn}_{\min} \ge 0$ then
        generate new positions of existing UAVs $Q'$ and new added UAVs $W$ to connect vertexes in pair;
        $S \leftarrow S \cup W$;
    else
        break;
    end if
end while
return $Q', S$;
```

Algorithm 4: Deploy-Across-Movement (DAM) algorithm.

```
Input: $p_1, p_2, Q, l, r, R$
    // $p_1, p_2$ are two ground nodes, $Q$ is the position of existing UAVs, $l$ is the motion-range of existing UAVs, $r$ is the communication-range between ground nodes and $R$ is the communication-range between ground nodes and UAVs.
Output: $Q', S$ //$Q'$ is the new position of existing UAVs and $S$ is the position of new added UAVs
Initially: $Q' \leftarrow Q$; $S \leftarrow \Phi$;
generate a minimum spanning tree $T$ over $p_1 \cup Q' \cup p_2$;
get UAV chain $C$ from $p_1$ to $p_2$
for each UAV $q_i \in C$ do
    get left node $n_l$ and right node $n_r$;
    compute three candidate positions $\mathrm{CP}_1, \mathrm{CP}_2, \mathrm{CP}_3$;
    choose one position $\mathrm{CP}_j$ with highest priority and satisfy $\mathrm{dist}(q_i, \mathrm{CP}_j) \le l$, here $j = 1, 2, 3$;
    $q_i' \leftarrow \mathrm{CP}_j$;
end for
for each edge $e_i$ in chain $C$ do
    test $|e_i|$ with $r$ and add new UAVs $W$ to $e_i$;
    $S \leftarrow S \cup W$;
end for
return $Q', S$;
```

Algorithm 5: Existing-UAVs-Chain (EUC) method.

Table 1: Simulation parameter.

Parameter	Value
Field size	1–100 km^2
Number of ground nodes	10–100
Number of existing UAVs	2–20
Communication range of ground nodes	50–500 m
Communication range of UAVs	100–1000 m
Motion range	10–100 m

positions for UAVs in the chain one by one. Take the first UAV A as an example. The three candidate positions for A are c_1, c_2, and c_3 shown in Figure 5(b). c_1 is the middle point of p_1 and p_2. c_2 is the shadow point of A to lines p_1, p_2. And c_3 is the point in the motion range of Athat is nearest to lines p_1, p_2. The priority of these three candidates' position is $c_1 > c_2 > c_3$. Due to the motion range of A, the new position it will choose is c_3 in this situation.

5.5. Discussion. In the proposed algorithms, we do not consider the power consumption of UAVs. In fact, the battery power of each UAV is limited and the limited power will certainly affect the deployment and movement of UAVs. Here, the power consumption is modeled as a constant factor. Existing UAVs that have very low battery power should land and recharge; thus it will be excluded from the set of existing UAVs. So all existing UAVs have adequate battery power. The battery power of UAVs can also be modeled as dynamic factor; thus existing UAVs may have different battery power due to duration time of their task. Then the deployment and movement UAVs should consider the power cost and optimize the survival time of the whole network. However, that is beyond the focus of this paper and we may consider it in our future work.

6. Simulation Experiment

A simulation environment was set up to test the performance of the proposed algorithm. Simulation parameters are shown in Table 1.

To demonstrate the effectiveness of proposed Existing UAVs Aware (EUA) algorithms, we compare the performance in terms of new added UAV numbers with non-EUA algorithm and the CBBA (Consensus-Based Bundle Algorithm) proposed by Ponda et al. [15]. Different from non-EUA algorithm, CBBA is an algorithm that considers existing UAVs in deploying new UAVs. It uses distributed planning strategies to control the movement of existing UAVs so that network connectivity can be ensured. We also compare performance among three proposed EUA algorithms: the DBM, MBD, and DAM algorithms. We change the testing scenario by varying 5 simulation parameters including the field size, number of ground nodes, number of existing UAVs, communication range, and motion range. For each scenario, we carry out the simulation on 100 randomly generated topologies and take the average performance as the final performance for this scenario. Our simulation scenarios can be classified into five cases. We will analyze our simulation results one by one.

