Wireless Networks
https://doi.org/10.1007/s11276-021-02777-x

# UAV deployment with grid modeling and adaptive multiple pruning search in complex forest scenarios

Tianyu Wang[1] · Wei Gu[1]

Accepted: 24 August 2021


**Abstract**

Nowadays, unmanned aerial vehicles (UAVs) have achieved rapid development due to their flexible flight modes and broad application prospects, which have played an important role in scenarios such as aerial photography, unmanned mapping, agricultural plant protection, and power inspection. At the same time, as the frequency of global forest fires is increasing year by year, traditional monitoring and search and rescue methods have little effect. People have begun to consider the introduction of UAVs for forest monitoring and disaster relief. However, in practical applications, when a single UAV performs tasks, there are problems of limited energy, poor robustness, and easy failures that affect the execution efficiency of global tasks. Therefore, it is necessary to obtain the deployment plan of the UAVs offline in advance, and then fine-tune the position of the UAVs online. This paper focuses on the solution of UAV deployment model offline. In complex forest scenarios, due to terrain and signal coverage issues, we need to optimize this when deploying UAVs. At the same time, our goal is to find the location of deployable UAVs in the forest area to maximize the global coverage area of UAVs. Therefore, we divide the forest areas where UAVs need to be deployed and perform a grid modeling on them. On the basis of grid modeling, an Adaptive Multiple Pruning Search Method (AMPSM) is proposed to solve the global UAV deployment and maximum coverage area. In our experiments, we conducted a comparative analysis with the geometric solution to solve the maximum coverage area, which proved the feasibility of the model and the method. The results show that in the complex forest environment, our research can meet the requirements to a great extent, which can be extended to more application scenarios.

**Keywords** Unmanned aerial vehicles · Monitoring · Grid modelling · Deployment plan · AMPSM

## 1 Introduction

Nowadays, unmanned aerial vehicles (UAVs) have achieved rapid development due to their flexible flight modes and broad application prospects [1, 2], which have played an important role in scenarios such as aerial photography, unmanned mapping, agricultural plant protection, and power inspection [3]. At the same time, as the frequency of global forest fires is increasing year by year, traditional monitoring and search and rescue methods have little effect. People have begun to consider the introduction of UAVs for forest monitoring and disaster relief [4]. There are increasingly more researches to leverage the use of UAVs for providing monitoring coverage, and in applications for complex scenarios, some work is keen on using UAVs for forest search and rescue [4–6], which may raise more challenges and problems to be solved.

Traditional forest monitoring methods and monitoring equipment are deployed in fixed locations for a long time by adapting to the average coverage load and forest terrain conditions in the space-time domain, while flexible UAVs do not have such constraints in space and time [7]. Due to this advantage, it is feasible to fly the UAVs to the corresponding altitude for intermittent monitoring in a complex forest environment.

Taking into account the complex terrain conditions of the forest area, before the deployment and monitoring of

✉ Wei Gu
guwei@nuist.edu.cn

Tianyu Wang
uu_wang@foxmail.com

1 Nanjing University of Information Science and Technology, Nanjing, China

Published online: 09 September 2021

UAVs, we need to conduct cognition and environmental modeling of the monitored forest area. In this article, the obtained forest environment is given by satellite images, and then we artificially analyze the forest terrain, altitude, signal coverage and other factors to divide the forest area [8]. After the task of dividing is completed, we model the forest area as a grid environment, use a grid of appropriate precision to represent the forest area grid points, and deploy UAVs on these grid points.

At present, relevant work has involved the emergency deployment of UAVs in corresponding environments. Relevant literature also provides some methods for coordinated deployment of multiple UAVs to achieve maximum wireless coverage. For example, Chen et al. [9] investigate the problem of minimizing the maximum moving distance of sensors on a line interval for reaching the full sensor coverage. Simultaneously, some methods about efficient UAV deployment within current cellular networks become popular gradually [10]. But these problems are all online UAV deployment problems, and only local optimal solutions under time constraints can be obtained. The work of this article is to train our model in an offline environment, which solved the global optimal solution and obtained the global optimal UAV deployment plan, and then deploy the UAVs on the spot to maximize the effective monitoring area. This method is well adapted to the problem of coordinated deployment of UAVs in a complex forest environment, and the forest environment is not changeable. Therefore, the solution obtained can be used for long-term UAV deployment under a specific forest.

The purpose of the study of this paper is to discover a global optimal static deployment plan for UAVs in a gridded forest environment with a specific search method. We want to obtain the optimal UAV deployment plan and maximum coverage area in the same forest environment and in different number of divided areas.

At present, when solving this type of deployment problem, there are many methods of environmental modeling, and ArcGIS is a popular method. However, we adopt a grid modeling method, and after gird modeling the forest, it is placed in a two-dimensional matrix for processing. Any point in the matrix may be the deployment location of the UAV.

After the above modeling of our problem, the problem is transformed into the processing of the matrix. Getting the optimal deployment points in the matrix and obtaining the largest coverage point set is our goal. We use an Adaptive Multiple Pruning Search Method (AMPSM) to search for feasible deployment points in the matrix. Continuously update and iterate to produce a set of optimal deployment plans for UAVs. In our experiments, we compared complex and more accurate geometric solution methods to further verify the feasibility of our experimental results. On the other hand, this comparative analysis also proves that the way we model forest areas and the way we search for UAV deployment plans is feasible.

Through this work we have done, in practical applications, the forest area can also be gridded. When a complex forest area needs to deploy UAVs to perform monitoring tasks, we can train the forest area model offline to obtain a global UAV deployment plan, and then place the UAVs at fixed points, which can save UAV resources. This work is of great practical significance in large and complex forest areas, which can be extended to more application scenarios. In addition, we will apply the offline model to online in further research, make dynamic deployment and anomaly detection for UAVs, and combine the current popular siamese neural network based few-shot learning [11] with existing models to improve the robustness of the fused model in the online environment.