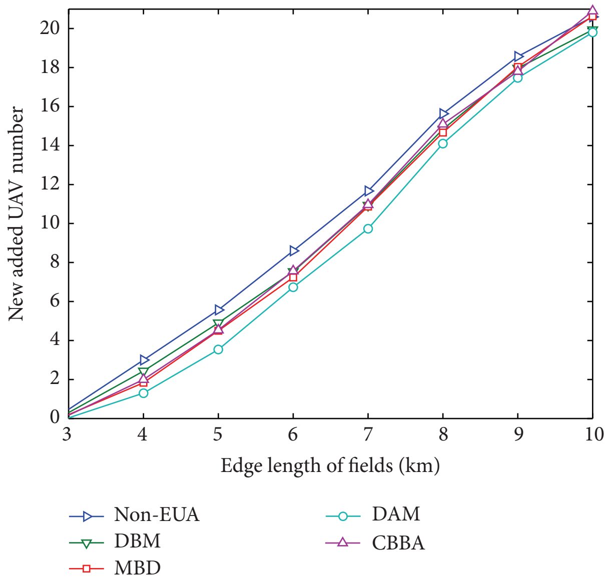


Figure 6: Comparison with varying field size.

6.1. Varying Field Size. We use a square as testing field. We set the number of ground nodes as 50, existing UAVs as 5, communication range as 500 m, and motion range as 50 m. In this scenario, we vary the edge of testing field from 1 km to 10 km by increments of 1 km and randomly generate 100 different topologies for both ground nodes and existing UAVs.

Figure 6 shows the average number of new added UAVs to connect all ground nodes. It is obvious that the number of new added UAVs computed using the EUA algorithms is smaller than the one using non-EUA algorithm. And it is possible to observe an increase in the number of new added UAVs with the increasing size of field. This is because the larger field size is the larger distance between ground nodes since the number of ground nodes is constant. So more relays are needed to maintain the connectivity of ground MANETs. The DAM method has best performance, while DBM method, MBD method, and CBBA method have similar performance.

6.2. Varying Number of Ground Nodes. In this scenario, we set the edge of test field as 5 km and keep other three parameters the same while varying the number of ground nodes from 10 to 100.

Figure 7 shows the average number of new added UAVs to connect all ground nodes. It is obvious that the number of new added UAVs computed using the EUA algorithms is also smaller than the one using non-EUA algorithm. And the performance of DAM algorithm is always the best one of all algorithms. At the beginning, the DBM algorithm is little better than MBD and CBBA algorithm, but, with the increase of ground nodes number, the performance of MBD algorithm

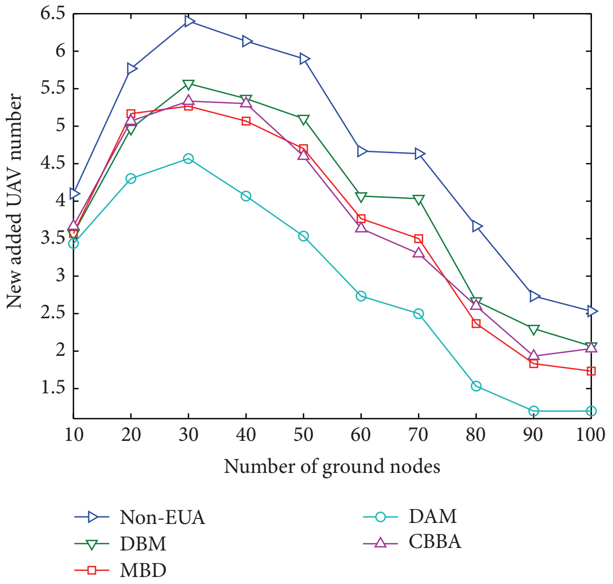


Figure 7: Comparison with varying numbers of ground nodes.

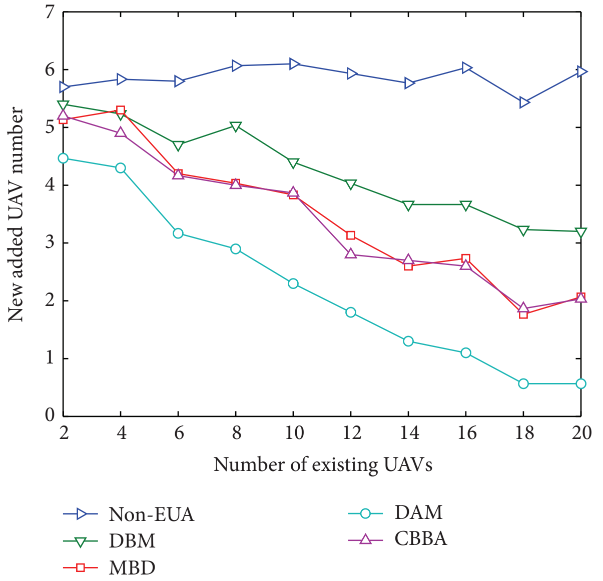


Figure 8: Comparison with varying numbers of existing UAVs.

is better than DBM algorithm. There is a trend in the simulation result that the number of new added UAVs firstly increases but after it reaches a peak, it decreases very quickly, because the increasing number of ground nodes requires more UAVs to maintain its connectivity at the beginning. But afterwards, sufficient number of ground nodes will increase the connectivity of MANETs so less UAVs are needed.

6.3. Varying Number of Existing UAVs. In this case, we set number of ground nodes as 50 and kept the field size, number of ground nodes, communication range, and motion range static. We vary the number of existing UAVs from 2 to 20, by increments of 2.

Figure 8 shows the average number of new added UAVs when given different number of existing UAVs. As we expected, the number of new added UAVs computed using the EUA method is smaller than the one using non-EUA method in all scenarios. Another observation can also emulate the importance of considering existing UAVs. The observation is that there is an obvious decrease in the number of new added UAVs with the increase of the number of existing UAVs when using EUA algorithms, while the number of new added UAVs using non-EUA algorithm almost keeps the same. We can also find that CBBA and MBD algorithm are better than DBM algorithm. DAM algorithm always has the best performance. From Figure 8, we can see that the MBD algorithm can averagely reduce about 30% of new UAVs number compared with CBBA and DBM algorithm. DAM algorithm can averagely reduce about 70% of new UAVs number compared with DBM algorithm.

6.4. Varying Motion Range. In this case, we set existing UAVs as 5 and keep the field size, number of ground nodes, number of existing UAVs, and communication range static. We vary the motion range from 10 m to 100 m, by increments of 10 m, and generate 100 different topologies for each motion range.

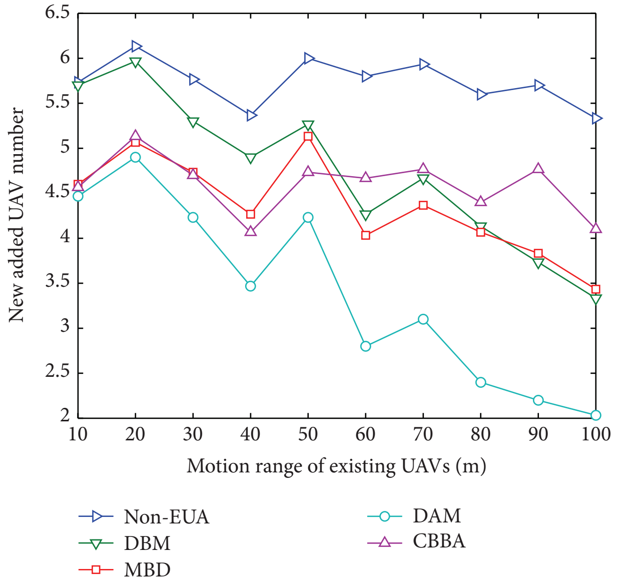


Figure 9: Comparison with varying motion range of existing UAVs.