The organization of this paper is as follows. Section 2 specifically introduces the related work of UAV deployment and the feasibility of grid modeling. In Sect. 3, our system model is introduced in detail and the objective function is defined. At the same time, the calculation method of the maximum coverage area is studied. Then, Sect. 4 gives our experimental method in detail and partly explains the algorithm. Finally, in Sects. 5 and 6, we analyze the results of our experiment and conduct a comparative analysis to get some conclusions and remarks.

## 2 Related work

Due to the maturity of UAV hardware technology and UAV communication network technology, and the addition of cloud computing services, the application scenarios of UAVs are becoming more and more extensive, and quite a few papers have explored UAV applications. In the field of UAV monitoring, forest farms with a large area coverage have gradually become the application scenarios of UAVs. Some researchers have investigated this and put forward the current challenges [1], but little work has focused on the deployment of UAVs in a regional environment of complex forests.

In recent years, due to the development of UAV nest technology, we can place the UAV nest at the target location before deploying UAVs for monitoring. The UAV nest can manage the corresponding UAVs and power them (non-working UAVs can also return to the corresponding UAV nests for standby). The UAV nest equipment greatly facilitates the deployment of UAVs in monitoring scenarios, which solves the problems of battery life and UAV management. Therefore, when we deploy UAVs, after we get a global deployment plan, we place a nest (one nest corresponds to one UAV) at the corresponding location, instead of letting a group of UAVs start flying from the starting point to the corresponding location. In fact, in complex scenarios, the deployment methods of the Internet of Vehicles have gradually matured and landed in real applications. For example, the method of using heterogeneous model aggregation to construct the Internet of Vehicles also gives us some enlightenment to our UAV deployment network [12].

When we consider the deployment of UAVs in forest scenarios, due to communication problems, we need to use Ultra Wideband (UWB) technology [13], which means that we need to deploy UWBLOC stations. Taking into account the signal problem, the size of the data transmission and the delay in the data transmission process, before making a UAV deployment decision in this scenario, we first partition the scenario so that the results after the decision have a better robustness.

### 2.1 UAV deployment algorithms

So far, as far as we know, only a few work has focused on the deployment of UAVs. After our analysis, it is found that such a UAV deployment problem is actually NP-hard. In a complex forest environment, it is very difficult to directly deploy UAVs globally and find their latitude and longitude positions. Zhao et al. [14] proposed two UAV deployment algorithms in 2018, the centralized algorithm and the distributed algorithm, to achieve on-demand coverage and maintain interconnection among UAVs at the same time. In their research, the centralized algorithm adopts a heuristic method to select UAVs from all candidate UAV locations iteratively while jointly considering the connectivity among UAVs and the associations between the UAVs and UEs. While the distributed algorithm requires no global information of UEs but autonomously controls the motion of each UAV in a distributed manner with the effect of virtual forces. However, the use scenarios of these two algorithms are limited to applications under certain special requirements, such as optimizing UAV altitude, optimizing tracking trajectory, and so on. It is hard to achieve better results when deploying UAVs in static scenarios to obtain the optimal coverage area with them, so we will further consider dispatching the UAVs dynamically in more complex scenarios [15].

### 2.2 Dynamic deployment strategy

The dynamic deployment strategy [16] of UAVs is also one of the interests of many researchers. Driven by special circumstances, several UAVs need to make decisions in a short time and redeploy their positions. In this kind of emergency dispatch and emergency deployment, reinforcement learning is a good solution. Liu et al. [17] proposed a Q-learning based deployment algorithm, in which each UAV acts as an agent, making their own decision for attaining 3-D position by learning from trial and mistake. This kind of reinforcement learning deployment strategy is particularly suitable for position adjustment and UAV scheduling in emergencies after the relevant UAV static deployment plan has been completed. In our complex forest environment, this kind of reinforcement learning algorithm is particularly suitable for redeploying UAVs in the event of a fire in the forest, completing the task of dynamic deployment (Table 1).

It needs to be mentioned again that the research focus of this article is still on the static deployment of UAVs, training deployment models offline, and conducting normalized monitoring deployment of UAVs in forest environments. When special circumstances occur in forest areas, we will give deployment models for emergencies in future researches. This paper first adopts grid modeling, expresses and partitions irregular forest areas with grids, then uses AMPSM to solve the global optimal deployment plan, and improves the search efficiency through pruning, proving the feasibility of the method.

## 3 System model and problem definition

### 3.1 System description

In the forest farm monitoring scenario and the mobile edge computing environment, the location and deployment of UAVs can be regarded as a network architecture, which is composed by multiple UAVs, multiple pieces of forest and a group of cloud servers. Among them, a UAV may be responsible for a main forest area, or a UAV may also perform monitoring tasks across multiple divided forest areas. In this article, we ignore the possible impact of high-altitude mountains or aerial obstacles on UAV monitoring and communication, because these are not our focus of the research. We regard the coverage of each UAV as a circular area with the centroid of the UAV as the center and $R$ as the radius. In our research, $R$ is a fixed value.

Given a total cost of $C$ to deploy UAVs, where $C$ is a constant value, the cost of deploying each UAV is assumed to be known and the same, and the size of the forest area covered by each UAV is the same. Each UAV can directly access the edge UWB station. Due to the conditions of the forest areas divided by forest farms are different, but the surveillance range is relatively large, it is not necessary to deploy UAVs in each forest area. If several UASs are randomly selected for deployment, the coverage of multiple UAVs will usually overlap, which will greatly waste resources.