Figure 9 shows the average number of new added UAVs under different motion range of existing UAVs. EUA algorithms have better performance than non-EUA algorithms and DAM algorithm has best performance. CBBA method firstly has better performance than DBM and MBD method, but when motion range is larger than 50 m the performance of CBBA is worse than BDM and MBD method. The observation of BDM and MBD method is similar to results under varying number of existing UAVs. However, the difference between DBM algorithm and MBD algorithm is decreasing with the increase of motion range and is almost equal to each other at the right end of the figure. This is because MBD algorithm uses DBM algorithm to generate new positions of existing UAVs. And it is based on the release of motion

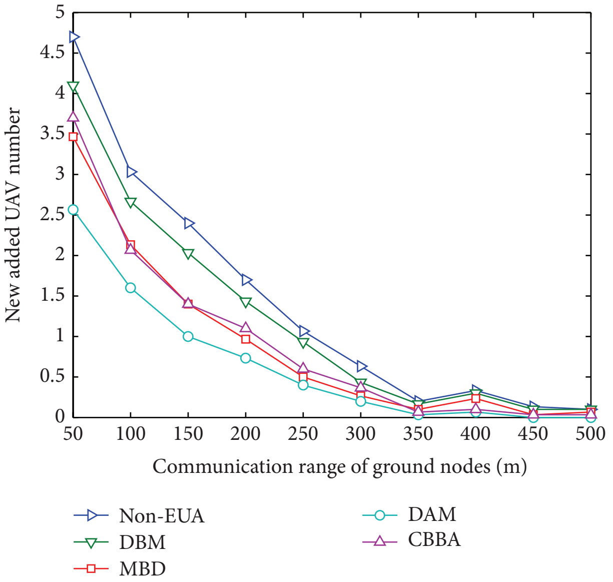


FIGURE 10: Comparison with varying communication range.

range constraint. So when the motion range increases, the difference between DBM and DBM algorithm will vanish.

6.5. Varying Communication Range. In this case, we set motion range as 50 m while keeping the number of ground nodes, number of existing UAVs, and motion range static and vary the communication range of ground nodes from 50 m to 100 m, by increments of 50 m. The communication range between ground nodes and UAVs is set as twice communication range of ground nodes.

Figure 10 shows the average number of new added UAVs under different communication ranges. We can observe that, for all algorithms, the number of new UAVs is decreasing with the increase of communication range. This is due to the fact that larger communication range will lead to better network connectivity. Thus fewer relays are needed to maintain the connectivity of the network. So when the communication range increases to certain value, the difference between all algorithms will vanish. The DAM method has the best performance. CBBA, DBM, and MBD methods have similar performance.

7. Conclusion

This paper studies the problem of using UAVs to maintain the connectivity of ground MANETs. Different from existing works, this paper considered a condition that some UAVs have already been deployed in the field. Due to the movement of ground MANETs and limited communication range, existing UAVs are not able to connect all ground nodes, so new UAVs need to be deployed to maintain the connectivity.

We present a joint optimization problem that combines the motion control of existing UAVs and the deployment of new added UAVs. We formulate this problem as a Minimum Steiner Tree problem with Existing Mobile Steiner points under Edge Length Bound constraints (MST-EMSELB) and prove NP completeness of the problem. We also propose three polynomial time heuristic algorithms named Deploy-Before-Movement (DBM), Move-Before-Deployment (MBD), and Deploy-Across-Movement (DAM) algorithms for the MST-EMSELB problem.

We demonstrate the effectiveness of proposed algorithms by comparing with non-Existing UAVs Aware (non-EUA) algorithm in simulation tests. We generate different scenarios by varying simulation parameters (including the number of ground nodes, number of existing UAVs, communication range, and motion range) and test the performance of both EUA and non-EUA algorithms. Simulation results show that EUA algorithms always have better performance than non-EUA method in terms of new added UAVs number. Among three EUA algorithms, MBD algorithm is better than DBM algorithm in most scenarios and DAM algorithm always has the best performance in all scenarios. In some scenarios, DAM algorithm can reduce at most 70% of new UAVs number compared with DBM algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the National Natural Science Foundation of China under Grant nos. 61379144, 61379145, and 61272485 for their supports of this work.

References

[1] Z. Han, A. L. Swindlehurst, and K. J. R. Liu, “Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3533–3546, 2009.

[2] K. Chandrashekar, M. R. Dekhordi, and J. S. Baras, “Providing full connectivity in large ad-hoc networks by dynamic placement of aerial platforms,” in *Proceedings of the IEEE Military Communications Conference (MILCOM ’04)*, vol. 3, pp. 1429–1436, IEEE, Monterey, Calif, USA, October-November 2004.

[3] F. Jiang and A. L. Swindlehurst, “Optimization of UAV heading for the ground-to-air uplink,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 993–1005, 2012.

[4] C. Dixon and E. W. Frew, “Optimizing cascaded chains of unmanned aircraft acting as communication relays,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 883–898, 2012.

[5] E. L. Lloyd and G. Xue, “Relay node placement in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 134–138, 2007.

[6] W. Zhang, G. Xue, and S. Misra, “Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms,” in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM ’07)*, pp. 1649–1657, IEEE, Anchorage, Alaska, USA, May 2007.

[7] S. Misra, S. D. Hong, G. Xue, and J. Tang, “Constrained relay node placement in wireless sensor networks: formulation and

approximations," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 434–447, 2010.

[8] S. Lee and M. Younis, "Optimized relay placement to federate segments in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 5, pp. 742–752, 2010.

[9] M. A. M. Marinho, E. P. de Freitas, J. P. C. Lustosa da Costa, A. L. F. de Almeida, and R. T. de Sousa Jr., "Using cooperative MIMO techniques and UAV relay networks to support connectivity in sparse wireless sensor networks," in *Proceedings of the International Conference on Computing, Management and Telecommunications (ComManTel '13)*, pp. 49–54, IEEE, Ho Chi Minh City, Vietnam, January 2013.

[10] W. Wang, V. Srinivasan, and K.-C. Chua, "Extending the lifetime of wireless sensor networks through mobile relays," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1108–1120, 2008.

[11] A. Venkateswaran, V. Sarangan, T. F. La Porta, and R. Acharya, "A mobility-prediction-based relay deployment framework for conserving power in MANETs," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 750–765, 2009.

[12] F. El-Moukaddem, E. Torng, and G. Xing, "Mobile relay configuration in data-intensive wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 261–273, 2013.

[13] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: performance and optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2068–2085, 2011.

[14] O. Cetin and I. Zagli, "Continuous airborne communication relay approach using unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1–4, pp. 549–562, 2012.

[15] S. S. Ponda, L. B. Johnson, A. N. Kopeikin, H.-L. Choi, and J. P. How, "Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 861–869, 2012.

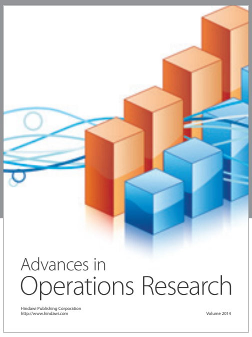
[16] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1996.

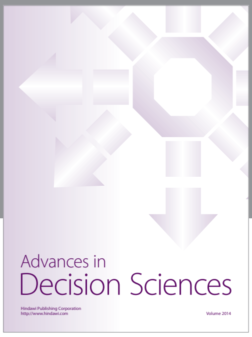
[17] G.-H. Lin and G. Xue, "Steiner tree problem with minimum number of Steiner points and bounded edge-length," *Information Processing Letters*, vol. 69, no. 2, pp. 53–57, 1999.

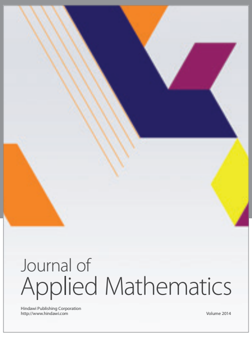
[18] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, and G. Xue, "Approximations for Steiner trees with minimum number of Steiner points," *Journal of Global Optimization*, vol. 18, no. 1, pp. 17–33, 2000.