**Table 1** Key terms and description

| Key Terms | Description |
|---|---|
| $R$ | The radius of UAV monitoring coverage |
| $C$ | The cost of UAV deployment |
| $K$ | The number of UAVs |
| $S_f$ | The area of total forest |
| $N$ | The total number of forest areas divided |
| $t$ | The remaining cost of deploying UAVs |
| $W_i$ | The cost of deploying the $i$-th UAV |
| $F_i$ | The $i$-th forest area |
| $U_i$ | The area covered by the first $i$ UAVs |
| $S(i, j)$ | The area of the i-th UAV covering of the $j$-th forest area |
| $L_{uav}$ | The location to deploy the UAVs, $L_uav = \{L_1, L_2, ..., L_n\}$ |
| $T_{uav}$ | The target coverage of current UAVs |

The architecture of the forest UAV deployment network is shown in Fig. 1. It consists of cloud server, UWB station and several UAVs. We take the UAV as the center and express its coverage as a circular envelope (the dotted circle in Fig. 1). The Chief UAV is responsible for peripheral monitoring and is always responsible for taking over the location of UAVs that cannot work for some reason (the main UAV is not our focus and is not included in the scope of solving the UAV deployment location and number of it). After the working UAV is deployed, it will hover over the corresponding area to perform its work, and report information to the cloud server when an emergency has been monitored or malfunction occurs. However, the coverage area between UAVs and UAVs often overlaps. As shown in Fig. 1, the area where multiple circles overlap. If there are more overlapping areas, when the UAV sends back monitoring data to the cloud server, there will be multiple data of the same forest area (considered as redundant data). At the same time, the larger the overlapping area, the smaller the average coverage area of a single UAV, which may increase the number of deployed UAVs [18]. Obviously, this is a waste of resources.

## 3.2 UAV location and deployment model

In this Paper, the goal of the UAV location and deployment is to maximize the UAV coverage area of the forest area given a certain cost budget. Therefore, the UAV deployment problem in the mobile cloud computing environment can be expressed as a single-objective optimization problem [19], that is, the problem of maximizing the effective coverage area of any number of UAVs.

We use $0-1$ Matrix $X_{i,j}$ to indicate whether the $i$-th UAV is monitoring the part of the $j$-th forest area. Among them, if $i$-th UAV covers the part of $j$-th forest area, then $X_{i,j} = 1$. Otherwise, $X_{i,j} = 0$. For all $i$ and $j$, where $1 \leq i \leq K$, $1 \leq j \leq N$. Meanwhile, we use binary decision variable $B_{i,j} \in 0, 1$ to Indicate whether to deploy $i$-th UAV

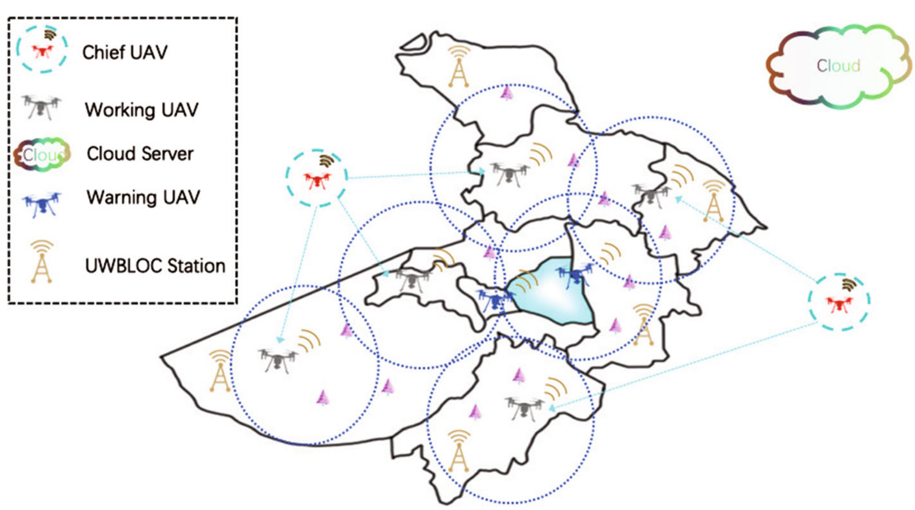


**Fig. 1** UAV deployment model under complex forest area

over the $j$-th forest area. Among then, if the i-th UAV is deployed over the $j$-th forest area, then $B_{i,j} = 1$. Otherwise, $B_{i,j} = 0$. At the same time, the range of $i, j$ is as described above.

Using this method, we assume that every UAV may be deployed over every divided forest area. Because the total number of divided forest areas is $N$, the maximum number of UAVs that can be deployed is $N$. We use $W_i(i = 1, 2, 3...K)$ to represent the cost of deploying every UAV in each forest area ($W_i$ is set to be the same in this paper). In addition, we use $C$ to represent the total cost budget, and $t$ to represent the remaining cost after deploying several UAVs, that is, $t <= C$.

The recursive formula of our UAV monitoring area coverage problem can be described as follows:

$$\begin{aligned} & S(i,0) = S(0,j) = 0 \\ & U_i = U_{i-1} \quad t \le W_i \\ & U_i = U_{i-1} + \sum_{j=1}^{N} S_{i,j} t \ge W_i \end{aligned} \tag{1}$$

### 3.3 Grid model

This work is to solve the global problem of static deployment of UAVs in a complex forest environment. The environment deployment diagram is shown in Fig. 1.

The required solution is the positions of several UAVs. In order to facilitate the solution, geographic ArcGIS [20] modeling is not used here (the location obtained is latitude and longitude), and a relatively simple grid environment modeling is selected (the location obtained is a grid of a certain row and column, and it will also facilitate the solution and optimization variation), the size of the grid does not use a variable-scale learning algorithm in the preliminary stage [21], and is only regarded as a parameter that is artificially adjusted. The setting of this parameter is also regarded as our scale for the division of forest environment.

Figure 1 is used to model the grid environment to get Fig. 2, and the edge of the environment needs to be gridded in the follow-up work.

After grid modeling, we can assume that the obtained forest environment falls in a two-dimensional matrix. Under complex forest conditions, considering the base station configuration and signal issues, the forest area is first artificially divided, as shown in Fig. 3.

Before giving the formal matrix representation of the forest area, we discuss the following constraints.

**Fig. 2** Forest area based on grid modeling


**Fig. 3** Forest area after artificial partition

- Due to the base station deployment and signal issues, only one UAV can be placed in each forest area after artificial division.
- The total cost $C$ determines how many UAVs can be deployed, that is, how many forest areas can deploy UAVs at most. To simplify the problem, assume that the cost of deploying each UAV $W_i$ is the same.
- The coverage radius of each UAV $R$ is equal and constant.

When the forest environment is grid modeled, we select the deployable UAVs in a two-dimensional matrix for a certain cost, and the goal is to obtain the maximum coverage area of the forest by the UAVs, which reflected that the most effective grid points with the most coverage in the matrix (Fig. 4).

The two-dimensional matrix of the entire forest area and surrounding environment is expressed as follows:

In this matrix, the invalid environment that does not belong to the forest area is marked as 0, and the point set in

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 8 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 8 | 8 | 8 | 9 | 9 | 9 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 10 | 10 |
| 0 | 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 10 | 10 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 7 | 7 | 7 | 10 | 10 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 7 | 7 | 7 | 7 | 7 | 10 | 10 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 5 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 4 | 6 | 1 | 1 | 1 | 7 | 7 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 4 | 4 | 1 | 1 | 1 | 7 | 7 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 7 | 7 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Fig. 4** The two-dimensional matrix of the entire forest area and surrounding environment

the forest area are numbered according to the numbers that divide the area.

Also, in this matrix, we specify the UAV selection rules and the formal representation of the covered area when solving the deployment plan. The formalized problem is dened by

$$\max T_{uav} \tag{2}$$

and

$$T_{uav} = effective \sum\nolimits_{i=1}^{K} S(i,j), \sum\nolimits_{i=1}^{K} W_i \leq C \tag{3}$$

## 4 Deployment algorithm

At this time, the coverage of the UAV when deployed is the set of neighbor points at its location. This method of updating the matrix point set is very similar to the chromosome mutation in genetic algorithm [22]. In our update process, the updated point set depends on the UAV's position in the matrix and the UAV's monitoring radius. When the neighbor point set of the UAV's location is completely contained in the circular coverage area of the UAV, these neighbor point set will be updated initially.

$$Matrix[i][j] = \begin{cases} A, & \begin{array}{c} \text{if } Matrix[i][j] \; ! = 0 \\ \text{and } Matrix[i][j] \\ \text{in the coverage,} \end{array} \\ Matrix[i][j], & \text{otherwise.} \end{cases} \tag{4}$$

After some point set in the matrix have been occupied by UAVs, we need to perform two tasks. On the one hand, it is to mark the area where the UAV is located to prevent the subsequent deployment of the UAV from occupying the points in the area; on the other hand, it is to calculate the coverage area generated by the current UAV with the points updated. When updating these neighbor point sets, it is obvious that the non-forest point set with 0 in the matrix should not be updated. In addition, when a UAV is deployed at a certain location, if some neighbor points in its coverage have been updated (that is, the coverage area of the UAV overlaps with another deployed UAV's coverage area), these neighbor points will not be updated and keep the state.

When searching for the coverage point set of a certain UAV, we use the location of the UAV as the center and gradually search from its surroundings. In addition, when getting a better deployment point, we perform a backtracking operation.

Obviously, after grid modeling the forest environment, our goal can be further optimized to solve so that the number of updated points in the matrix reaches the maximum.

$$\max \; num(Matrix[i][j] == A) \tag{5}$$

**Algorithm 1** Solving the UAV Deployment Plan by Adaptive Multiple Pruning Search Method

```
Require: F, R, N, M = {μ1, μ2, ..., μn}
Ensure: L*:final locations in Matrix    C*:optimal coverage
C= −INT_MAX. L = φ, M̄=M
for x = 1 to F do
    if calc(x, R, F)≤ C then
        return C
    end if
    if x == F then
        return C
    end if
    for y = 0 to N do
        for t = 0 to R do
            Update current optimal coverage (y)
            Update current deployment plan (x)
        end for
    end for
    while L! = φ do
        L*.back←L
        Update current optimal coverage (x)
    end while
end for
return C ← maxC
```

In this paper, we use an adaptive multiple pruning search method (AMPSM), as shown in Algorithm 1. Our goal is to meet the aforementioned constraints under the condition of a certain cost, so that the deployment position of the UAVs meets the maximum coverage.

In this algorithm, we conducted an in-depth analysis of the model and realized that this NP-hard search process can be pruned. Therefore, we innovatively propose this adaptive multi-pruning search method. The maximum possible number of coverage points is compared with the maximum number of coverage points in the current search state. If the maximum possible number of coverage points is less than the maximum number of coverage points in the current search state, stop the search state [23]. At the same time, we also carried out pruning during the backtracking process.

At this time, the $M$ in the algorithm is the two-dimensional matrix we get, which represent the forest area and the surroundings [24]. Since the cost $C$ is determined in advance, and assuming that the cost of deploying each UAV $W_i$ is the same, we can get the maximum number of UAVs that can be deployed in advance based on the cost and the number of the divided areas. $F$ is the total number of divided areas, and the UAV's monitoring coverage radius $R$ is also given in the algorithm.

It can be seen from the output of the algorithm that the final deployment locations $L^*$ and the maximum coverage area $C^*$ of the UAVs are what we need. We store the deployment locations $L^*$ of the UAVs in an array. During the execution of the algorithm, we generate the array gradually.

The two most important parts of this algorithm are to update the maximum coverage area and the optimal deployment plan. As we can see, we have updated both in the process of iteration and backtracking. In the process of backtracking, we focus on processing $L^*$, which is the key to generating the final deployment plan. This process will be repeated until the maximum number of iterations $F$ is met. Finally, the optimal coverage $C$ is obtained.

**Table 2** Parameter settings

| Key Parameter | Value |
|---|---|
| $F$ | 10 |
| $R$ | 2 |
| $C$ | INT_MAX |
| *Coverage* | -INT_MAX |
| $N$ | 11 |
| $M$ | *Matrix* [18][14] |

# 5 Eeperimental results and performance evaluation

## 5.1 Simulation setup

In our experiment, MacBook Air with Quad-Core Intel Core i5 at 1.1GHz, 256GB Hard Disk and 8GB RAM is leveraged to conduct simulation experiments. In Table 2, the key parameters are illustrated as shown.

**Fig. 5** UAV deployment results with AMPSM (Partition pattern: F = 4, K = 4)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 3 | 3 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 3 | 3 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 3 | 3 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 3 | 3 | 3 | 1 | 1 | 1 | 4 | 4 | 2 | 2 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(a) Initial state**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | -1 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 3 | 3 | -1 | 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 3 | 3 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 3 | 3 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 3 | 3 | -1 | 1 | 1 | 1 | 4 | 4 | 2 | 2 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | -1 | 2 | 2 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(b) Deployment result**

**Table 3** Experimental results in a given two-dimensional matrix

| The number of divided areas | Maximum coverage (Unit) | Average algorithm running time (Sec) |
|---|---|---|
| 4 | 52 | 0.004530 |
| 6 | 78 | 0.019516 |
| 8 | 102 | 0.456046 |
| 10 | 120 | 2.034920 |

The comparative approach is leveraged to make comprehensive evaluations, and this approach is described as follows:

Geometric solution: This approach regards the coverage area of a UAV as a circle, the center point coordinates are the grid position selected by the UAV, and the radius of the circle is the UAV coverage radius. Such area of a circular is the coverage area of a UAV [25]. Since UAVs are deployed in various locations, if the areas covered by the two UAVs do not overlap, the effective total coverage area is directly added to the area of the two circular areas. If they overlap, the effective total coverage area must be

**Fig. 6** UAV deployment results with AMPSM (Partition pattern: F = 6, K = 6)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 6 | 6 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 4 | 4 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 0 |
| 0 | 0 | 0 | 3 | 3 | 5 | 6 | 6 | 6 | 6 | 4 | 4 | 4 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 4 | 6 | 4 | 4 | 4 | 4 | 4 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 4 | 4 | 4 | 1 | 1 | 2 | 2 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(a) Initial state**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | -1 | 5 | 6 | 6 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 4 | 4 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 4 | 4 | -1 | 4 | 4 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | -1 | 4 | 4 | 4 | 4 | 0 |
| 0 | 0 | 0 | 3 | 3 | 5 | 6 | 6 | 6 | 6 | 4 | 4 | 4 | 0 |
| 0 | 0 | 0 | 3 | 3 | -1 | 4 | 6 | 4 | 4 | 4 | 4 | 4 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 4 | 4 | 4 | 1 | -1 | 2 | 2 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | -1 | 2 | 2 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(b) Deployment result**

**Fig. 7** UAV deployment results with AMPSM (Partition pattern: F = 8, K = 8)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 8 | 8 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 6 | 6 | 8 | 8 | 8 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 8 | 8 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 8 | 8 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 7 | 8 | 8 | 8 | 8 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 8 |
| 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 5 | 6 | 6 | 6 | 7 | 7 | 4 | 4 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 4 | 6 | 4 | 4 | 4 | 4 | 4 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 4 | 4 | 4 | 1 | 1 | 2 | 2 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(a)** **Initial state**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 8 | 8 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | -1 | 5 | 6 | 6 | 8 | 8 | 8 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | -1 | 8 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 8 | 8 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | -1 | 8 | 8 | 8 | 8 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 8 |
| 0 | 0 | 0 | 5 | 5 | 5 | -1 | 6 | 6 | 7 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 5 | 6 | 6 | 6 | 7 | 7 | 4 | 4 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 4 | 6 | 4 | 4 | -1 | 4 | 4 | 0 |
| 0 | 1 | 1 | 3 | -1 | 1 | 4 | 4 | 4 | 1 | 1 | 2 | 2 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 |
| 1 | -1 | 1 | 1 | 1 | 2 | 2 | -1 | 2 | 2 | 2 | 2 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(b)** **Deployment result**

**Fig. 8** UAV deployment results with AMPSM (Partition pattern: F = 10, K = 10)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 8 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 8 | 8 | 8 | 9 | 9 | 9 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 10 | 10 |
| 0 | 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 10 | 10 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 7 | 7 | 7 | 10 | 10 |
| 0 | 0 | 0 | 0 | 0 | 5 | 5 | 7 | 7 | 7 | 7 | 7 | 10 | 10 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 5 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 4 | 6 | 1 | 1 | 1 | 7 | 7 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 4 | 4 | 1 | 1 | 1 | 7 | 7 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 7 | 7 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(a)** **Initial state**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 8 | 8 | -1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 8 | 8 | 8 | 9 | 9 | 9 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | -1 | 9 | 9 | 9 | 10 | 10 | 10 |
| 0 | 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 9 | -1 | 10 | 10 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 7 | 7 | 7 | 10 | 10 |
| 0 | 0 | 0 | 0 | 0 | -1 | 5 | 7 | 7 | 7 | 7 | 7 | 10 | 10 |
| 0 | 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 0 | 5 | 5 | 6 | 6 | -1 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 0 |
| 0 | 0 | 0 | 3 | -1 | 5 | 6 | 6 | 6 | 6 | 7 | -1 | 7 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 4 | 6 | 1 | 1 | 1 | 7 | 7 | 0 |
| 0 | 1 | 1 | 3 | 3 | 1 | 4 | -1 | 1 | 1 | 1 | 7 | 7 | 0 |
| 1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 7 | 7 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | -1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(b)** **Deployment result**

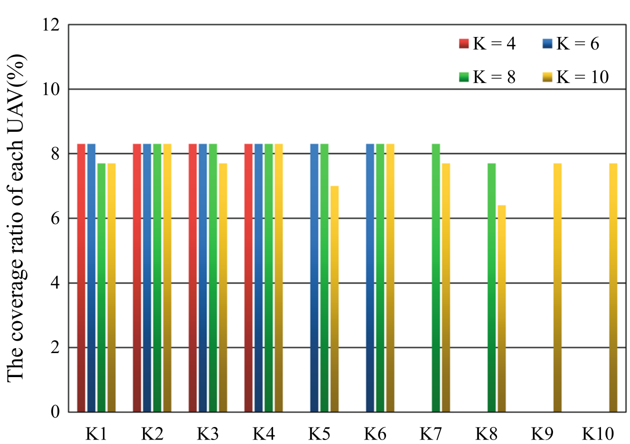


**Fig. 9** The coverage ratio of each UAV in different number of divided areas

added to each individual small area. In the following series of geometric figures, we will give calculation methods for various overlapping situations [26].

## 5.2 Performance evaluation on adaptive multiple pruning search method

In this section, we conduct simulations to evaluate the performances of our proposed deployment algorithm. All the values reported later are collected from the average of 100 runs for each algorithm. The number of divided forest areas is set to 10, and the coverage radius of the UAV is set to 2 units. These values are obtained by equivalent transformation after mapping the forest area to a two-dimensional matrix.

#### 5.2.1 Deploy UAVs in the given two-dimensional matrix

We first experiment with Algorithm 1 in an 18*14 two-dimensional matrix. At this time, the total grid points are 252 and the total forest area is 156. Figure 5 shows the maximum coverage area obtained by deploying UAVs when the forest area is divided into different numbers of areas [27].

It can be seen from Table 3 that as the number of divided areas increases, the maximum coverage area increases regularly. As the number of divided areas increases, the running time of the algorithm also exponentially increases. Note that when the number of deployed UAVs is greater than 6, the running time of the algorithm has an explosive growth, but the performance of the algorithm in solving the global deployment plan is very stable. Therefore, depending on the size of the target area and heterogeneity of UAVs, an appropriate number of UAVs needs to be selected and deployed. In particular, the experimental data show that on the basis of ensuring the effective monitoring coverage of UAV clusters deployed in complex scenarios, the running time of the algorithm is maintained at the level of milliseconds and seconds under small-scale area division. So it can be predicted under this data that in practical engineering applications, large-scale area division may make the number of UAVs and area division reach 100 or more, and the model can ensure that the UAVs deployment can be completed in a few minutes and ensure more accurate global optimal coverage benefits. The above analysis reflects the good robustness of the model in this paper.

At the same time, in our experiment, we give the UAV deployment results under different forest divisions. As shown in Figs. 5, 6, 7, and 8, (a) represents the initial state of the forest area in the matrix, and the points marked -1 in (b) are the location of the UAVs we need to deploy.

After deploying these UAVs, we analyze the coverage ratio of UAVs in different areas relative to the forest area [28]. This work can help us analyze the experimental results by ASPSM. Figure 9 shows the UAV coverage ratio of each area in different regions [29]. It should be mentioned that the maximum coverage area of each UAV is 13 Units, which means that the maximum coverage ratio of each UAV relative to the entire forest area is 8.3% When we analyze the experimental results in a certain situation [30], we first use the formula $\sum_{i=1}^{K} ratio(K_i)$ to find the total coverage of all deployed UAVs. For example, when the Partition pattern was chosen ($F$ = 10, $K$ = 10), we evaluate the difference between the total coverage ratio of UAVs in each region and the overall effective coverage. By calculation, the value of $\sum_{i=1}^{10} ratio(K_i)$ is 76.923%, and the value of effective coverage ratio is 76.923%. This means that our AMPSM algorithm reduces the overlapping coverage area to a minimum on the basis of meeting the

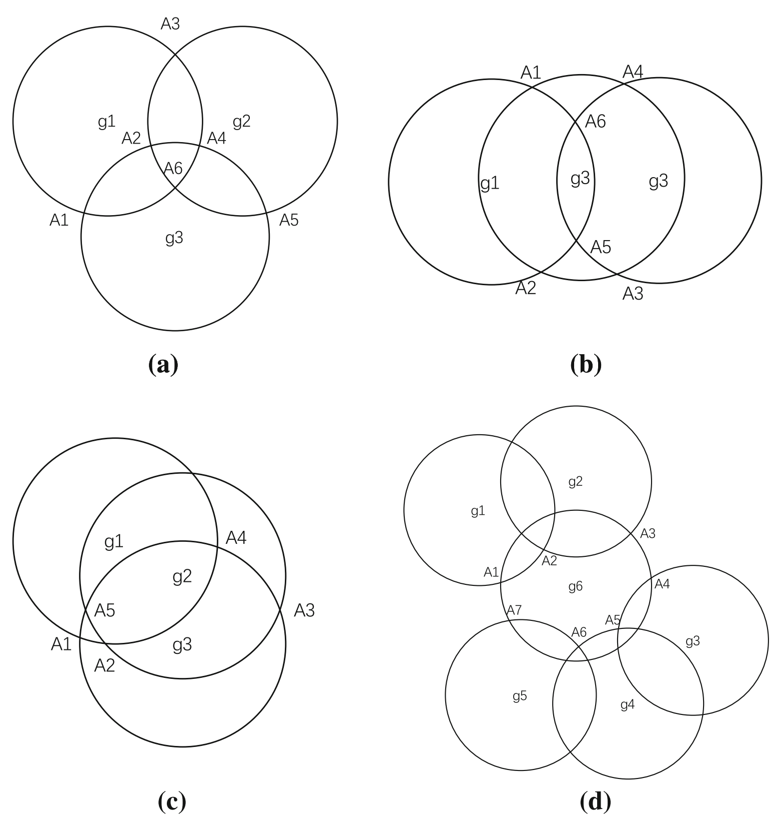


**Fig. 10** The several situations of the UAV deployment

maximum coverage, and the average effective coverage area of each UAV in the forest area is larger, which proves the advantage of the algorithm in this scenario.

### 5.2.2 Coverage comparison with geometric solution

In this paper, we use geometric figures to detail the calculation method of the effective coverage area of the UAVs, which will be our comparative analysis. We regard the coverage area of a UAV as a circle, the center point coordinates are the grid position selected by the UAV, and the radius of the circle is the UAV coverage radius. Such area of a circular is the coverage area of a UAV. Since UAVs are deployed in various locations, if the areas covered by the two UAVs do not overlap, the effective total coverage area is directly added to the area of the two circular areas. If they overlap, the effective total coverage area must be added to each individual small area. In the following series of geometric figures, we will give calculation methods for various overlapping situations [26].

Figure 10 shows the several situations of the UAV deployment. In the Fig. 10a, g1, g2 indicate that two UAVs have been deployed, and the coverage area of the two UAVs g1 and g2 is known, and then we start to deploy the third UAV g3 (assuming $t >\ = W_i$ at this time). Since the coverage area of g3 overlaps with g1 and g2 at this time, when calculating the total coverage area, the circular area of g3 is not directly added, but the non-overlapping area is added [31, 32]. At this time, the calculation formula for the non-overlapping area is:

$$S = S_{(arch)}A_1A_5 + S_{(polygon)}A_1A_6A_5\S_{(arch)}A_1A_6 - S_{(arch)}A_5A_6 \tag{6}$$

In the Fig. 10b, g1, g2 indicate that two UAVs that have been deployed, and then the third UAV g3 is to be deployed. The difference from (a) is that the non-overlapping area in (b) is divided into two parts. The calculation formula for the area of the lower half of the non-overlapping area is:

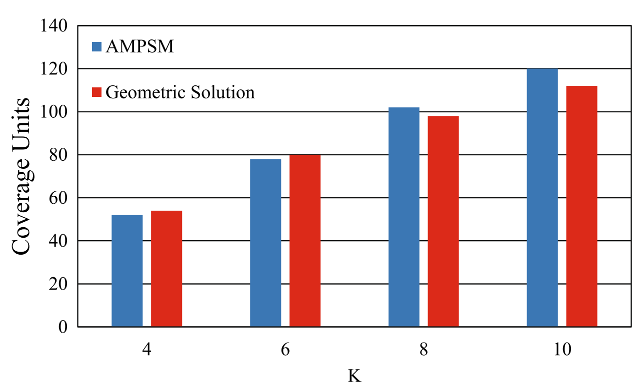


**Fig. 11** The coverage comparison of two methods

$$S = S_{(arch)}A_2A_3 + S_{(polygon)}A_2A_3A_5\S_{(arch)}A_2A_5 - S_{(arch)}A_3A_5 \tag{7}$$

Similarly, the calculation formula for the area of the upper non-overlapping area is:

$$S = S_{(arch)}A_1A_4 + S_{(polygon)}A_1A_4A_6\S_{(arch)}A_1A_6 - S_{(arch)}A_4A_6 \tag{8}$$

In the Fig. 10c, g1, g2 indicate that two UAVs that have been deployed, and then the third UAV g3 is to be deployed. The calculation formula for the non-overlapping area is:

$$S = S_{(arch)}A_1A_3 - S_{(polygon)}A_1A_2A_3\S_{(arch)}A_1A_2 - S_{(arch)}A_2A_3 \tag{9}$$

In the Fig. 10d, g1, g2, g3, g4, g5 indicate that five UAVs that have been deployed, and then the 6-th UAV g6 is to be deployed. The calculation formula for the non-overlapping area is:

$$\begin{aligned} S = {} & S_{(polygon)}A_1A_2A_3A_4A_5A_6A_7 + S_{(arch)}A_3A_4 \\ & + S_{(arch)}A_1A_7 - S_{(arch)}A_1A_2 - S_{(arch)}A_2A_3 \\ & - S_{(arch)}A_4A_5 - S_{(arch)}A_5A_6 - S_{(arch)}A_6A_7 \end{aligned} \tag{10}$$

The advantage of geometrically solving the coverage area is the accuracy when solving the overlapping area, but the iteration is often complicated, and multiple iterations are required to calculate the overall effective coverage area of multiple UAVs. When we use grid modeling, we also optimize the overlap area calculation, which makes our algorithm perform well compared to the geometric solution when searching the maximum coverage area [33]. The specific comparison is given in Fig. 10.

Obviously, we can see from Fig. 11 that grid modelling and AMPSM have also obtained good results for more accurate but complex geometric solutions, which further proves the feasibility of our model. And in the complex forest environment, our method is more suitable for UAV deployment, which means that it is convenient to obtain a feasible deployment plan.

At the same time, we can find that when $K < 8$, the geometric solution method can obtain a larger coverage area than ASPSM, and when $K \geq 8$, ASPSM can obtain better results than geometric solution. When the $K$ value is small, the geometric solution method is better and easy to understand. When the $K$ value is large, the grid model we proposed greatly reduces the search space for UAV deployment, and the search units in the grid model are all clear and knowable, which can quickly reach the global convergence. This makes us finally obtain the minimum overlap area in the example of $K = 10$, so that the effective coverage area achieves a better result of the relatively geometric solution.

## 6 Conclusion

The deployment of multiple UAVs in forest to provide monitoring coverage is of great practical importance. To the best of our knowledge, this is the rst work to deal with the UAV deployment of maxing coverage under a grid modelling. At the same time, we also proposed an adaptive multiple pruning search method to solve the optimal deployment plan of UAVs. When UAVs are initially located in grid points, the deployment problem is proved to be NP-hard. By using AMPSM, we have obtained the optimal UAV deployment plan and maximum coverage area under different number of divided areas. The theoretical results draw in this paper are further conrmed by our experimental evaluation.

## References


1. Wang, Z., Duan, L., & Zhang, R. (2019). Adaptive deployment for UAV-aided communication networks. *IEEE Transactions on Wireless Communications, 18*(9), 4531–4543.
2. Mengqi, L., Liang, L., Ying, G., Youwei, D., & Lisong, W. (2021). Minimizing energy consumption in wireless rechargeable UAV networks. *IEEE Internet of Things Journal*, pp. 1–1 .
3. Anderson, B. O., Fidan, Bariş, C. Y., & Walle, D. (2008). Uav formation control: Theory and application. *Recent Advances in Learning and Control* (pp. 15–33). Springer.
4. Sankey, T., Donager, J., McVay, J., & Sankey, J. B. (2017). UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA. *Remote Sensing of Environment, 195,* 30–43.
5. Fang, F., Qi Jiao, F., Richard, Y., Zhang, Z., & Jianbo, D. (2021). Securing uav-to-vehicle communications: A curiosity-driven deep q-learning network (c-dqn) approach. (pp. 1–6)
6. Gharib, M., Nandadapu, S., & Afghah, F. (2021). An exhaustive study of using commercial lte network for uav communication in rural areas. (pp. 1–6)
7. Samad, A.M., Kamarulzaman, N., Muhammad, A. H., Thuaibatul, A. M., & Khairil, A. H. (2013). The potential of unmanned aerial vehicle (UAV) for civilian and mapping application. (pp. 313–318).
8. Juan Jesús, R., Guillaume, J., David, S., Jaime, D. C., & Antonio, B. (2015). Mini-UAV based sensory system for measuring environmental variables in greenhouses. *Sensors, 15*(2), 3334–3350.
9. Chen, D. Z., Yan, G., Li, J., & Wang, H. (2013). Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. *Discrete and Computational Geometry, 50*(2), 374–408.
10. Pan, C., Yin, C., Beaulieu, N. C., & Jian, Yu. (2019). 3d UAV placement and user association in software-defined cellular networks. *Wireless Networks, 25*(7), 3883–3897.
11. Zhou, X., Liang, W., Shimizu, S., Ma, J., & Jin, Q. (2020). Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics, 17*(8), 5790–5798.
12. Zhou, X., Liang, W., She, J., Yan, Z., & Wang, K. (2021). Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles. *IEEE Transactions on Vehicular Technology*.
13. Robert, J. F. (2004). Recent system applications of short-pulse ultra-wideband (UWB) technology. *IEEE Transactions on Microwave Theory and Techniques, 52*(9), 2087–2104.
14. Zhao, H., Wang, H., Weiyu, W., & Wei, J. (2018). Deployment algorithms for UAV airborne networks toward on-demand coverage. *IEEE Journal on Selected Areas in Communications, 36*(9), 2015–2031.
15. Wang, J., Liu, K., & Pan, J. (2020). Online UAV-mounted edge server dispatching for mobile-to-mobile edge computing. *IEEE Internet of Things Journal, 7*(2), 1375–1386.
16. Zhou, L. V., Swain, Akshya, & Ukil, Abhisek. (2018). Q-learning and dynamic fuzzy q-learning based intelligent controllers for wind energy conversion systems. (pp. 103–108).
17. Liu, X., Liu, Y., & Chen, Y. (2019). Reinforcement learning in multiple-UAV networks: Deployment and movement design. *IEEE Transactions on Vehicular Technology, 68*(8), 8036–8049.
18. William, B., Jesse, R., & Derral, T. (2006). Identifying a UAV landing location, July 27 . US Patent App. 11/041,505.
19. Sarnak, N., & Tarjan, R. E. (1986). Planar point location using persistent search trees. *Communications of the ACM, 29*(7), 669–679.
20. Liu, H., Zhen, Y., Li, J., Cong, H., & Pan, J.-S. (2017). The framework of remote image management based on arcgis server. *In 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS)*, (pp. 1–5), IEEE.
21. Berger, M., & Rigoutsos, I. (1991). An algorithm for point clustering and grid generation. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(5), 1278–1286.
22. Koushik, A. M., Fei, H., & Kumar, S. (2019). Deep learning-based node positioning for throughput-optimal communications in dynamic UAV swarm network. *IEEE Transactions on Cognitive Communications and Networking, 5*(3), 554–566.
23. Hua, Q., Zhang, W., Zhao, J., Luan, Z., & Chang, C. (2020). Rapid deployment of uavs based on bandwidth resources in emergency scenarios. *In 2020 Information Communication Technologies Conference (ICTC)*, (pp. 86–90). IEEE.
24. Baiocchi, V., Dominici, D., & Mormile, M. (2013). UAV application in post-seismic environment. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1,* W2.
25. Xu, X., Wu, Q., Qi, L., Dou, W., Tsai, S. B., & Bhuiyan, M. Z. A. (2021). Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems, 22*(3), 1787–1796.
26. Itkin, M., Kim, M., & Park, Y. (2016). Development of cloud-based UAV monitoring and management system. *Sensors, 16*(11), 1913.
27. Chen, H., & He, K.F. (2015). U-turn optimization for three-dimensional area coverage of uav. *In 2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, (pp. 98–103). IEEE.
28. Jia, J., Liu, C., Chen, J., & Xueli, W. (2012). Design of energy aware movement-assisted deployment in wireless sensor network. *In 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, (pp. 290–292), IEEE.
29. Xu, X., Shen, B., Yin, X., Khosravi, M. R., Wu, H., Qi, L., & Wan, S. (2021). Edge server quantification and placement for offloading social media services in industrial cognitive IOV. *IEEE Transactions on Industrial Informatics, 17*(4), 2910–2918.

30. Enoc, S.-A., Jim, H. C., & Celestino, C. (2018). Accuracy of unmanned aerial vehicle (uav) and sfm photogrammetry survey as a function of the number and location of ground control points used. *Remote Sensing*, 10(10):1606, 2018@miscbodin2006identifying, title=Identifying a UAV landing location, author=Bodin, William and Redman, Jesse and Thorson, Derral, year=2006, month=jul, publisher=Google Patents, note=US Patent App. 11/041,505.
31. Ruan, L., Wang, J., Chen, J., Yitao, X., Yang, Y., Jiang, H., et al. (2018). Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework. *China Communications, 15*(10), 194–209.
32. Xiaolong, X., Qihe, H., Yiwen, Z., Shancang, L., Lianyong, Q., & Wanchun, D. (2021). An lsh-based offloading method for iomt services in integrated cloud-edge environment. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(3s).
33. Xia, L., & Anthony, G.-O.Y. (2005). Integration of genetic algorithms and gis for optimal location search. *International Journal of Geographical Information Science, 19*(5), 581–601.



**Tianyu Wang** is currently working towards his B.S. degree in Software Engineering at School of Computer and Software in Nanjing University of Information Science and Technology. His research interests include IoT, UAV application, and Deep Learning.

**Wei Gu** received his M.S. degree from the Nanjing University of Information Science and Technology, China, in 2007. He is currently a senior lab master at the school of computer and software, Nanjing University of Information Science and Technology. His research interests include edge computing, information security, cloud computing and big data.