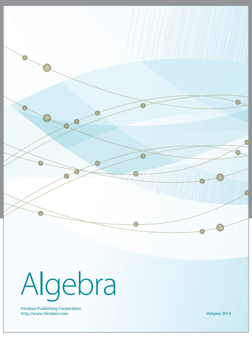
[19] M. Sarrafzadeh and C. K. Wong, "Bottleneck Steiner trees in the plane," *IEEE Transactions on Computers*, vol. 41, no. 3, pp. 370–374, 1992.

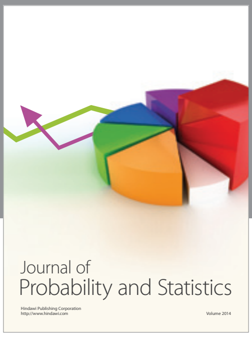
[20] M. Zhu, Z. Cai, D. Zhao, J. Wang, and M. Xu, "Using multiple unmanned aerial vehicles to maintain connectivity of MANETs," in *Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN '14)*, pp. 1–7, IEEE, Shanghai, China, August 2014.


Advances in
Operations Research

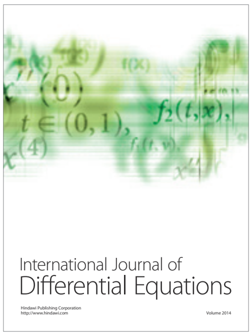

Advances in
Decision Sciences


Journal of
Applied Mathematics

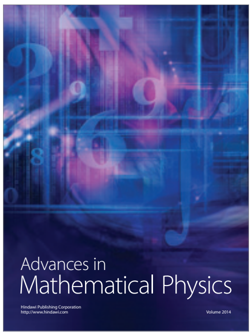

Algebra

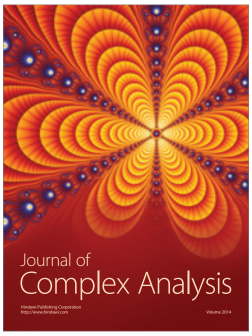

Journal of
Probability and Statistics

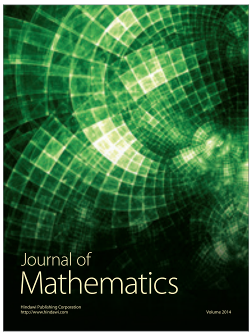
The Scientific
World Journal

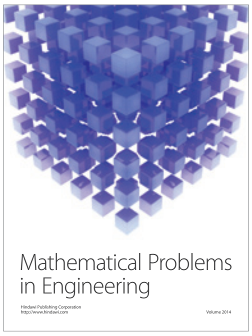

International Journal of
Differential Equations

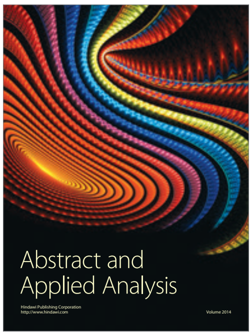
International Journal of
Combinatorics

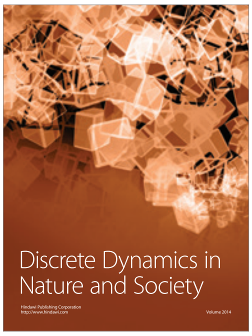

Advances in
Mathematical Physics

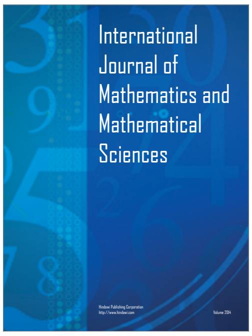

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

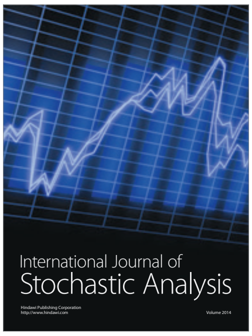

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization